Frontiers in Bioscience Landmark

Front. Biosci. (Landmark Ed) **2025**; 30(4): 36603
https://doi.org/10.31083/FBL36603

Original Research

DTC-m6Am: A Framework for Recognizing N6,2′-O-dimethyladenosine Sites in Unbalanced Classification Patterns Based on DenseNet and Attention Mechanisms

Hui Huang[1], Fenglin Zhou[1,*], Jianhua Jia[1], Huachun Zhang[1]

[1]School of Information Engineering, Jingdezhen Ceramic University, 333403 Jingdezhen, Jiangxi, China
*Correspondence: 004278@jcu.edu.cn (Fenglin Zhou)
Academic Editor: Xudong Huang
Submitted: 26 December 2024 Revised: 12 March 2025 Accepted: 25 March 2025 Published: 24 April 2025

Abstract

Background: m^{6}Am is a specific RNA modification that plays an important role in regulating mRNA stability, translational efficiency, and cellular stress response. m^{6}Am's precise identification is essential to gain insight into its functional mechanisms at transcriptional and post-transcriptional levels. Due to the limitations of experimental assays, the development of efficient computational tools to predict m^{6}Am sites has become a major focus of research, offering potential breakthroughs in RNA epigenetics. In this study, we present a robust and reliable deep learning model, DTC-m6Am, for identifying m^{6}Am sites across the transcriptome. **Methods**: Our proposed DTC-m6Am model first represents RNA sequences by One-Hot coding to capture base-based features and provide structured inputs for subsequent deep learning models. The model then combines densely connected convolutional networks (DenseNet) and temporal convolutional network (TCN). The DenseNet module leverages its dense connectivity property to effectively extract local features and enhance information flow, whereas the TCN module focuses on capturing global time series dependencies to enhance the modeling capability for long sequence features. To further optimize feature extraction, the Convolutional Block Attention Module (CBAM) is used to focus on key regions through spatial and channel attention mechanisms. Finally, a fully connected layer is used for the classification task to achieve accurate prediction of the m^{6}Am site. For the data imbalance problem, we use the focal loss function to balance the learning effect of positive and negative samples and improve the performance of the model on imbalanced data. **Results**: The deep learning-based DTC-m6Am model performs well on all evaluation metrics, achieving 87.8%, 50.3%, 69.1%, 41.1%, and 76.5% for sensitivity (Sn), specificity (Sp), accuracy (ACC), Mathew's correlation coefficient (MCC), and area under the curve (AUC), respectively, on the independent test set. **Conclusions**: We critically evaluated the performance of DTC-m6Am using 10-fold cross-validation and independent testing and compared it to existing methods. The MCC value of 41.1% was achieved when using the independent test, which is 19.7% higher than the current state-of-the-art prediction method, m6Aminer. The results indicate that the DTC-m6Am model has high accuracy and stability and is an effective tool for predicting m^{6}Am sites.

Keywords: m^{6}Am site identification; deep learning; DenseNet; TCN; CBAM; class imbalance

1. Introduction

RNA modifications are chemical alterations to RNA nucleotides that have profound effects on RNA structure and function. To date, more than 170 RNA modifications have been identified in all classes of RNA molecules [1]. In contrast, N6,2′-O-dimethyladenosine (m^{6}Am) is a widespread RNA modification that was first identified at the 50 ends of viral and animal cell mRNAs in 1975 [2] and catalyzed by the Phosphorylated CTD Interacting Factor 1 (PCIF1) [3]. A recent study suggests that m^{6}Am may play an important role in the pathogenesis of type 2 diabetes mellitus (T2DM), making it a future target for next-generation antidiabetic drug research [4].

As a dynamic and reversible epigenetic marker, m^{6}Am plays a variety of functional roles in disease research, viral infection, cancer biology, and other related fields. m^{6}Am modification plays an important role in gene expression regulation, mainly by affecting precursor mRNA splicing, mRNA stability, and translation efficiency [5]. In disease-related studies, m^{6}Am modification has been shown to be associated with weight regulation. For example, reduced m^{6}Am modification accompanied by weight loss was observed in a *PCIF1* knockout mouse model [6]. In addition, m^{6}Am-modified messenger ribonucleic acids (mRNAs) were enriched in a variety of metabolism-related processes in lean and obese mice [7], suggesting that m^{6}Am may be involved in the regulation of processes such as energy homeostasis and fat storage. m^{6}Am has likewise been shown to play a critical role in viral infections. The RNAs of certain viruses such as vesicular stomatitis virus (VSV), rabies virus (RABV), and human immunodeficiency virus (HIV) undergo m^{6}Am modifications, and these modifications can either enhance or inhibit viral gene expression, which in turn affects the susceptibility of host cells to viral invasion [8,9]. In particular, during SARS-CoV-2 infection, *PCIF1* promotes m^{6}Am modification of *ACE2* and *TMPRSS2* mRNAs, increasing

host cell susceptibility to entry of this virus [10], whereas HIV infection reduces m^6Am levels within about one-third of the modified genes [9]; in cancer biology, m^6Am modification also demonstrates a complex dual effect. On the one hand, it can promote tumor growth by stabilizing proto-oncogenes such as *Fos* mRNA; on the other hand, it can resist anti-PD-1 therapy by destabilizing *STAT1* and *IFITM3* mRNA [11]. Therefore, m^6Am modification not only plays an important role in the development of various cancer types, such as gastric and colorectal cancers, but may also be a potential target in future precision medicine strategies [11,12]. Regarding the identification techniques of the m^6Am site, early studies mainly relied on chemical analysis methods, which could confirm the presence of m^6Am but could not comprehensively resolve its distribution and dynamic changes in the transcriptome. With technological advances, high-throughput sequencing methods based on immunoprecipitation were gradually introduced, such as m^6A individual-nucleotide-resolution cross-linking and immunoprecipitation (miCLIP) [13] and m6ACE-seq [14]. These techniques have initially achieved the localization of the m^6Am site in the transcriptome by using anti-m^6A/m^6Am antibodies combined with RNA immunoprecipitation and high-throughput sequencing. However, due to the chemical similarity between m^6Am and the neighboring m^6A, these methods have difficulty distinguishing between the two. To address this problem, researchers have developed several m^6Am-specific recognition techniques, such as m6Am-seq [15] and m6Am-Exo-seq [3]. m6Am-seq achieves precise localization of m^6Am by optimizing the enzyme reaction conditions of FTO (Fat Mass and Obesity Associated Proteins), which selectively demethylates m^6Am without affecting m^6A. The m6Am-Exo-seq further improved the resolution of m^6Am-specific sites by removing RNA fragments not protected by cap structures through pretreatment. These methods significantly improved the resolution of m^6Am site identification and revealed the dynamics of m^6Am modifications and their roles in different biological contexts. Although continuous technological advances have greatly facilitated the development of m^6Am research, some limitations still exist. For example, many current methods rely on antibodies, whose specificity for m^6Am may be disturbed by other similar modifications. In addition, it is still difficult for existing techniques to distinguish the site distribution of m^6Am from that of m^6A in the region of high-density modifications.

With the deepening of m^6Am modification research, machine learning-based, especially deep learning, methods are playing an increasingly important role in m^6Am site identification. These methods can significantly improve the accuracy and efficiency of prediction through feature extraction and pattern analysis of large-scale, high-throughput sequencing data. In the field of RNA modification research, several deep learning methods have been applied to different types of modification site prediction. For example, Shaon *et al.* [16] proposed GRUpred-m5U combined gated recurrent unit (GRU) and convolutional neural networks (CNNs) to achieve excellent performance in m5U site prediction, while Zhao *et al.* [17] proposed Moss-m7G to utilized the Transformer structure to extract deep features from motif information of RNA sequences to achieve highly accurate m7G Site Prediction. These studies show that deep learning methods can effectively integrate RNA sequence information, structural features, and other epigenetic modification data to improve the comprehensiveness and accuracy of modification site identification. Compared with traditional methods, deep learning has significant advantages in processing complex and high-dimensional data and can automatically learn key features of RNA modifications, reducing the reliance on manual feature engineering. Therefore, deep learning has become an important tool for advancing m^6Am research, providing powerful computational support for in-depth exploration of RNA modifications. To date, related researchers have proposed a variety of computational methods based on traditional machine learning and deep learning to predict m^6Am sites. In 2021, Song *et al.* [18] developed MultiRM, a multi-label neural network approach based on the attention mechanism, which is not only capable of predicting m^6Am sites but also identifying a variety of other RNA modifications at the same time, which significantly extends the functional range of prediction tools. Subsequently, in 2022, Jiang *et al.* [19] proposed m6AmPred, a prediction tool based on the XGBoost algorithm and a hybrid electron-ion interaction potential (EIIP) and pseudo-EIIP (PseEIIP) coding strategy, marking the initial exploration of m^6Am site prediction. In the same year, Luo *et al.* [20] introduced DLm6Am, a deep learning tool that further improved the accuracy of m^6Am site identification by combining three sequence feature encoding schemes: one-hot, nucleotide chemical property (NCP), and nucleotide density (ND). In 2023, Jia *et al.* [21] proposed EMDL_m6Am, a prediction model based on stacked integrated deep learning. It used one-hot coding to express RNA sequence features and integrated different CNN models, demonstrating the potential of deep learning in m^6Am site prediction. By 2023, Liu *et al.* [22] published m6Aminer, a prediction tool based on the CatBoost algorithm, which incorporates a variety of sequence-derived features and provides a user-friendly web server, making m^6Am site prediction more convenient and practical. These research efforts not only demonstrate the trajectory of m^6Am site prediction techniques but also reflect the continuous optimization in feature extraction, model construction, and prediction accuracy. Despite the significant progress, there is still room for improving the accuracy and generalization ability of m^6Am site prediction. Both the EDMDL_m6Am and the DLm6Am use an integrated learning approach to improve prediction performance by building a multi-model architecture. However, this strategy requires parallel training of multiple base learners, leading to

a significant increase in computational resource consumption, especially when dealing with large-scale epitranscriptome data where the training cost is particularly prominent. Whereas m6Aminer demonstrated high sensitivity in m^6Am site prediction, there are still limitations in its ability to recognize negative samples. Existing models may result in high false-positive rates due to the underrepresentation of negative samples or limited feature differentiation, making it difficult to meet the demand for accurate identification in practical applications. Based on the above considerations, we propose an innovative hybrid neural network architecture, DTC-m6Am, which builds a multi-scale feature learning framework by deeply fusing densely connected convolutional networks (DenseNet), temporal convolutional network (TCN), and attention mechanisms (CBAMs). The model utilizes DenseNet to extract local semantic features of RNA sequences and passes their high-dimensional features to the TCN module through a cascade to achieve an efficient fusion of local features with global context. The TCN extends the temporal dependency perception range to capture long program column associations, and then CBAM dynamically filters the key modification features through the channel-space attention mechanism. Finally, the full connectivity layer integrates and nonlinearly maps the calibrated features, outputs the classification probability of m^6Am sites, and completes the accurate discrimination from sequence features to functional sites.

Second, we used various RNA sequence coding techniques, such as one-hot, nucleotide chemical property (NCP), and nucleotide density (ND), which were experimentally tested to show that the use of one-hot coding as the model input can propose the features of the sequences in a relatively simple and efficient way. The whole flowchart is displayed in Fig. 1. To address the class imbalance problem in the training set, we also used the focal loss function to control the training process, a practice that effectively prevents training from being biased toward the majority class. Finally, we conducted ablation experiments to verify the effectiveness of each module in the model. The code is available in the GitHub repository (https://github.com/hhui0/DTC-m6Am).

2. Materials and Methods

2.1 Benchmark Dataset

The benchmark dataset is derived from the recently published single nucleotide resolution m^6Am sequencing data [3,13], which was used by Jiang *et al.* [19] to construct the m6Ampred machine learning classifier; however, they do not remove redundant sequences from the benchmark dataset, which may result in similar or duplicate sequences potentially dominating the training process, causing the model to rely on repetitive patterns rather than learning a wider, more robust set of features. Therefore, to remove sequence redundancy, Liu *et al.* [22] used CD-HIT [23] (a fast clustering tool based on short-word filtering) to filter the baseline dataset with a strict threshold of 0.8, and then efficiently clustered the similar sequences and retained the representative sequences through short-word frequency analysis, which significantly reduced the sequence redundancy in the dataset. After the above operation, the training dataset was allocated with 3700 positive samples and 37,000 negative samples, and the ratio of positive samples to negative samples was 1:10. Whereas the independent test dataset consisted of 320 positive samples and 320 negative samples, all of which are 41 nt in length.

