Measurement 226 (2024) 114157

Contents lists available at ScienceDirect

Measurement

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

CTCNet: A CNN Transformer capsule network for sleep stage classification

Weijie Zhang [a], Chang Li [a,*], Hu Peng [a,b], Heyuan Qiao [a], Xun Chen [c,d]

[a] Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China
[b] Anhui Province Key Laboratory of Measuring Theory and Precision Instrument, Hefei University of Technology, Hefei 230009, China
[c] Department of Neurosurgery, the First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei, 230001, Anhui, China
[d] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China

ARTICLE INFO

Keywords:
Electroencephalogram (EEG)
Sleep stage classification
CNN
Transformer
Capsule network

ABSTRACT

In this paper, we propose a novel neural network architecture called CTCNet. First, we adopt a multi-scale convolutional neural network (MSCNN) to extract low and high-frequency features, adaptive channel feature recalibration (ACFR) to enhance the model's sensitivity to important channel features in the feature maps and reduce dependence on irrelevant or redundant features, a multi-scale dilated convolutional block (MSDCB) to capture characteristics of different types among feature channels. Second, we use Transformer to extract global temporal context features. Third, we employ capsule network to capture spatial location relationships among EEG features and refine these features. Besides, the capsule network module is used as our model's classifier to classify the final results. It is worth noting that our model better solves the problem that previous researches failed to take into account the simultaneous extraction of local features and global temporal context characteristics of EEG signals, and ignored the spatial location relationships between these features. Eventually, we assess our model on three datasets and it achieves better or comparable performance than most state-of-the-art methods.

1. Introduction

Sleep is important for human health, especially deep sleep, which restores brain capacity and removes waste from the brain [1]. However, nowadays, many people have trouble sleeping. Polysomnography (PSG) is a great aid for sleep specialists, which helps them to be familiar with patients' sleep patterns and address their sleep issues.

PSG is composed of electroencephalogram (EEG), electroocular (EOG), electromyogram (EMG) and electrocardiography (ECG) [2–4]. EEG is favored by sleep experts due to its low cost, ease of use, and high temporal resolution [5–8]. According to the guidelines of the American Academy of Sleep Medicine (AASM) [9] or the Rechtschaffen and Kales (R&K) [10], sleep specialists divide the data in PSG into many frames (30s per frame) and assign a sleep stage to each frame. Sleep stage classification is considered an effective way to diagnose sleep disorders [11]. However, manual classification by sleep specialists is not only time-consuming and laborious but also prone to subjective mistakes [12]. Therefore, it is vital to design good automatic sleep classification methods.

Some researchers have proposed methods based on traditional machine learning. First, they denoise EEG signals and then extract useful features from the processed signals. Second, they feed the extracted features into some classifiers to classify sleep stages, such as XGBoost [13], support vector machines (SVM) [14,15], random forest (RF) [16,17], and K-nearest neighbors (KNN) [18,19]. Nevertheless, researchers need to have relevant expertise to capture valuable features.

Nowadays, many methods based on deep learning, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Transformer have become a popular direction for dealing with sleep stage classification problems based on EEG signals. Several studies used CNN for this task [20,21]. For example, in [20], the authors proposed a prediction framework based on CNN and used the joint classification method to classify sleep stages. Tsinalis et al. [21] used convolutional layers to extract information and then utilized maximum pooling layers to filter it. All in all, CNNs perform excellently in sleep stage classification, but they do not learn well about the temporal context relationship between the EEG data. RNNs are proposed to solve the problem [22,23]. Phan et al. [22] combined RNN and the attention mechanism to classify sleep stages. Supretak et al. [23] used CNN to extract features and adopted BiLSTM to learn the temporal relationship among these features. However, due to the unique recurrent nature

* Corresponding author.
E-mail addresses: weijiezhang@mail.hfut.edu.cn (W. Zhang), changli@hfut.edu.cn (C. Li), hpeng@hfut.edu.cn (H. Peng), qiao_heyuan@hfut.edu.cn (H. Qiao), xunchen@ustc.edu.cn (X. Chen).

https://doi.org/10.1016/j.measurement.2024.114157
Received 6 October 2023; Received in revised form 1 January 2024; Accepted 7 January 2024
Available online 9 January 2024

of RNNs, they have high complexity and are hard to train in parallel. Some researchers employ Transformer for this task. For example, Phan et al. [24] used Transformer to classify sleep stages and achieved active results. In summary, CNN lacks the ability to extract spatial position information and global temporal context relationships between features. It is difficult for RNN to train the extracted global temporal context features in parallel. In addition, although Transformer can learn global temporal context information well, its ability to obtain local information is weak.