Table 1. Distribution of the benchmark data set.

Dataset	Positive	Negative
Training	3700	37,000
Independent	320	320

To evaluate DTC-m6Am and compare it with other predictors, we used the benchmark and independent datasets adopted by Liu *et al.* [22]. In this study, the unbalanced training data are all used as the training set to ensure that the features in the samples are fully extracted, and the dataset size is shown in Table 1.

2.2 Feature Extraction Methods

Feature coding technology is an important part of deep learning training, and choosing the appropriate sequence coding method is crucial for the recognition of m^6Am sites. In this paper, we used three of the more pervasive and widespread RNA coding techniques in deep learning, including one-hot, NCP, and ND, and experimented with the combination of the three coding techniques.

2.2.1 One-Hot Encoding

One-hot coding is a commonly used feature coding method, which is mainly used to convert category data into numerical data for easy processing by machine learning models. In one-hot coding, each category is represented as a binary vector, the length of the vector is equal to the total number of categories, and only one position in the vector is 1 (representing the category), and the other positions are 0. The mRNA contains four different nucleotides, A, U, C, and G, and their one-hot coding is as follows:

$$\begin{aligned} A &\rightarrow (1,0,0,0) \\ U &\rightarrow (0,1,0,0) \\ G &\rightarrow (0,0,1,0) \\ C &\rightarrow (0,0,0,1) \end{aligned} \tag{1}$$

In this study, after one-hot coding each sequence is transformed into a 41×4 feature matrix, this coding has the advantage of avoiding the numerical order relationship between the categories, so that the model will not misinterpret the size relationship between the categories.

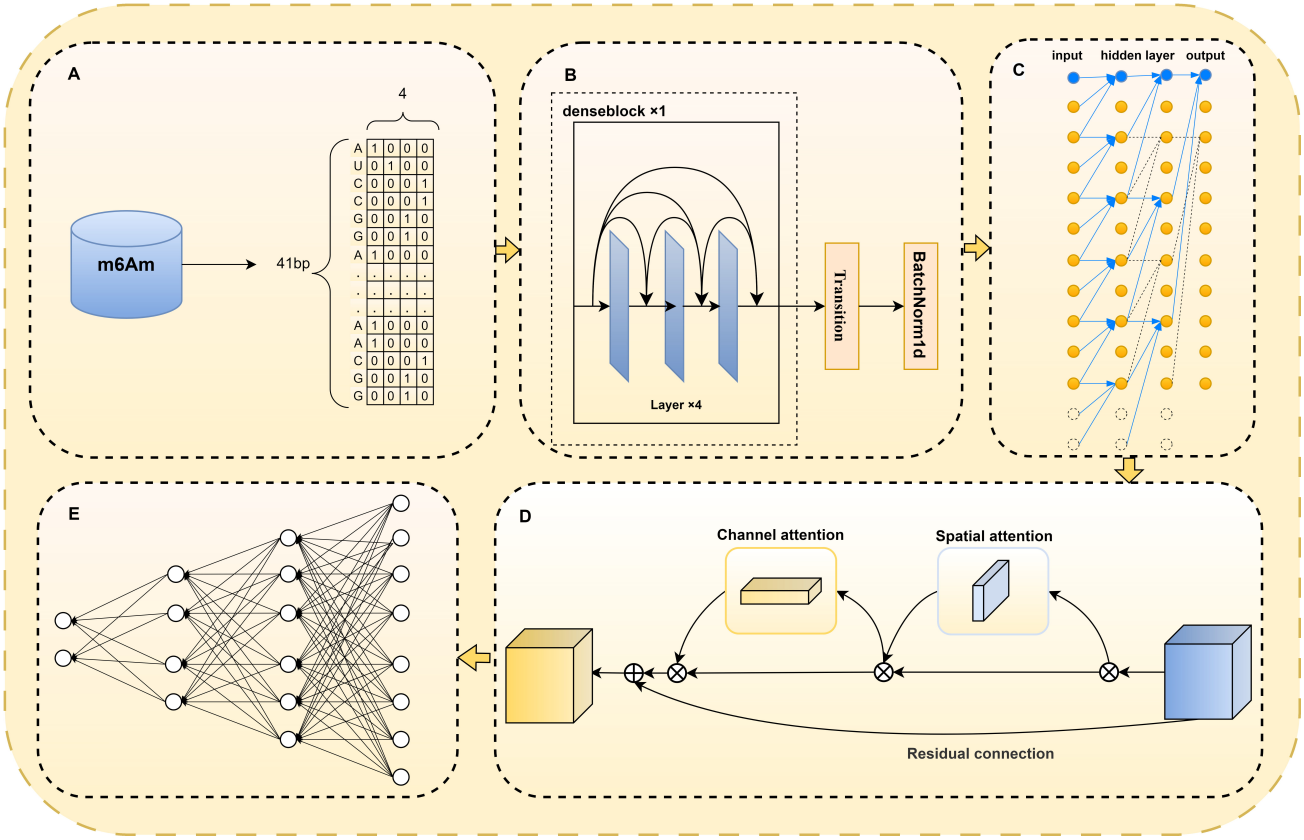

Fig. 1. An overview of the DTC-m6Am model. (A) Data preprocessing: RNA sequences (41 bp) were one-hot coded and converted to a model-acceptable input format. (B) Feature extraction (DenseNet): local feature extraction was performed using the DenseNet structure, consisting of a DenseBlock, Transition and BatchNorm1d layers to optimize feature flow and reduce computational cost. (C) Temporal Convolutional Network (TCN): sequence modeling via temporal convolutional networks to extract deep features. (D) Convolutional Block Attention Module (CBAM): combines Channel Attention and Spatial Attention to enhance key features and uses residual connectivity to optimize information flow. (E) Classification Layer: the fully connected layer classifies the extracted features and finally outputs the prediction results. Created using draw.io.

2.2.2 NCP Encoding

Nucleotide Chemical Property encoding, proposed by Bari *et al.* [24], is a widely used feature extraction method in computational biology studies of RNA and DNA sequences, based on the chemical properties of bases [25,26]. Its core idea is to transform each nucleotide (A, U, G, C) into a biologically meaningful numerical vector reflecting its chemical properties. These properties typically include the number of hydrogen bonds, polarity, molecular volume, and other physicochemical attributes, thus providing a richer representation of features for sequence analysis tasks. In NCP encoding, each nucleotide is mapped to a vector based on its chemical properties, where each dimension corresponds to a specific chemical characteristic. Purines (A, G) and pyrimidines (C, U) are both ring-containing compounds, where purines contain two rings while pyrimidines have only one ring. Therefore, they can be attributed to the same x-coordinate position. Similarly, amino (A, C) and keto (G, U) groups can be attributed to the same y-coordinate position because they have the same chemical functionality. Finally, the z-coordinate is determined based on the strength of the hydrogen bonds: strong (C, G) and weak (A, U). Specifically, the i-th nucleotide of an RNA sequence of length L can be represented by the vector $P_i = (X_i, Y_i, Z_i)$ where $i = 1, 2, 3, \cdots, L$. In this study $L = 41$, if X denotes the ring structure, Y denotes the hydrogen bond, and Z denotes the chemical functional group, the sequence is transformed into a 41×3 feature matrix after NCP encoding, and the NCP feature expression and calculation formula are as follows:

$$\begin{cases} X_i = \begin{cases} 1, & \text{if } P_i \in A, G \\ 0, & \text{if } P_i \in C, U \end{cases} \\ Y_i = \begin{cases} 1, & \text{if } P_i \in A, U \\ 0, & \text{if } P_i \in C, G \end{cases} \\ Z_i = \begin{cases} 1, & \text{if } P_i \in A, C \\ 0, & \text{if } P_i \in G, U \end{cases} \end{cases} \tag{2}$$

2.2.3 ND Encoding

Nucleotide Density Encoding (ND Encoding) is a simple and effective method of sequence feature representation, which transforms a sequence into a numerical feature by calculating the cumulative frequency distribution of nucleotides (A, U, C, G) in the sequence. Specifically, for a sequence $R_1R_2R_3\cdots R_L$ of length L, the ND encoding d_i at position i is calculated as:

$$d_i = \frac{1}{i}\sum_{j=1}^{i} f\left(R_j\right), f\left(R_j\right) = \begin{cases} 1, & \text{if } R_j = R_i \\ 0, & \text{otherwise} \end{cases}, i = 1, \ldots, L \tag{3}$$

For example, for the sequence "ATCGA", there is an "A" nucleotide at the fifth position, and there are five nucleotides from the beginning to that position, of which there is two "A", so the ND value of that position is 2/5 = 0.4. A sequence of length 41 will be converted into a 41×1 feature matrix after ND coding. In the field of DNA/RNA modification site prediction, ND coding combined with machine learning models has demonstrated high prediction performance. Site prediction field, ND coding combined with machine learning models, has shown high prediction performance. Several studies [27,28] have shown that using ND coding in combination with other coding methods can significantly improve prediction accuracy.

2.3 Classification Model

Choosing the right model is crucial for predicting the m^6Am site. We used an innovative hybrid neural network that combines the efficient feature reuse capability of DenseNet, the long-range dependency modeling advantage of Temporal Convolutional Network (TCN) for sequence data, and the adaptive attention capability of Convolutional Block Attention Module (CBAM) for important features. By integrating these three mechanisms, the model is able to extract and understand the multidimensional features of RNA sequences more comprehensively, improving the accuracy and generalization of the predictions.

2.3.1 DenseNet

Densely Connected Convolutional Networks (DenseNet) [29] is a popular deep learning architecture that demonstrates powerful feature extraction capabilities with its unique dense connectivity mechanism. Wang *et al*. [30] focused on the prediction of lysine succinylation sites and proposed MDCAN-Lys, a multi-pathway deep learning framework based on DenseNet and CBAM. DenseNet enhances the interaction between high and low-level features through the dense connectivity mechanism, effectively reduces information loss, and optimizes the ability to focus on key regions for feature extraction by combining the CBAM module. On the other hand, Jia *et al.* [31] proposed a deep learning framework for the recognition of 5mC modification sites in DNA by combining the improved DenseNet, bi-directional GRU (BiGRU), and self-attention mechanism in the study of i5mC-DCGA. DenseNet is mainly used to extract local features efficiently in this framework, and the number of layers and growth rate of different DenseBlock will directly affect the number of model parameters, feature reuse efficiency and gradient propagation effect, which will lead to the difference in computational performance and generalization ability.

The phenomenon that the training error increases instead as the network deepens is called the degradation problem of deep learning. To solve this problem, ResNet [32] adds a skip connection to bypass the nonlinear transformation with an identity function:

$$x_l = H_l\left(x_{l-1}\right) + x_{l-1} \tag{4}$$

The jump connection provides a path for the gradient without nonlinear activation and prevents the gradient from disappearing in backpropagation, thus improving the training efficiency and stability of the deep network. On the basis of ResNet, in order to improve the feature information extraction ability, DenseNet uses a dense connection method for feature multiplexing, as shown in Fig. 2, for the output x_l of layer l can be expressed as:

$$x_l = H_l\left(\left[x_0, x_1, \ldots, x_{l-1}\right]\right) \tag{5}$$

That is, from the input $[x_0, x_1, \ldots, x_{l-1}]$ of the previous $1-l$ layer through $H_l(\cdot)$ can be obtained from the output of the lth layer, x_l, where $H_l(\cdot)$ can be expressed as Batch Normalization (BN) [33], rectified linear units (ReLU) [34], Pooling [35], or Convolution. The characteristic of DenseNet dense connection makes the feature information fully reused in the network, which can effectively alleviate the problem of gradient vanishing and, at the same time, reduce the network parameters.

2.3.2 Temporal Convolutional Networks

TCN [36] is a neural network architecture for processing sequential data designed with the goal of efficiently modeling long-range dependencies in time series. TCN is based on a one-dimensional convolutional operation that combines causal convolution [37] and dilated convolution [38] to achieve efficient capture of time-dependent properties.

Causal convolution is a convolution operation that is mainly used in sequence modeling to ensure that the output at the current moment depends only on the current and previous inputs to avoid future information leakage. The output y_t for time t can be calculated by the following equation:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-i} + b \tag{6}$$

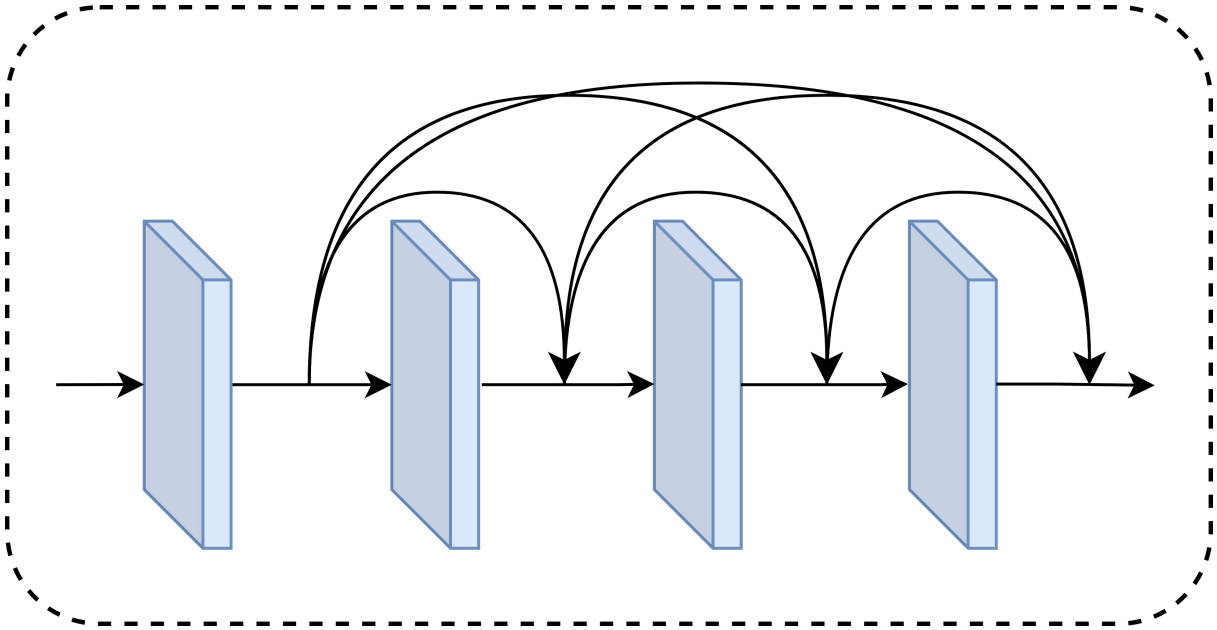

Fig. 2. The structure of a dense block. The inputs of each layer are integrated by channel splicing (Concat) to integrate the output feature maps of all previous layers, and 1×1 convolutional dimensionality reduction (Bottleneck layer) is used within dense blocks to reduce the number of parameters. Created using draw.io.

Where w_i denotes the weight of the convolution kernel, b is the bias, k is the filter size, and $t-i$ denotes the past direction. TCN adds dilated convolution to this by using a larger dilation to make the output of the top layer representative of a wider range of inputs. For the output Y_t at time t, the dilatation coefficient d is introduced over y_t, and mathematically, the expression is:

$$Y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-d \cdot i} + b \tag{7}$$

Through the design of dilated convolution, TCN is able to capture long-range dependencies in a shallower network structure, which is suitable for processing sequence data with long time spans. Furthermore, as shown in Fig. 3, combined with residual connectivity [32], TCN effectively mitigates the gradient vanishing and gradient explosion problems, making the training of deep networks more stable. What is more, TCN can process data at all time steps in parallel, which significantly improves the speed of training and inference compared to Recurrent Neural Network (RNN).

2.3.3 CBAM Attention

Convolutional Block Attention Module [39] (CBAM) is a lightweight and efficient attention module designed to enhance the neural network's attention to important features while suppressing irrelevant or distracting information through channel and spatial dimensional attention mechanisms. It mainly consists of two parts, the Channel Attention Module and the Spatial Attention Module, which augment the input features in sequential order.

The channel attention module is shown in Fig. 4, where CBAM first compresses the input features in the spatial dimension by global average pooling and global maximum pooling to generate two-channel feature vectors describing the global information. Subsequently, these two vectors are fed into a two-layer fully connected network with shared weights to learn the importance of different channels. After fusion by weighting, the generated channel weight vectors are used to realign the importance of each channel of the Input feature map. Specifically, for Input feature F, a channel attention function $M_c \in R^{c \times 1 \times 1}$ is defined with the following formula:

$$\begin{aligned} M_c(F) &= \sigma(\text{MLP}(\text{Avgpool}(F)) + MLP(\text{MaxPool}(F))) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\text{max}}\right)\right)\right) \end{aligned} \tag{8}$$

Where σ is the sigmoid function, F^c_{avg}, F^c_{max} denote the average pooling and maximum pooling, respectively, and it is worth noting that W_0 and W_1 are shared by two inputs.

The spatial attention module is shown in Fig. 5. After compressing the feature maps in the channel dimension, the spatial attention module generates two feature maps by max-pooling and mean-pooling, which are spliced together and undergo a convolution operation to obtain the spatial attention weight map. This weight map is used to emphasize the saliency of a particular spatial location. Specifically, for Input feature F, a spatial attention function $M_s \in R^{H \times W}$ is defined with the following formula:

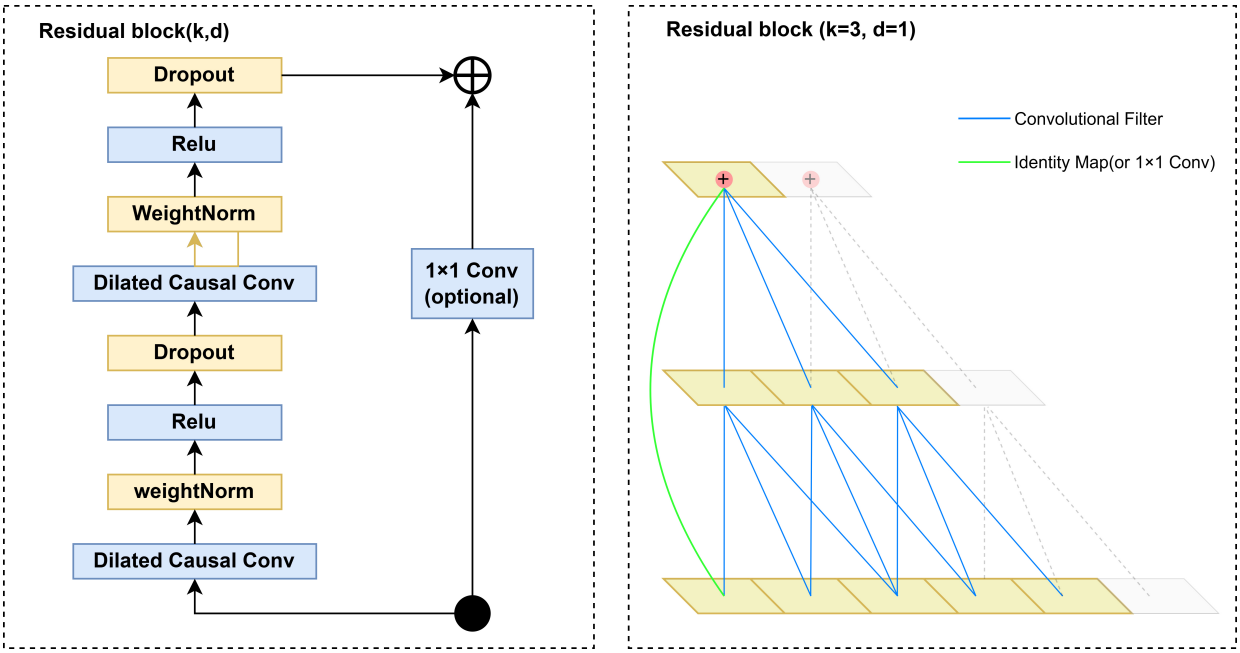

Fig. 3. TCN residual block and an example of residual connection in a TCN. Created using draw.io.

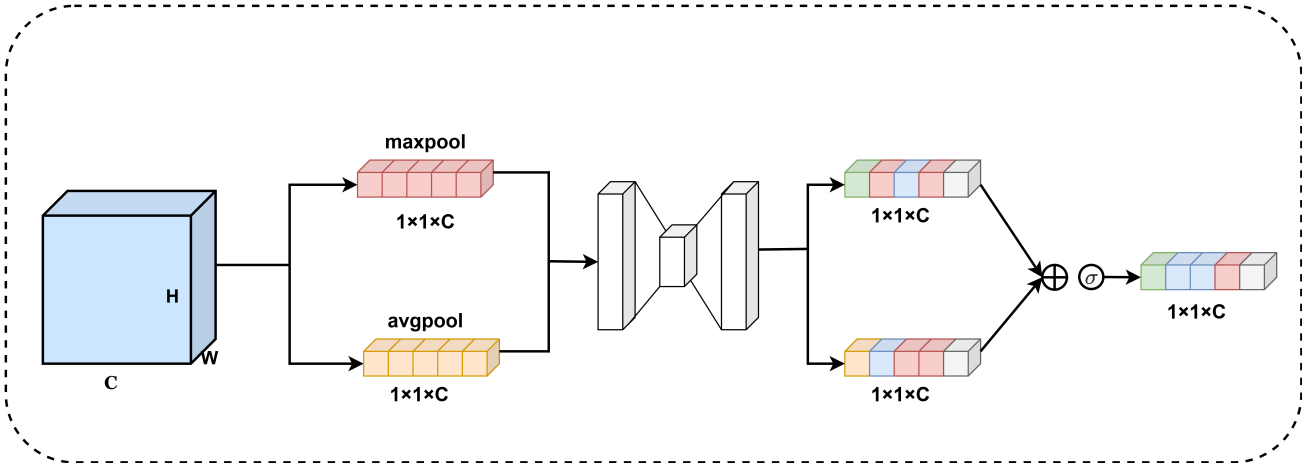

Fig. 4. Channal attention module structure. Created using draw.io.

$$M(F) = \sigma\left(f^{7\times7}([\text{Avgpool}(F); \text{MaxPool}(F)])\right) = \sigma\left(f^{7\times7}\left(\left[F^s_{\text{avg}}; F^s_{\text{max}}\right]\right)\right) \tag{9}$$

Where $f^{7\times7}$ denotes a convolution operation with a filter size of 7×7. By combining the channel and spatial attention mechanisms, CBAM can select key features more effectively and improve the network's understanding of the target task.