To resolve the above problems well, we design a neural network called CTCNet, which consists of multi-scale CNN (MSCNN), adaptive channel feature recalibration (ACFR), multi-scale dilated convolutional block (MSDCB), a Transformer module, and a capsule network module. In the CTCNet model, in order to fully extract the EEG features of high and low frequencies, we adopt MSCNN. ACFR is used to recalibrate channel characteristics, so as to obtain important channel information and discard useless channel information. In order to capture the different frequencies and complex features in EEG signals, a multi-scale dilated convolutional block (MSDCB) is used. In the transformer module, we divide the EEG features into multiple tokens, and then all tokens are embedded in different blocks, finally by obtaining the context relationships between the blocks, the Transformer module can learn the global temporal context characteristics. In the capsule network module, we encode various properties of EEG features into vector neurons (i.e., capsules). Through the unique dynamic routing mechanism of capsule network, the capsule network module is able to capture the spatial location relationships among EEG features. The contribution of CTCNet is able to be described as follows:

(1) We design a novel model called CTCNet for sleep stage classification. CTCNet is capable of capturing the local features of EEG signals. Besides, it does well in capturing the global temporal context information of these features. In addition, it has the ability to obtain the spatial position relationship between features well.
(2) We perform multiple experiments on three datasets. The Specific results are that the accuracy on Sleep-EDF-20 is 86.2%, on Sleep-EDF-78 is 82.5%, and on SHHS is 85.7%. From the above results, the performance of our model is at a leading level.

The remainder of the article is arranged as follows: we briefly introduce the related work in Section 2. In Section 3, we explain the CTCNet model in detail. We introduce the datasets, the input to our model, the experimental details, and the final experimental results in Section 4. Section 5 provides the related discussions. At last, we summarize this article in Section 6.

2. Related work

Since our model uses CNN, Transformer, and capsule network, we briefly introduce them.

2.1. CNN

CNN is a kind of feedforward neural network that contains convolutional computation and has a deep structure. In 1998, Lecun et al. [25] introduced LeNet-5, a pioneering convolutional neural network that defined the basic framework of CNNs: convolutional layer, pooling layer, and fully connected layer. It also spurred growing interest among researchers in CNNs due to its remarkable achievements on the Minist dataset. Since then, a succession of convolutional neural network architectures have emerged and the most notable of them are AlexNet [26], VGG [27], and ResNet [28].

In EEG-related fields, CNN also shows good performance. Roy et al. designed a high-performance multi-scale CNN to extract EEG features for moving image classification of brain–computer interfaces (MI-BCI) [29]. Dai et al. proposed a novel method for EEG moving images based on CNN [30]. Mao et al. used a CNN model based on EEG signals to predict the duration of seizures [31]. Kong et al. proposed a CNN architecture based on neural structure search to classify sleep stages and performed well [32]. In a word, the development of CNN is characterized by continuous innovation. From early models such as LeNet-5 to today's deep and complex structures, CNN has made significant advances in image processing, pattern recognition, and artificial intelligence. It also has enabled breakthrough performance improvements in a variety of applications.

2.2. Transformer

Transformer was proposed by Google in 2017 [33]. It revolutionizes natural language processing and various other tasks by employing the concept of self-attention mechanism to learn relationships among different words in a sentence without relying on sequential processing. Transformer generally adopts an encoder–decoder structure. They have the same architecture. Taking the encoder as an example, it consists of three parts, namely, multi-head attention mechanism, add and normalize layers, and feedforward layers [34]. So far, Transformer has excelled in computer vision (CV) and natural language processing (NLP).

In NLP, Devlin et al. designed an improved Transformer-based model called BERT, and it has shown excellent performance in the field of NLP. [35]. In [36], they proposed a context-based generative model called GPT-3. It has 175 billion parameters, which means that GPT-3 can supplement the rest with less contextual content information. In CV, Dosovitskiy et al. proposed a variant of Transformer called the vision Transformer (ViT). They first split the image into patches (16×16) and then fed these patches into Transformer to make predictions [37]. Another one is DETR. Carion et al. used CNN and Transformer to predict the classes [38]. In object detection, it achieved satisfactory results.