2.4 Focal Loss Function

Focal loss function is a loss function designed to solve the problem of category imbalance proposed by Lin *et al.* [40] in 2018. The traditional cross-entropy loss is prone to being dominated by the majority class when dealing with datasets with extremely uneven class distribution, which makes it difficult for the model to focus on hard-to-categorize samples. Focal loss effectively reduces the weights on easy-to-categorize samples by introducing a modulating factor and a focusing parameter in the loss function, which effectively reduces the weight of easy-to-classify samples and makes the model focus more on the learning of difficult samples, its formula is as follows:

$$FL\left(p_t\right) = -\alpha_t\left(1-p_t\right)^{\gamma}\log\left(p_t\right) \tag{10}$$

Where $p_t \in [0,1]$ indicates the probability that the predicted sample belongs to 1, α_t is called the weighting factor, which serves to control the shared weight of the positive and negative samples to the total loss, and the smaller the value represents the lower the weight of the negative sam-

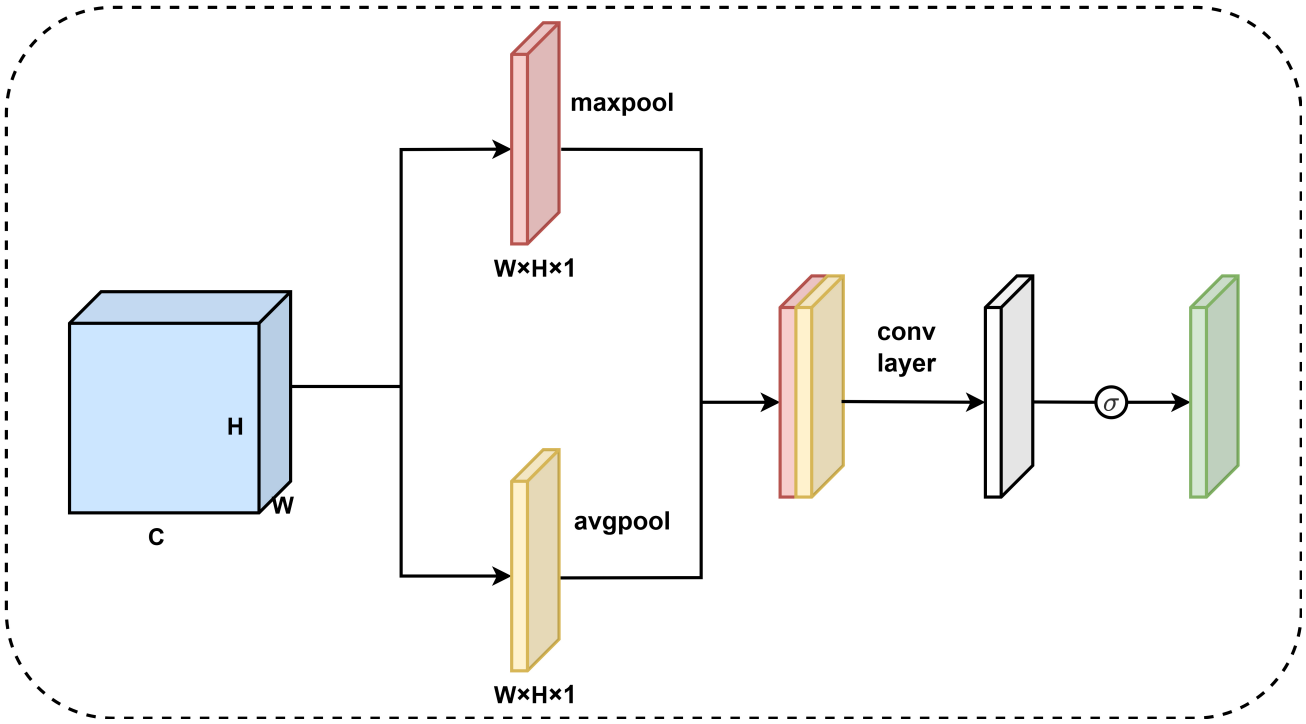

Fig. 5. Spatial attention module structure. Created using draw.io.

ples, γ is called the focusing Parameter, whose role is to make the model more focused on the hard-to-classify samples by reducing the weight of the easy-to-classify samples, thus making the model more focused on the hard-to-classify samples during the training. It is to reduce the weight of easy-to-categorize samples, thus making the model more focused on difficult-to-categorize samples during training.

2.5 Performance Evaluation

In this model evaluation, we used six indicators: accuracy (ACC), sensitivity (Sn), specificity (Sp), F1 score, Matthews correlation coefficient (MCC), and area under the ROC curve (AUC). These metrics can comprehensively evaluate the performance of the model from different perspectives to ensure the comprehensiveness and validity of the assessment. In particular, the MCC is more comprehensive and reliable than accuracy and F1 score in the category imbalance dataset, and it is suitable for dichotomous classification problems that need to accurately identify positive and negative class samples at the same time and it has performed well in the field of genomics [41]. Therefore, the use of these indicators can effectively assess the comprehensive performance of the model and ensure the rationality and validity of the evaluation results. The specific formulas of the six indicators are as follows:

$$\begin{cases} Sn = \frac{TP}{TP+TN} \\ Sp = \frac{TN}{TN+FP} \\ Acc = \frac{TP+TN}{TP+TN+FP+FN} \\ F1 = \frac{2\times TP}{2\times TP+FP+FN} \\ Mcc = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \end{cases} \tag{11}$$

Where TP, FP, TN, and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively, and the sensitivity and specificity are calculated using a threshold of 0.5. AUC, as a commonly used metric, effectively eliminates the effect of category imbalance, and the higher the value, the better the model performance is, and an AUC of 1 denotes a perfect classifier, and 0.5 denotes a random classifier.

2.6 Instructions for Setting Hyperparameters

To ensure that the training results of the DTC-m6Am model can be fairly compared with existing studies, we trained it on the same dataset and used NVIDIA GeForce RTX 4080 GPU (NVIDIA Corporation, Santa Clara, CA, USA) to accelerate the computation. During the training process, we chose AdamW as the optimizer to improve the stability of the gradient update and prevent the loss function from falling into a local optimum. At the same time, we use the focal loss function to adjust the gradient propagation to better handle the category imbalance problem. To avoid model overfitting, we combine regularization, discard and early stopping strategies. In addition, we optimize

Table 2. Description of the hyperparameters in the DTC-m6Am model.

Parameters	Number
Dense block	1
Convolution layer number of a dense block	4
Convolution kernel size	3
Growth rate	30
Dilation rate	2
Dropout ratio	0.1

the hyperparameter configuration through comparative experiments to obtain the best performance. The whole model implementation relies on Python 3.10 (Python Software Foundation, Wilmington, DE, USA) and PyTorch 2.0.0 + cu11.8 (Meta AI, Menlo Park, CA, USA), and the specific hyperparameter settings are shown in Table 2.

3. Results and Discussion

3.1 Contrasting Various Feature Extraction Techniques

This section explores in detail the effects of different coding styles on the experimental results, using One-Hot coding, ND coding, NCP coding, and various combinations of these coding styles, with the aim of evaluating their ability to express the features and their impact on the model performance when processing the data. In order to comprehensively compare the effects of these coding approaches, we conducted 10-fold cross-validation and independent tests on the dataset. The experimental results show that One-Hot coding is the most balanced and stable in terms of overall performance, and the results of both cross-validation and independent tests are slightly better than the other coding approaches, as shown in Fig. 6 and Table 3. However, it is worth noting that on individual metrics (e.g., Sp), other coding methods (e.g., ND and NCP) sometimes show some advantages, but in this experiment, the training process of the model is affected by the problem of extreme imbalance between positive and negative samples in the training set, and the ND and NCP coding may be limited by the redundancy of features and the tendency of overfitting, which leads to the bias of its training process to learn more information about negative samples resulting in insufficient feature extraction for the feature extraction of positive samples is insufficient. In contrast, One-Hot coding, with its comprehensive feature expression capability, can provide a better learning basis for the model, thus achieving a more comprehensive performance. The results of this study provide a strong rationale for the choice of the encoding method, suggesting that One-Hot encoding is a reliable choice for the task at hand, although it may need to be optimized in combination with other encoding methods in some specific scenarios.

Table 3. Ablation studies of the feature encoding ways on the independent testing dataset.

One-hot	√		√	√		√
NCP		√	√		√	√
ND				√	√	√
Sn	**0.878**	0.856	0.837	0.873	0.794	0.856
Sp	0.503	0.516	0.553	0.484	0.578	0.506
ACC	0.691	0.686	**0.695**	0.681	0.686	0.681
MCC	**0.411**	0.396	0.407	0.394	0.381	0.387
AUC	0.765	0.763	0.761	0.742	0.752	**0.766**
F1	**0.739**	0.732	0.733	0.734	0.717	0.729

The symbol "√" indicates that this module is included in the model of the experiment. NCP, nucleotide chemical property; ND, nucleotide density; ACC, accuracy; MCC, Matthews correlation coefficient; AUC, area under the curve. The bold font is used to distinctly indicate the highest values achieved for each evaluation metric.

3.2 Analysis of Model Structure

In order to explore the effect of the number of DenseBlock in the DenseNet module on the model performance, we selected the number of DenseBlock as 1, 2, 3, and 4 to conduct the experiments, respectively. The experimental results are shown in Fig. 7, where the DenseBlock number of 1 performs best in several indicators, among which MCC and AUC reach 0.541 and 0.910, respectively, showing strong overall prediction ability. As the number of DenseBlock increases, the MCC metric, which reflects the overall performance of the model, is relatively low. This suggests that too many DenseBlocks may lead to an increase in model complexity without effectively extracting additional useful features, instead introducing noise or overfitting problems. In the current task, the 1-block DenseNet structure can better balance the prediction demand of positive and negative class samples, which is suitable for model accuracy optimization.

In addition, we conducted comparative experiments between the CBAM attention mechanism and the Multi-head self-attention (MHSA) [42] and Squeeze-and-Excitation Networks (SENets) [43] attention mechanisms to evaluate their impact on model performance. According to the experimental results in Table 4, CBAM outperforms the other two attention mechanisms in several metrics, especially achieving higher values in MCC and AUC. This indicates that CBAM is able to capture key feature information more efficiently while reducing the interference of irrelevant or redundant features, thus improving the overall prediction ability of the model. In contrast, MHSA, while modeling longer-range dependencies, may introduce too much noise in the current task, while SENet focuses primarily on channel attention and fails to fully utilize spatial information. Therefore, in this study, CBAM is more suitable for optimizing the performance of the m^6Am modification site prediction model.

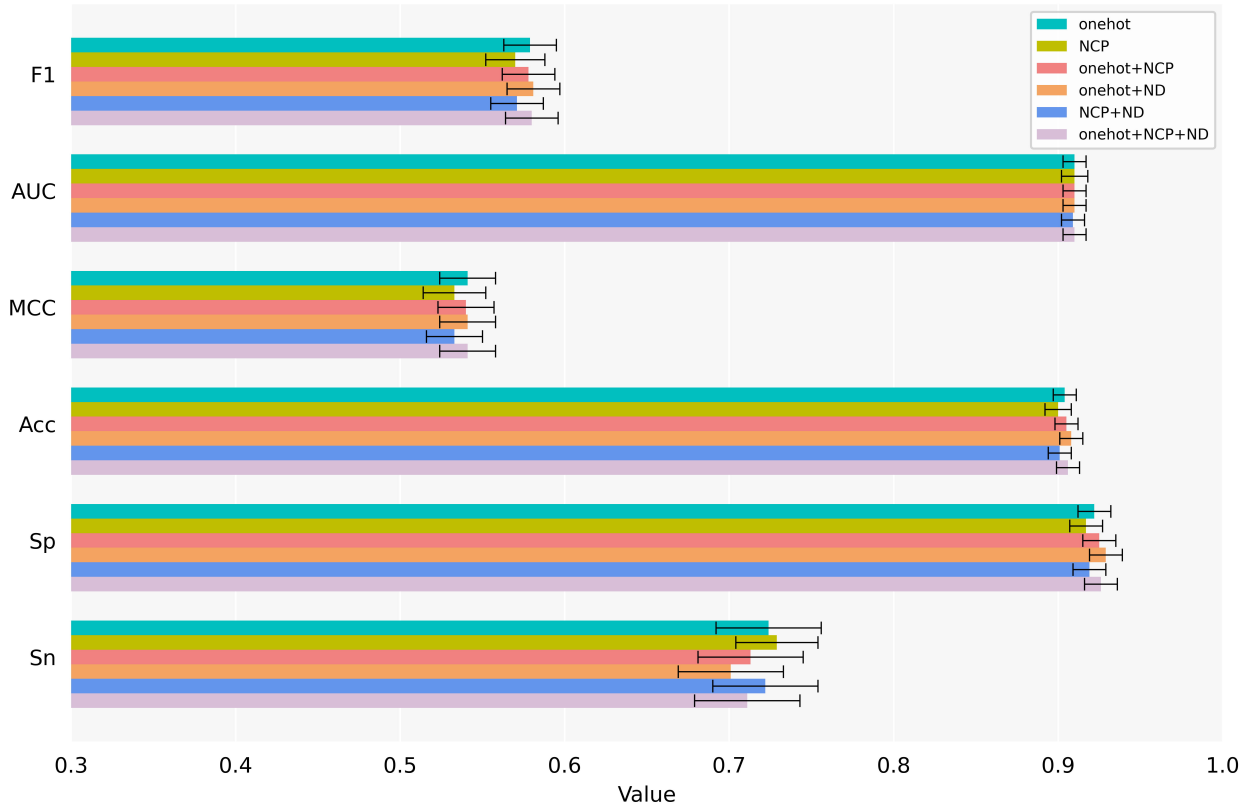

Fig. 6. Performance comparison of One-hot encoding with NCP encoding and ND encoding on the training dataset. The independent feature representation of One-Hot coding maintains the linear divisibility of the feature space and presents a relatively balanced performance between specificity (Sp) and sensitivity (Sn) metrics. ND, NCP coding resulted in a slight increase in Sp but a significant decrease in Sn due to the interference of feature redundancy. NCP, nucleotide chemical property; ND, nucleotide density; ACC, accuracy; MCC, matthews correlation coefficient; AUC, area under the curve. Generated using Python, specifically with the matplotlib and seaborn libraries.