Deep learning methods based on Transformer are also applied to fields that are related to EEG. Sun et al. proposed a model combining Transformer and 3D CNN for emotion recognition [39]. In [40] the authors designed a model of emotion recognition based on Transformer. They used Transformer to acquire spatial information from EEG. Gong et al. combined CNN and Transformer for emotion recognition [41]. In essence, Transformer is a milestone that takes deep learning to the next level.

2.3. Capsule network

In 2017, capsule network was first proposed by Sabour et al. [42]. The core of it is dynamic routing mechanism. Due to this mechanism, capsule network can simultaneously encode spatial information and the probability of feature existence and store them in capsules. This means that the modulus of the capsule vector is the probability of the existence of the feature, the direction of the vector represents the pose information of the feature. In addition, moving features may change the orientation of the capsule vectors but do not change the probability of the existence of the features.

It is first applied in computer vision (CV). Based on the capsule network, Paoletti et al. developed a CNN model extension for classifying hyperspectral images [43]. Gupta et al. used COVID-WideNet to diagnose COVID-19-infected patients and performed better than any other CNN-based method [44]. Jaiswal et al. combined CapsNet with generative adversarial networks (GANs) and showed strengths in MNIST and CIFAR-10 datasets [45].

Nowadays, due to the excellent performance of the capsule network, it has been extensively used in EEG-related fields. Liu et al. designed a binary capsule network (Bi-CapsNet) for EEG emotion recognition. This model has the benefits of low computation and low memory usage [46]. Chen et al. combined the improved capsule network with BiLSTM, achieving good results [47]. Liu et al. proposed an end-to-end capsule network-based architecture for EEG emotion recognition. It can enhance feature representation to improve recognition accuracy [48].

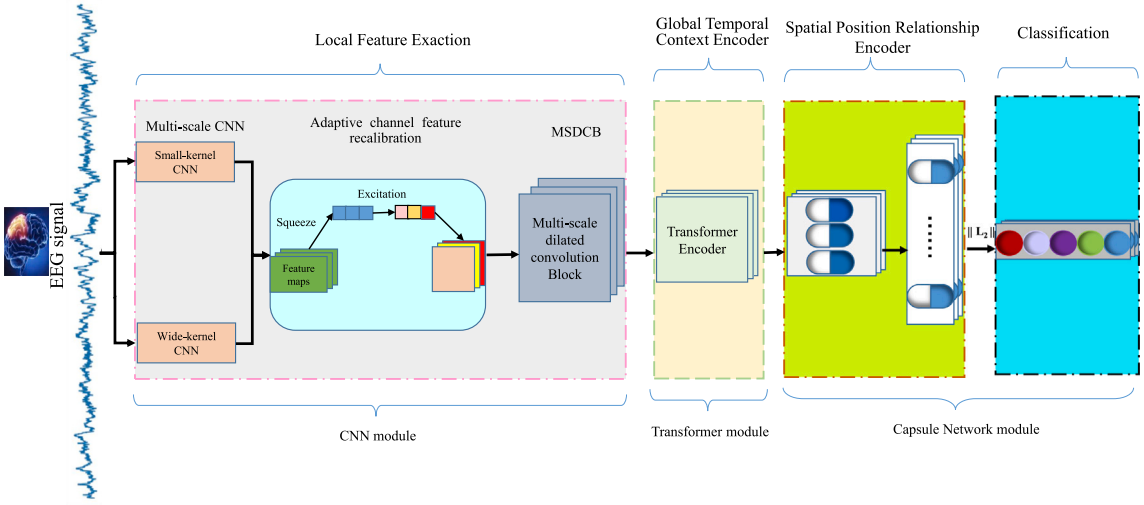

Fig. 1. The specific structure of CTCNet.

3. Methods

In the section, we explain our CTCNet model specifically.

3.1. Overview of CTCNet model

The structure of the CTCNet model is shown in Fig. 1. It is made up of four portions, (1) local feature extraction, (2) global temporal context encoder, (3) spatial position relationship encoder, and (4) classification.