Table 4. The performance of models with different attention mechanisms.

Attention methods	ACC	AUC	Sn	Sp	F1	MCC
MHSA	0.903 ± 0.008	0.905 ± 0.008	0.704 ± 0.030	**0.923 ± 0.011**	0.571 ± 0.018	0.531 ± 0.019
SENets	0.903 ± 0.010	0.907 ± 0.007	0.714 ± 0.035	0.922 ± 0.013	0.572 ± 0.021	0.534 ± 0.022
CBAM	**0.904 ± 0.007**	**0.910 ± 0.007**	**0.724 ± 0.032**	0.922 ± 0.010	**0.579 ± 0.016**	**0.541 ± 0.017**

Each value in the table is expressed as mean ± SD. where the mean is the average of the 10-fold cross-validation results and SD is their standard deviation. The best experimental results are highlighted in bold. MHSA, Multi-head self-attention; SENets, Squeeze-and-Excitation Networks; CBAM, Convolutional Block Attention Module.

3.3 Model Architecture Ablation Experiment

In order to verify the role played by each module in the overall performance of the model, we conducted ablation experiments in 10-fold cross-validation as well as independent tests, the results of which are shown in Tables 5,6. First, as can be seen visually from Fig. 8, the DenseNet module performs well in negative class identification with specificity (Sp) of 0.962, which is much higher than the other module combinations, indicating that DenseNet has an advantage in correctly identifying negative samples. This may be due to the fact that the deep structure of DenseNet is able to capture more complex features and, thus, is more accurate in the distinction of negative samples. On the other hand, the role of the TCN module in positive example recognition cannot be ignored. From Table 5, we can see that the sensitivity (Sn) of the combined model after the introduction of the TCN module reaches 0.710, indicating that the TCN is very effective in identifying positive samples. This property of the TCN may be related to its ability to capture long-term dependencies when dealing with sequence data.

When considering the combination of modules, the DenseNet + CBAM configuration shows excellent performance in ten-fold cross-validation, especially in terms of accuracy (ACC) and Matthews correlation coefficient (MCC). However, when we compare this combination with

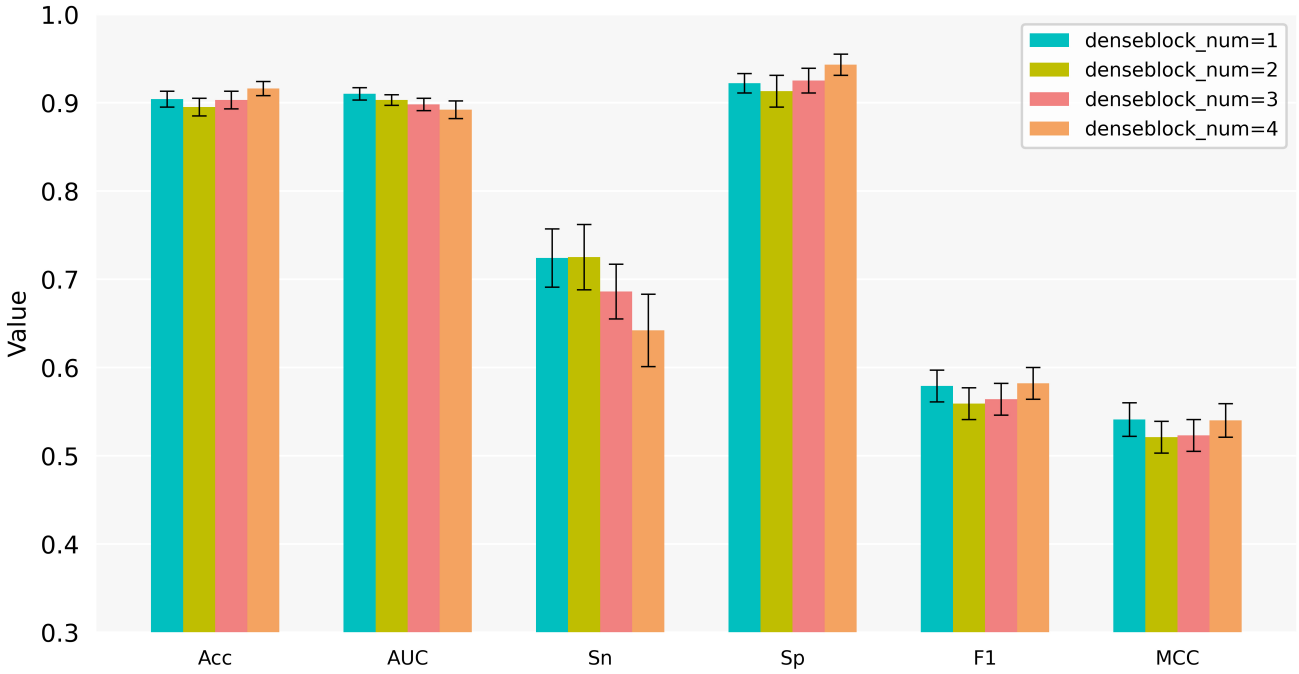

Fig. 7. The performance of models with different numbers of dense blocks. This figure shows the 10-fold cross-validation results on six metrics (ACC, AUC, Sn, Sp, F1, and MCC) for different number of DenseBlocks (1, 2, 3, and 4), with the bars representing the mean of the corresponding metrics, and the error line indicating the standard deviation. Generated using Python, specifically with the matplotlib and seaborn libraries.

the combination of all three modules (all), we find that the latter performs better in terms of AUC and Sn scores. In an independent test, the all combination achieved Sn, MCC, and F1 scores of 0.878, 0.411, and 0.739, which are the highest of all combinations.

Taken together, the combination of DenseNet, TCN, and CBAM can give full play to their respective advantages, enabling the model to achieve a better balance between sensitivity (Sn) and specificity (Sp). Specifically, DenseNet extracts complex features through the deep network structure, which improves the recognition ability of negative samples and excels in specificity; TCN enhances the recognition ability of positive samples by capturing long-range dependency information and improves the model's sensitivity; and CBAM further optimizes the feature extraction process, which enables the model to focus on key features more effectively, thus improving the overall prediction performance. The experimental results show that the model combining the three models has the best performance in the comprehensive indexes such as MCC and F1, indicating that it is more balanced and accurate in distinguishing between positive and negative samples. Therefore, DenseNet + TCN + CBAM was finally selected as the optimal configuration to ensure that the model has stronger prediction ability and more stable performance in recognizing m^6Am modification sites.

3.4 Comparison of Two Balancing Strategies

In Focal Loss, alpha and gamma are two key hyperparameters that play a crucial role in the model training process. Since the settings of these two parameters directly affect the model's learning of different types of samples, we first conducted a comparison of the model performance under different alpha and gamma parameter settings to explore the best hyperparameter combination. In this test, we systematically tested the parameter combinations of weighting factor α (0.2, 0.4, 0.6, 0.8) and focusing Parameter γ (0, 1, 2, 3) by grid search method, and chose the MCC value as the key index for selection. The MCC value under each parameter combination is shown in Fig. 9. Through our experiments, we found that the model performs best in the comprehensive evaluation metric MCC when $\gamma = 2$ and $\alpha = 0.8$, and the MCC value reaches 0.541 in the 10-fold cross-validation. Choosing $\alpha = 0.8$ reduces the weight on the dominant negative samples, thus alleviating the category imbalance problem, while $\gamma = 2$ enhances the focus on the difficult-to-categorize samples, making the model more focused on the positive samples, which enhances the sensitivity and improves the overall prediction performance. Therefore, this parameter combination can effectively balance the model's focus on difficult-to-classify samples and the weights between categories and significantly improves the model's performance on the unbalanced performance on the dataset.

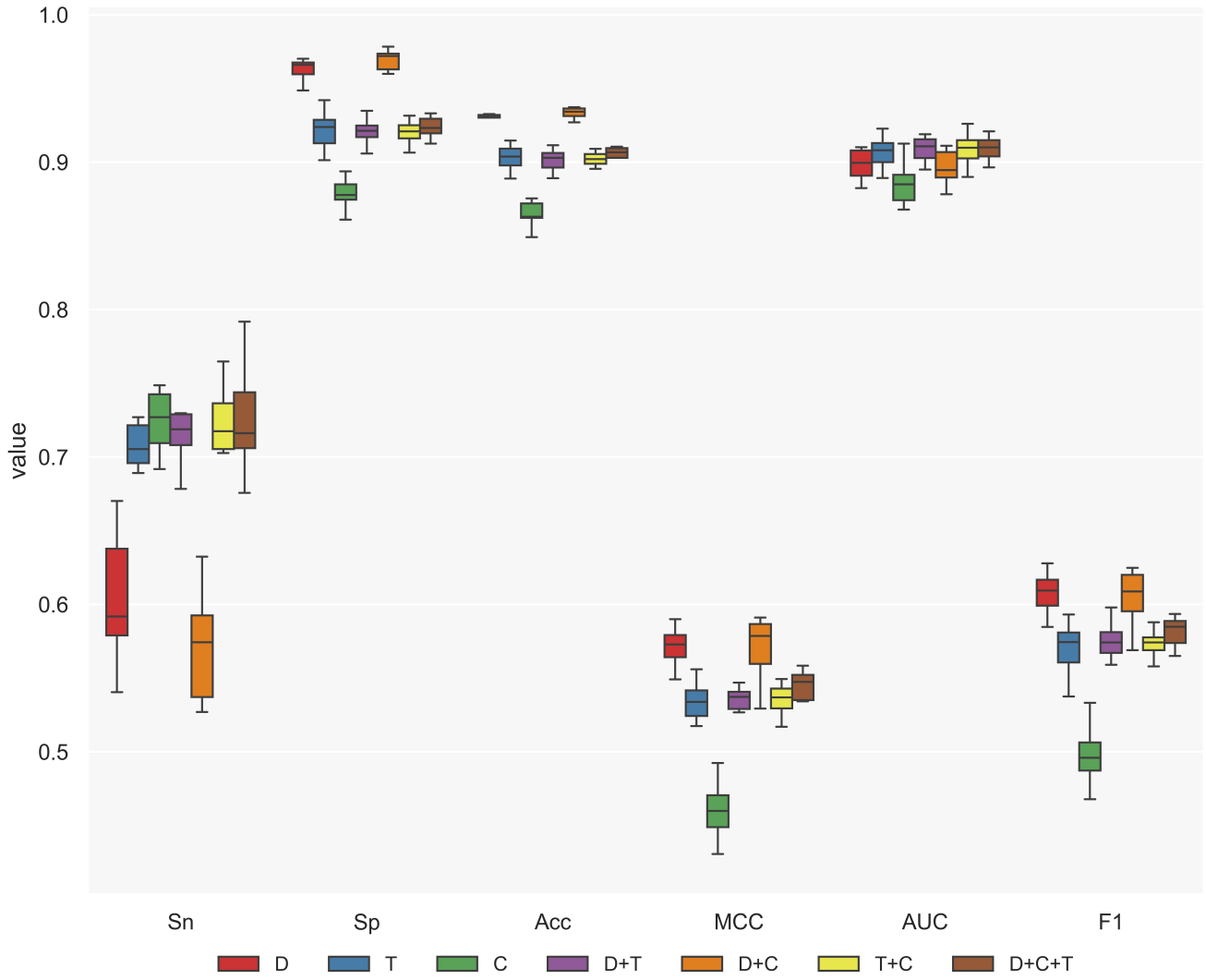

Fig. 8. Ablation results of DTC-m6Am on the training dataset. Generated using Python, specifically with the matplotlib and seaborn libraries.

Table 5. Ablation experiments results for DTC-m6Am.

Module combination	ACC	AUC	Sn	Sp	F1	MCC
DenseNet	0.930 ± 0.004	0.899 ± 0.010	0.602 ± 0.040	0.962 ± 0.008	**0.608 ± 0.012**	0.571 ± 0.012
TCN	0.903 ± 0.008	0.907 ± 0.010	0.710 ± 0.035	0.922 ± 0.012	0.571 ± 0.015	0.532 ± 0.016
CBAM	0.867 ± 0.010	0.885 ± 0.013	0.721 ± 0.019	0.881 ± 0.012	0.498 ± 0.017	0.460 ± 0.017
DenseNet + TCN	0.902 ± 0.007	0.909 ± 0.008	0.722 ± 0.030	0.920 ± 0.010	0.573 ± 0.015	0.535 ± 0.015
DenseNet + CBAM	**0.931 ± 0.010**	0.897 ± 0.010	0.581 ± 0.030	**0.966 ± 0.016**	0.605 ± 0.017	**0.572 ± 0.019**
TCN + CBAM	0.902 ± 0.004	0.909 ± 0.010	0.722 ± 0.034	0.920 ± 0.007	0.573 ± 0.008	0.535 ± 0.110
All	0.904 ± 0.007	**0.910 ± 0.007**	**0.724 ± 0.032**	0.922 ± 0.010	0.579 ± 0.016	0.541 ± 0.017

Each value in the table is expressed as mean ± SD. where the mean is the average of the 10-fold cross-validation results and SD is their standard deviation. The best experimental results are highlighted in bold.

In addition, to demonstrate the advantages of Focal Loss, we also explore the performance of the model under the downsampling strategy versus Focal Loss. Downsampling is usually used to solve the category imbalance problem by reducing the number of majority class samples to balance the category distribution. In this experiment, we adopt the downsampling strategy consistent with Liu *et al.* [22]. The experimental results are shown in Table 7. By comparing the experiments, we find that the model with Focal Loss retains the diversity of the data better than the traditional downsampling method and significantly improves the model's prediction ability for the minority class samples. However, downsampling may lead to information loss, which in turn affects the performance of the model.

Table 6. Ablation studies of the models on the independent testing dataset.

DenseNet	√			√	√		√
TCN		√		√		√	√
CBAM			√		√	√	√
Sn	0.712	0.831	0.834	0.828	0.831	0.859	**0.878**
Sp	**0.678**	0.533	0.512	0.534	0.550	0.487	0.503
ACC	**0.695**	0.683	0.673	0.681	0.690	0.673	0.691
MCC	0.391	0.383	0.366	0.379	0.397	0.374	**0.411**
AUC	0.771	0.738	0.76	0.732	**0.778**	0.735	0.765
F1	0.700	0.724	0.718	0.722	0.729	0.725	**0.739**

The symbol "√" indicates that this module is included in the model of the experiment. The bold font is used to distinctly indicate the highest values achieved for each evaluation metric.

Table 7. Comparison of two balancing strategies in independent testing.

Balancing strategies	ACC	AUC	Sn	Sp	F1	MCC
Downsampling	0.680	0.754	0.623	**0.736**	0.660	0.361
Focal loss	**0.691**	**0.765**	**0.878**	0.503	**0.739**	**0.411**

The best experimental results are highlighted in bold.

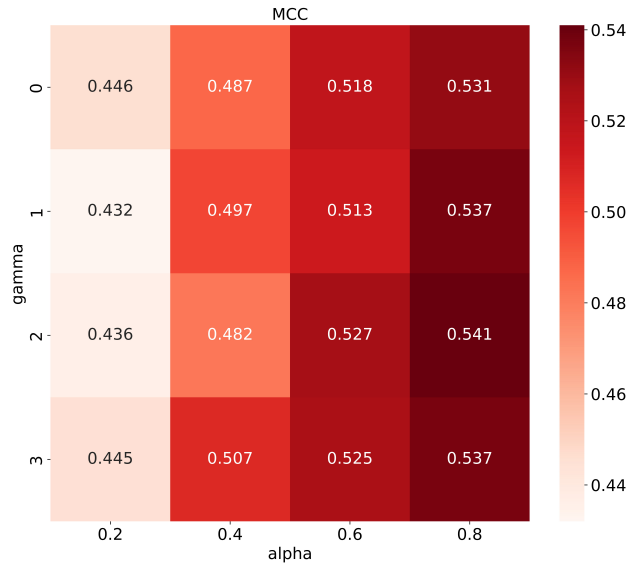

Fig. 9. MCC values for different combinations of parameters. This figure shows the effect of different combinations of α $(0.2, 0.4, 0.6, 0.8)$ and γ $(0, 1, 2, 3)$ on the predicted MCC values of m^6Am. Darker colors indicate higher MCC. Generated using Python, specifically with the matplotlib and seaborn libraries.

As a result, the Focal Loss function shows superior performance when dealing with unbalanced datasets.

3.5 Comparative Analysis of Other Models

To demonstrate the effectiveness of DTC-m6Am, we compared it to the current state-of-the-art m6Aminer model. We performed ten-fold cross-validation on the training dataset and performance evaluation on an independent test dataset in which the cross-validation was repeated ten times. Based on the data results in Tables 8,9 (Ref. [22]), the DTC-m6Am model shows significant superiority over the state-of-the-art m6Aminer model in several key performance metrics. First, the 10-fold cross-validation results in Table 8 show that DTC-m6Am achieves 0.904 in Accuracy (ACC), which is higher than m6Aminer's 0.891, indicating that DTC-m6Am performs more consistently and has higher prediction accuracy on the training set. In addition, DTC-m6Am also performs more prominently on specificity (Sp), reaching 0.922, which is significantly better than m6Aminer's 0.904, implying that DTC-m6Am has a stronger ability to correctly identify negative samples, which helps to reduce the false-positive rate. Fig. 10 provides a clear visual comparison between DTC-m6Am and existing predictors.

Observing the results of the independent tests in Fig. 11, DTC-m6Am again demonstrated its advantages in various metrics. DTC-m6Am reached 0.878 in accuracy (Sn) compared to the m6Aminer model to remain competitive, and in the five metrics of specificity (Sp), accuracy (ACC), matthews correlation coefficient (MCC), area under the curve (AUC), and F1 respectively amounted to 0.503, 0.691, 0.411, 0.765, and 0.739, which were improved by 6.8%, 1.6%, 19.7%, 3.6%, and 24.1% compared with m6Aminer. Its accuracy (ACC) of 0.691 was higher than that of m6Aminer of 0.647, indicating better prediction on unseen data. Especially in specificity (Sp), DTC-m6Am significantly improved to 0.503, which showed higher negative sample recognition ability compared to 0.420 of m6Aminer. That is, it is shown that in practical applications, our model can effectively reduce the false positive rate, shorten the validation period and significantly save the experimental resources and time cost compared with the existing tools. More importantly, DTC-m6Am is

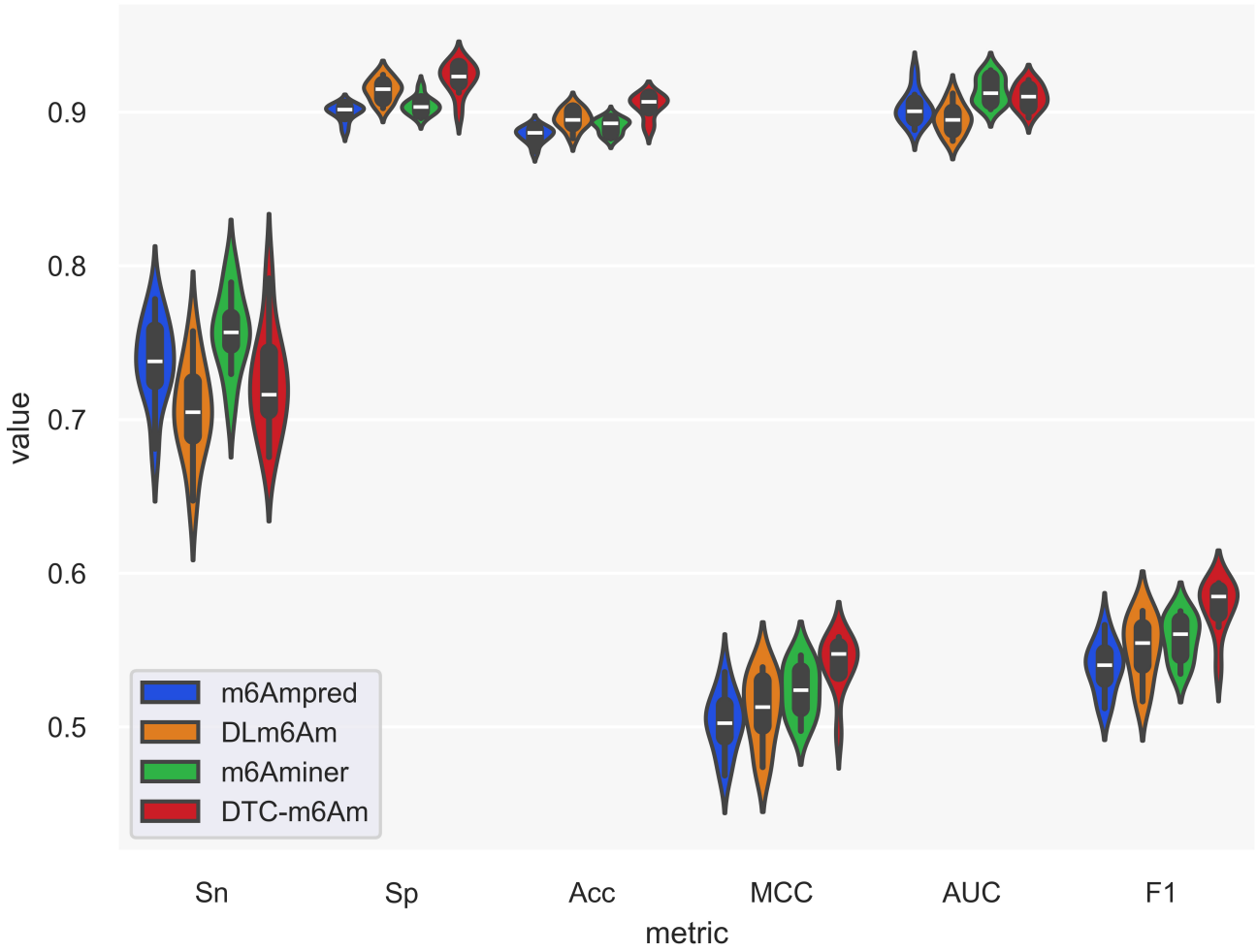

Fig. 10. Comparison of the DTC-m6Am model with state-of-the-art prediction models. The figure illustrates the performance distribution of the four models under 10-fold cross-validation. DTC-m6Am (red) performs better in several metrics, especially in the MCC and F1 scores. Generated using Python, specifically with the matplotlib and seaborn libraries.

Table 8. 10-fold cross-validation on imbalanced training sets for various models.

Predictor	ACC	AUC	Sn	Sp	F1	MCC
m6AmPred	0.885 ± 0.005	0.903 ± 0.010	0.738 ± 0.026	0.900 ± 0.005	0.539 ± 0.016	0.503 ± 0.019
DLm6Am	0.895 ± 0.006	0.895 ± 0.009	0.705 ± 0.029	0.914 ± 0.007	0.551 ± 0.019	0.511 ± 0.022
m6Aminer	0.891 ± 0.005	**0.914 ± 0.009**	**0.756 ± 0.025**	0.904 ± 0.005	0.557 ± 0.014	0.523 ± 0.017
DTC-m6Am	**0.904 ± 0.007**	0.910 ± 0.007	0.724 ± 0.032	**0.922 ± 0.010**	**0.579 ± 0.016**	**0.541 ± 0.017**

Each value in the table is expressed as mean ± SD. where the mean is the average of the 10-fold cross-validation results and SD is their standard deviation. The best experimental results are highlighted in bold.

also ahead of m6Aminer in terms of Matthews correlation coefficient (MCC), reaching 0.411, while m6Aminer is only 0.331. The higher MCC value of DTC-m6Am indicates that it can recognize negative samples in all types of data.

In both ten-fold cross-validation and independent tests, the DTC-m6Am model shows excellent prediction performance, which is significantly better than tools such as m6Aminer. The core advantage lies in the powerful feature extraction capability of DenseNet, which can capture the negative sample information more adequately, thus improving the discriminative ability of the model. In addition, we introduce Focal Loss ($\alpha = 0.8$, $\gamma = 2$) during the training process, which effectively reduces the model's bias towards majority class samples by dynamically adjusting the loss weights, prompting it to pay more attention to the minority samples that are difficult to categorize, so as to alleviate the problem of class imbalance and improve the overall prediction performance. Taken together, DTC-m6Am, with both robustness and generalization ability, is an efficient and reliable m^6Am prediction model, providing a powerful tool for RNA modification sites research.