3.2. Local feature extraction

3.2.1. Mutil-scale CNN

To better capture features of different frequencies in EEG signals, we design a multi-scale CNN (MSCNN). In Tables 1 and 2, the frequency of brain waves used for sleep stage classification is between 0–50 Hz [16], so we adopt two CNNs with different kernel sizes at the first layer to exact different frequencies. The small one can capture high-frequency features of EEG signals, while the large one is able to capture low-frequency features of EEG signals. For example, if a dataset has a sampling rate (Fs) of 100 Hz (100 points per second), we set the large kernel size to 400 (Fs×4) to capsule points with 4s (F = 4) windows, so that it can capture the entire period of sinusoidal signals as low as ≤0.25 Hz. The frequency of the range corresponds to δ band, and we set the convolution kernel of another CNN to 50 (Fs/2), so it can exact the entire period of sinusoidal signals down to ≤2 Hz, which means that the frequency of this range corresponds to α and θ bands.

In MSCNN, each CNN is composed of four convolutional layers, two maximum pooling layers, and a batch normalization layer. Besides, we adopt GELU activation function to maintain the non-linearity of MSCNN. Each pooling layer is used to downsample the inputs. The specific hyperparameter settings are shown in Fig. 2.

3.2.2. Adaptive channel feature recalibration (ACFR)

The role of ACFR is to recalibrate the channel features in the feature maps learned from MSCNN, so that important features can be selectively enhanced and unimportant features compressed. Its core part is the squeeze and excitation block (SE Block) in Fig. 3. SE block can reassign the weight distribution between channels by learning the correlation between channels and finally redistributing the channel feature weight. It is composed of an adaptive average pooling layer and two fully connected (FC) layers. First, the adaptive average pooling layer compresses the feature information of the feature maps. Then, two

Table 1
Different kinds of brain waves correspond to the bands.

Brain wave	Band (Hz)
Delta (δ)	0–4
Theta (θ)	4–8
Alpha (α)	8–12
Sigma (σ)	12–15
Beta 1 (β_1)	15–22
Beta 2 (β_2)	22–30
Gamma 1 (γ_1)	30–40
Gamma 2 (γ_1)	40–50

Table 2
The correspondence between brain waves and sleep stages.

Sleep stage	Brain waves
W	α, β
N1	α, θ
N2	k complex, spindle wave, δ, θ
N3	δ, spindle wave
REM	α, β, θ

FC layers are used to aggregate these information. Besides, we utilize the ReLU and sigmoid activation functions after the two fully connected layers, respectively. Their role is to make a non-linear mapping of the global features in the previous step. The specific formulas are expressed as follows:

$$z_n = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} M_n(i,j), \tag{1}$$

$$Z = (z_1, z_2, \ldots, z_n), \tag{2}$$

$$C_r = \delta(W_1 Z), \tag{3}$$

$$C_s = \sigma(W_2 C_r), \tag{4}$$

$$O_{se} = M \bigotimes C_s, \tag{5}$$

where for adaptive average pooling, $H = 1$, M_n is one of the feature maps, z_n is the average eigenvalue of M_n, and W_1 and W_2 refer to the two FC layers, C_r and C_s are instantiated by two FC layers with different neurons and different activation functions (i.e., σ: sigmoid and δ: ReLU). O_{se} is the output, $\bigotimes$ means the point-wise multiplication between M and C_s.