Table 9. Model comparison with the state-of-the-art models on the independent testing dataset.

Predictor	ACC	AUC	Sn	Sp	F1	MCC
m6AmPred*	0.623	0.735	**0.887**	0.358	0.702	0.289
DLm6Am*	0.642	0.730	0.875	0.409	0.710	0.322
m6Aminer*	0.647	0.754	0.874	0.420	0.713	0.331
DTC-m6Am	**0.691**	**0.765**	0.878	**0.503**	**0.739**	**0.411**

The conclusions were from the previous study [22], as stated by the asterisk (*) , and the best experimental results are highlighted in bold.

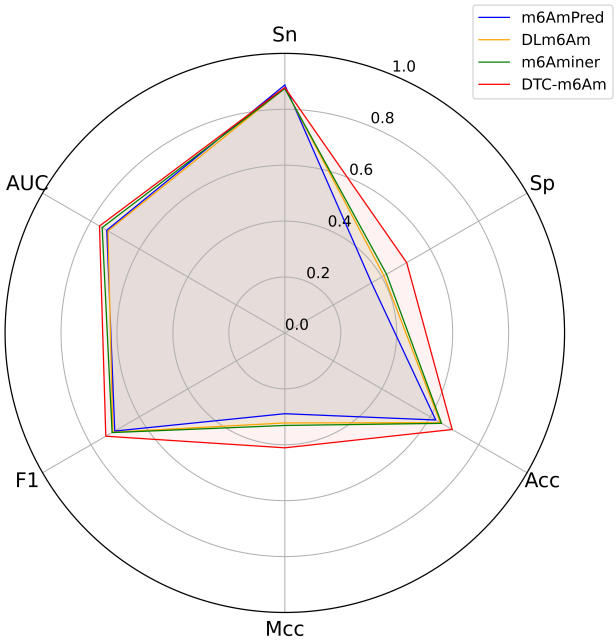

Fig. 11. Results of the DTC-m6Am model and other models in independent tests. The metrics include Sn, Sp, ACC, MCC, AUC and F1, with values closer to the outer circle indicating better performance. The results show that DTC-m6Am (red) excels in several metrics, indicating its strong generalization ability. Generated using Python, specifically with the matplotlib and seaborn libraries.

3.6 Sequence Analysis of m^{6}Am Sites

In order to find out the motif pattern of the m^{6}Am site, we used the kplogo [44] tool to analyze the m^{6}Am positive sequence data, and the visualization results are shown in Fig. 12. In this study, adenine (A) was located in the center of the RNA sequence fragment, with 20 nucleotides before and after. k-mer Logo analysis results highlighted that at positions 19 to 22, specific k-mer (e.g., U, C, CAU, etc.) with significant enrichment, suggesting that these sequences may play a key role in the recognition and formation of the m^{6}Am modification. Especially at position 23, significant variability was observed, and the k-mer represented by ANC may indicate that more sequence diversity or regulation by environmental factors may be involved in m^{6}Am modification at this position. In addition, the significant depletion of G at positions 19 and 23 may be closely related to the selective recognition and modification of m^{6}Am modification for these positions, suggesting that the modification may be regulated by the context upstream and downstream of the sequence. Thus, combining contextual information can help identify m^{6}Am sites.

3.7 An Available Web Server for DTC-m6Am

The m^{6}Am site identification methods using deep learning approaches often require the configuration of cumbersome deep learning environments as well as the debugging of complex computational models, which constitutes a high threshold of use for most non-specialized researchers. To solve this problem, we developed the DTC-m6Am website to provide efficient, convenient, and accurate online m^{6}Am site prediction services. The platform is built to lower the technical threshold of m^{6}Am site prediction and help researchers focus on experimental design and biological function validation, thus accelerating the advancement of scientific research. By integrating advanced machine learning models with known data training, the platform is able to quickly identify potential m^{6}Am sites in input RNA sequences.

The web interface is shown in Fig. 13. Users only need to paste the target mRNA sequence as text into the input box of the website and click “Submit” to get the prediction results. In addition, the website supports batch uploading of FASTA format files and returns the results via email, which is convenient for users to perform high-throughput data analysis. Our website can be accessed at https://hardy-leech-conversely.ngrok-free.app/.

4. Conclusions

As an important RNA epitope marker, m^{6}Am modification plays a key role in gene expression regulation, affecting mRNA stability, splicing, and translation efficiency. Based on its importance, we propose a deep learning model that combines the feature extraction capability of DenseNet, the sequence modeling advantage of TCN, and the attention mechanism of the CBAM module to design a high-precision m^{6}Am predictor to identify the m^{6}Am site better and facilitate epigenetic studies. DTC-m6Am has the following advantages over previous studies: (1) DTC-m6Am used an algorithm-based non-equilibrium strategy in the m^{6}Am site prediction study, which to some extent solved the problem of bias toward the majority class during the training process and ensured the recognition ability of positive samples while significantly improving the recognition ability of negative samples, with significant improvement compared to the data-based downsampling method. (2) DTC-m6Am employs a simple and efficient one-hot coding method, which is intuitive and easy to implement without complex feature engineering. It can retain the integrity and clarity of the original data. In addition, one-hot coding

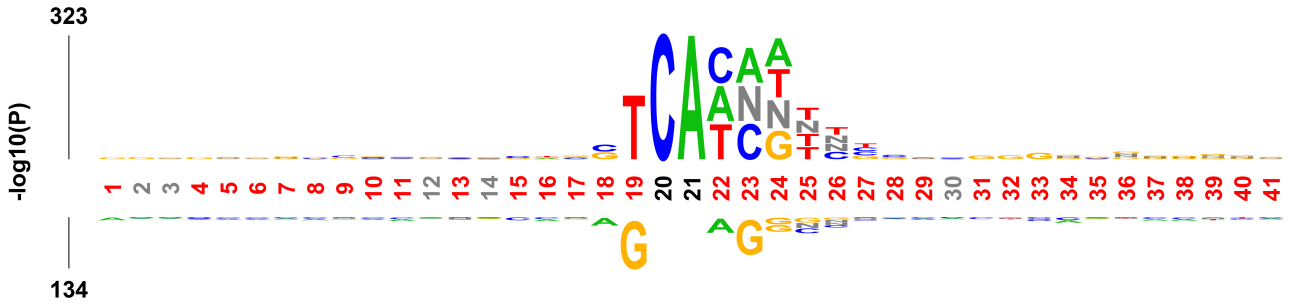

Fig. 12. Visualisation results of kpLogo for k-mer logo. The sequence information in the figure highlights the base preference and informativeness of each position, with blue, green, red, and yellow indicating bases C, A, U (T), and G, respectively, and the font size reflecting the importance and frequency of the bases at that position, which helps to identify key sequence patterns. Created using the web tool available at http://kplogo.wi.mit.edu/manual.html.

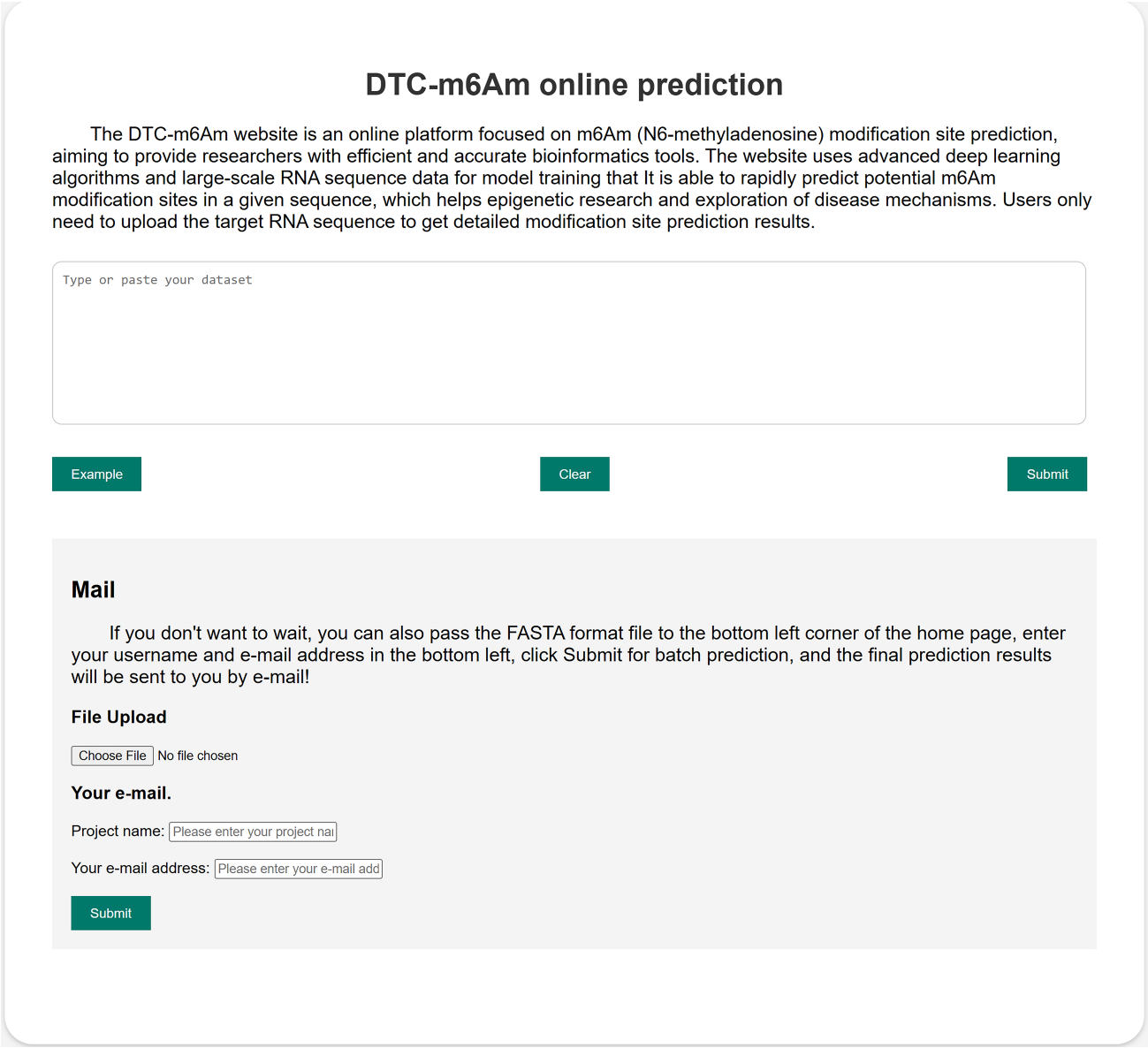

Fig. 13. Homepage of the DTC-m6Am predictor. Sourced from https://hardy-leech-conversely.ngrok-free.app/.

is more advantageous in terms of computational overhead, which helps to improve the training efficiency and prediction performance of the model. (3) We have also developed an m^6Am prediction website. This platform not only enables real-time access and application of the DTC-m6Am model we developed, but also automates prediction based on RNA sequences uploaded by users. Through simple sequence uploading and clicking operations, users can obtain the prediction results of m^6Am sites without the need for professional programming ability, which greatly reduces the threshold of use.

Although DTC-m6Am exhibits superior predictive performance, there are some potential limitations. First, the performance of the model is limited by the size and diversity of the training data, which may undergeneralize due to data scarcity especially in cross-species or rare modification scenarios. Second, although we employ Focal Loss to mitigate the category imbalance problem, the prediction of a few categories may still be affected in extremely imbalanced datasets. In addition, signaling overlap in high-density regions (e.g., m^6A/m^6Am competitive modifications) and the complexity of RNA dynamic structure make it difficult to accurately resolve modification sites with existing features. In terms of feature encoding, One-Hot encoding, while simple and efficient, has limitations in representing complex structural features, especially when it is necessary to capture relationships between neighboring base pairs in a sequence. This limitation may affect the model's ability to resolve complex RNA modification sites. Therefore, exploring other coding schemes such as PseEIIP and PSSM coding has the potential to improve the model's ability to represent complex structural features by better capturing the neighboring relationships and chemical properties in the sequence. Future studies can combine methods such as migration learning or self-supervised learning by introducing larger and more diverse datasets in order to improve the generalization and robustness of the model. Meanwhile, the integration of RNA secondary structure or multimodal data into the model can be further explored, and the coupling of sequence-structure-function can be captured by graph neural networks or 3D convolutional architectures to enhance the decoding ability of the spatial distribution of modification sites and the dynamic regulatory mechanisms.