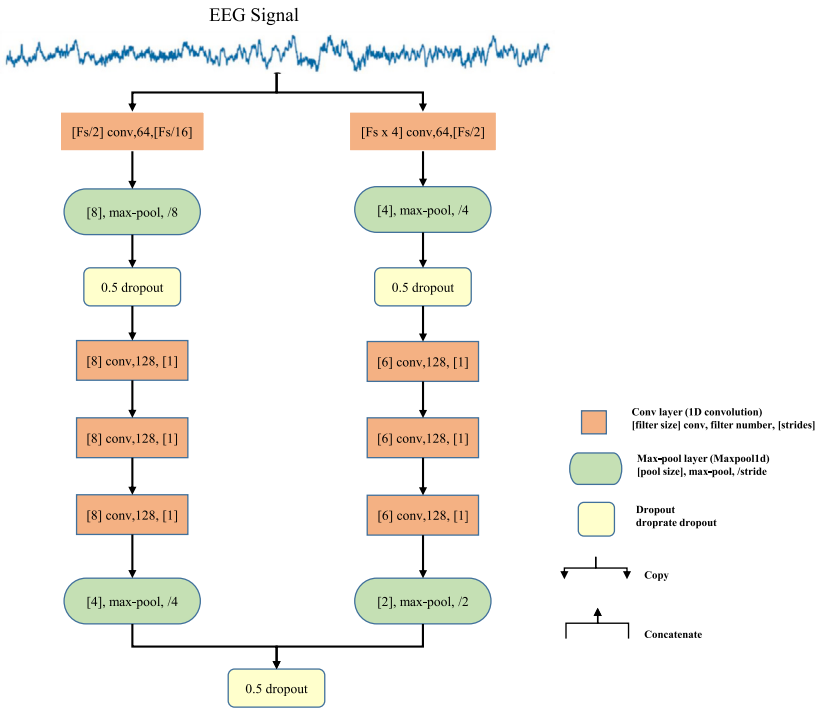

Fig. 2. The specific architecture of MSCNN.

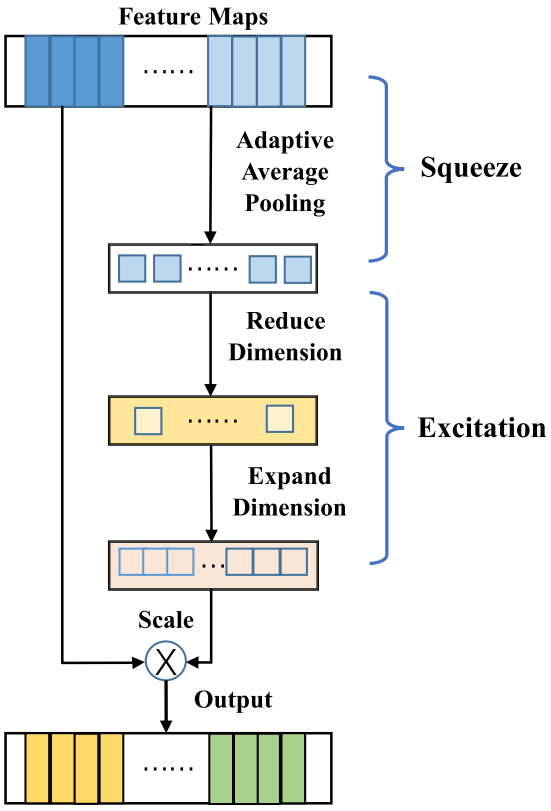

Fig. 3. The architecture of SE block.

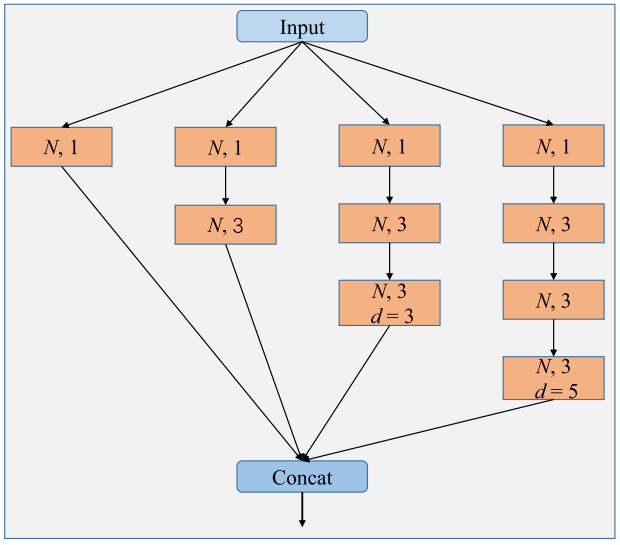

Fig. 4. Basic Framework of MSDCB. Where d represents the dilated rate, and N refers to the number of channels.

3.2.3. Multi-scale dilated convolutional block (MSDCB)

The larger the receptive field of the model, the more comprehensive it can extract different frequency components and complex features from the EEG signals. To expand the receptive field of our model, we use a multi-scale dilated convolution block (MSDCB). MSDCB is composed of convolutional layers of different depths and dilation rates in Fig. 4. MSDCB consists of four branches, and the depth of each branch increases from left to right. We use a convolutional layer with a convolution kernel size of 1×1 in the first layer of each branch, which improves the nonlinear capability of the module. As for the filter size of the rest of the convolutional layers of each branch, we set them to 3×3. Dilated convolution enables small convolution kernel sizes to achieve larger receptive fields by using the dilation rate without increasing a large number of convolutional layers and the size of the convolutional kernels. It allows our model to obtain the maximum receptive field at a small computational cost.