Availability of Data and Materials

Both the dataset and source code for this work are freely available at https://github.com/hhui0/DTC-m6Am. Data supporting the findings are available from the corresponding or first author upon reasonable request.

Author Contributions

Conceptualization, JJ, HH; methodology, JJ, HH and HZ; Formal analysis, FZ, HH; validation, FZ; investigation, HH, HZ; resources, FZ; data curation, HH, ZH; writing—preparation of the original project, HH, FZ; writing—reviewing and editing, FZ, HH and JJ; visualization, HH; supervision, JJ, FZ. All authors read and approved the final manuscript. All authors have participated sufficiently in the work and agreed to be accountable for all aspects of the work.

Ethics Approval and Consent to Participate

Not applicable.

Acknowledgment

Not applicable.

Funding

This work was partially supported by the Scientific Research Plan of the Department of Education of Jiangxi Province, China (GJJ2400909, GJJ2402711). These funders had no role in the study design, data collection and analysis, decision to publish or preparation of the manuscript.

Conflict of Interest

The authors declare no conflict of interest.

Declaration of AI and AI-assisted Technologies in the Writing Process

In preparing this work, the authors used ChatGPT-4 (OpenAI) to find appropriate word substitutes and overcome non-native language barriers. With the support of this AI tool, the authors reviewed and edited the content as needed and take full responsibility for the final publication.

References

[1] Cappannini A, Ray A, Purta E, Mukherjee S, Boccaletto P, Moafinejad SN, *et al*. MODOMICS: a database of RNA modifications and related information. 2023 update. Nucleic Acids Research. 2024; 52: D239–D244. https://doi.org/10.1093/nar/gkad1083.

[2] Wei C, Gershowitz A, Moss B. N6, O2'-dimethyladenosine a novel methylated ribonucleoside next to the 5' terminal of animal cell and virus mRNAs. Nature. 1975; 257: 251–253. https://doi.org/10.1038/257251a0.

[3] Sendinc E, Valle-Garcia D, Dhall A, Chen H, Henriques T, Navarrete-Perea J, *et al*. PCIF1 Catalyzes m6Am mRNA Methylation to Regulate Gene Expression. Molecular Cell. 2019; 75: 620–630.e9. https://doi.org/10.1016/j.molcel.2019.05.030.

[4] Benak D, Benakova S, Plecita-Hlavata L, Hlavackova M. The role of m^6A and m^6Am RNA modifications in the pathogenesis of diabetes mellitus. Frontiers in Endocrinology. 2023; 14: 1223583. https://doi.org/10.3389/fendo.2023.1223583.

[5] Jin H, Shi Z, Zhou T, Xie S. Regulation of m6Am RNA modification and its implications in human diseases. Journal of Molecular Cell Biology. 2024; 16: mjae012. https://doi.org/10.1093/jmcb/mjae012.

[6] Pandey RR, Delfino E, Homolka D, Roithova A, Chen KM, Li L, *et al*. The Mammalian Cap-Specific m^6Am RNA Methyltransferase PCIF1 Regulates Transcript Levels in Mouse Tissues. Cell Reports. 2020; 32: 108038. https://doi.org/10.1016/j.celrep.2020.108038.

[7] Ben-Haim MS, Pinto Y, Moshitch-Moshkovitz S, Hershkovitz V, Kol N, Diamant-Levi T, *et al*. Dynamic regulation of N^6,2'-O-dimethyladenosine (m^6Am) in obesity. Nature Communications. 2021; 12: 7185. https://doi.org/10.1038/s41467-021-27421-2.

[8] Li N, Rana TM. Regulation of antiviral innate immunity by chemical modification of viral RNA. Wiley Interdisciplinary Reviews. RNA. 2022; 13: e1720. https://doi.org/10.1002/wrna.1720.

[9] Zhang Q, Kang Y, Wang S, Gonzalez GM, Li W, Hui H, *et al*. HIV reprograms host m^6Am RNA methylome by viral Vpr protein-mediated degradation of PCIF1. Nature Communications. 2021; 12: 5543. https://doi.org/10.1038/s41467-021-25683-4.

[10] Wang L, Wu L, Zhu Z, Zhang Q, Li W, Gonzalez GM, *et al*. Role of PCIF1-mediated 5'-cap N6-methyladeonsine mRNA methylation in colorectal cancer and anti-PD-1 immunotherapy. The EMBO Journal. 2023; 42: e111673. https://doi.org/10.15252/embj.2022111673.

[11] Wang L, Wang S, Wu L, Li W, Bray W, Clark AE, *et al*. PCIF1-mediated deposition of 5'-cap N^6,2'-*O*-dimethyladenosine in ACE2 and TMPRSS2 mRNA regulates susceptibility to SARS-CoV-2 infection. Proceedings of the National Academy of Sciences of the United States of America. 2023; 120: e2210361120. https://doi.org/10.1073/pnas.2210361120.

[12] Zhuo W, Sun M, Wang K, Zhang L, Li K, Yi D, *et al*. m^6Am methyltransferase PCIF1 is essential for aggressiveness of gastric cancer cells by inhibiting TM9SF1 mRNA translation. Cell Discovery. 2022; 8: 48. https://doi.org/10.1038/s41421-022-00395-1.

[13] Linder B, Grozhik AV, Olarerin-George AO, Meydan C, Mason CE, Jaffrey SR. Single-nucleotide-resolution mapping of m6A and m6Am throughout the transcriptome. Nature Methods. 2015; 12: 767–772. https://doi.org/10.1038/nmeth.3453.

[14] Koh CWQ, Goh YT, Goh WSS. Atlas of quantitative single-base-resolution N^6-methyl-adenine methylomes. Nature Communications. 2019; 10: 5636. https://doi.org/10.1038/s41467-019-13561-z.

[15] Sun H, Li K, Zhang X, Liu J, Zhang M, Meng H, *et al*. m^6Am-seq reveals the dynamic m^6Am methylation in the human transcriptome. Nature Communications. 2021; 12: 4778. https://doi.org/10.1038/s41467-021-25105-5.

[16] Shaon MSH, Karim T, Ali MM, Ahmed K, Bui FM, Chen L, *et al*. A robust deep learning approach for identification of RNA 5-methyluridine sites. Scientific Reports. 2024; 14: 25688. https://doi.org/10.1038/s41598-024-76148-9.

[17] Zhao Y, Jin J, Gao W, Qiao J, Wei L. Moss-m7G: A Motif-Based Interpretable Deep Learning Method for RNA N7-Methlguanosine Site Prediction. Journal of Chemical Information and Modeling. 2024; 64: 6230–6240. https://doi.org/10.1021/acs.jcim.4c00802.

[18] Song Z, Huang D, Song B, Chen K, Song Y, Liu G, *et al*. Attention-based multi-label neural networks for integrated prediction and interpretation of twelve widely occurring RNA modifications. Nature Communications. 2021; 12: 4011. https://doi.org/10.1038/s41467-021-24313-3.

[19] Jiang J, Song B, Chen K, Lu Z, Rong R, Zhong Y, *et al*. m6AmPred: Identifying RNA N6, 2'-O-dimethyladenosine (m^6A_m) sites based on sequence-derived information. Methods (San Diego, Calif.). 2022; 203: 328–334. https://doi.org/10.1016/j.ymeth.2021.01.007.

[20] Luo Z, Su W, Lou L, Qiu W, Xiao X, Xu Z. DLm6Am: A Deep-Learning-Based Tool for Identifying N6,2'-O-Dimethyladenosine Sites in RNA Sequences. International Journal of Molecular Sciences. 2022; 23: 11026. https://doi.org/10.3390/ijms231911026.

[21] Jia J, Wei Z, Sun M. EMDL_m6Am: identifying N6,2'-O-dimethyladenosine sites based on stacking ensemble deep learning. BMC Bioinformatics. 2023; 24: 397. https://doi.org/10.1186/s12859-023-05543-2.

[22] Liu Z, Lan P, Liu T, Liu X, Liu T. m6Aminer: Predicting the m6Am Sites on mRNA by Fusing Multiple Sequence-Derived Features into a CatBoost-Based Classifier. International Journal of Molecular Sciences. 2023; 24: 7878. https://doi.org/10.3390/ijms24097878.

[23] Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics (Oxford, England). 2012; 28: 3150–3152. https://doi.org/10.1093/bioinformatics/bts565.

[24] Bari AG, Reaz MR, Choi HJ, Jeong BS. DNA encoding for splice site prediction in large DNA sequence. In Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, International Workshops: BDMA, SNSM, SeCoP, Wuhan, China, April 22–25, 2013. Proceedings 18 (pp. 46–58). Springer: Berlin Heidelberg. 2013. https://doi.org/10.1007/978-3-642-40270-8_4.

[25] Ao C, Zou Q, Yu L. NmRF: identification of multispecies RNA 2'-O-methylation modification sites from RNA sequences. Briefings in Bioinformatics. 2022; 23: bbab480. https://doi.org/10.1093/bib/bbab480.

[26] Pham NT, Rakkiyapan R, Park J, Malik A, Manavalan B. H2Opred: a robust and efficient hybrid deep learning model for predicting 2'-O-methylation sites in human RNA. Briefings in Bioinformatics. 2023; 25: bbad476. https://doi.org/10.1093/bib/bbad476.

[27] Feng P, Ding H, Yang H, Chen W, Lin H, Chou KC. iRNA-PseColl: Identifying the Occurrence Sites of Different RNA Modifications by Incorporating Collective Effects of Nucleotides into PseKNC. Molecular Therapy. Nucleic Acids. 2017; 7: 155–163. https://doi.org/10.1016/j.omtn.2017.03.006.

[28] Jia J, Deng Y, Yi M, Zhu Y. 4mCPred-GSIMP: Predicting DNA N4-methylcytosine sites in the mouse genome with multi-Scale adaptive features extraction and fusion. Mathematical Biosciences and Engineering: MBE. 2024; 21: 253–271. https://doi.org/10.3934/mbe.2024012.

[29] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700–4708). 2017. https://doi.org/10.1109/CVPR.2017.243.

[30] Wang H, Zhao H, Yan Z, Zhao J, Han J. MDCAN-Lys: A Model for Predicting Succinylation Sites Based on Multilane Dense Convolutional Attention Network. Biomolecules. 2021; 11: 872. https://doi.org/10.3390/biom11060872.

[31] Jia J, Lei R, Qin L, Wei X. i5mC-DCGA: an improved hybrid network framework based on the CBAM attention mechanism for identifying promoter 5mC sites. BMC Genomics. 2024; 25: 242. https://doi.org/10.1186/s12864-024-10154-z.

[32] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778). 2016. https://doi.org/10.1109/CVPR.2016.90.

[33] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167. 2015. https://doi.org/10.48550/arXiv.1502.03167.

[34] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks. Proceedings of the fourteenth international conference on artificial intelligence and statistics. 2011. Available at: https://proceedings.mlr.press/v15/glorot11a.html (Accessed: 9 March 2025).

[35] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE.

1998; 86: 2278–2324. https://doi.org/10.1109/5.726791.

[36] Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271. 2018. https://doi.org/10.48550/arXiv.1803.01271.

[37] Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, *et al*. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499. 2016; 12. https://doi.org/10.48550/arXiv.1609.03499.

[38] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122. 2015. https://doi.org/10.48550/arXiv.1511.07122.

[39] Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3–19). 2018. https://doi.org/10.1007/978-3-030-01234-2_1.

[40] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980–2988). 2018. https://doi.org/10.1109/TPAMI.2018.2858826.

[41] Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. 2020; 21: 6. https://doi.org/10.1186/s12864-019-6413-7.

[42] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, *et al*. Attention is all you need. Advances in Neural Information Processing Systems. 2017; 30. https://doi.org/10.48550/ARXIV.1706.03762.

[43] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132–7141). 2018. https://doi.org/10.48550/arXiv.1709.01507.

[44] Wu X, Bartel DP. kpLogo: positional k-mer analysis reveals hidden specificity in biological sequences. Nucleic Acids Research. 2017; 45: W534–W538. https://doi.org/10.1093/nar/gkx323.