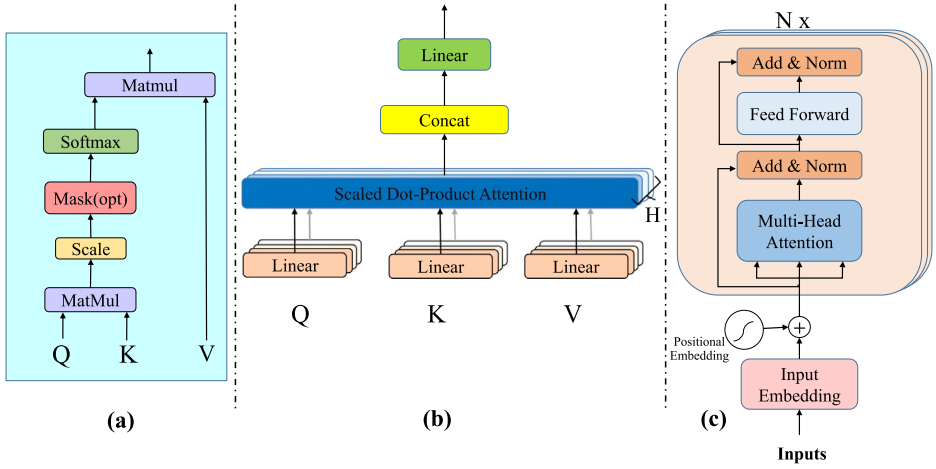

Fig. 5. (a) scaled dot-product attention, (b) multi-head attention, and (c) Transformer encoder.

3.3. Global temporal context encoder

In the global temporal context encoder as shown in Fig. 5, Transformer is adopted to extract the global temporal context relationships among EEG features. In this article, we adopt the encoder structure of Transformer. It is composed of two important modules: the multi-head attention module and the feedforward connection layer module. The multi-head attention module adopts the scaled dot-product mechanism to correlate different position elements of an input sequence and finally exports the output sequence. Multi-head attention is made up of H scaled dot-product attention modules. First, the input is fed to H linear coding layers to obtain queries, keys, and values. Then, these queries, keys, and values are processed by H scaled dot-product attention modules. Finally, the H attention heads are connected and then linearly projected to produce attention output. The above steps are capable of expressing as follows:

$$Q_i = ZW_i^Q, K_i = ZW_i^K, V_i = ZW_i^V, 1 \le i \le H, \tag{6}$$

$$X_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d}})V_i, \tag{7}$$

$$Y_o = \text{Concat}(X_1, X_2, \dots, X_H)W^y, \tag{8}$$

where $Z \in R^{(l\times d)}$ is the input with length l and dimension d. $Q_i, K_i, V_i \in R^{(l\times(\frac{d}{H}))}$ represent the mapped queries, keys, and values, respectively. $X_i \in R^{(l\times(\frac{d}{H}))}$ is ith the attention head. $W_i^Q, W_i^K, W_i^V \in R^{(l\times(\frac{d}{H}))}$ and $W^y \in R^{(d\times d)}$ are the updatable weight matrices. $Y_o \in R^{(d\times d)}$ is the output of the module.

The feedforward connection layer module is made up of two fully connected (FC) layers and it is worth noting that there is a ReLU activation function in the middle of the two layers. In addition to two main modules, Transformer has two normalization layers. Layer normalization is the key to fast convergence and stable training of Transformer. In summary, it can be described as follows:

$$LN = \text{LayerNorm}(Z + Y_o), \tag{9}$$

$$FFN = \text{Relu}((LN)W_1 + b_1)W_2 + b_2, \tag{10}$$

$$Out = \text{LayerNorm}(LN + FFN), \tag{11}$$

where FFN is the output of the feedforward connection layer module. W_1, W_2, and b_1, b_2 are the weight and bias in the fully connected layer (FC).

Since Transformer can calculate the input data in parallel, it improves the computational efficiency while also losing the position relationship between the input data. This is not conducive to obtaining global temporal context information of EEG features, so we use relative positional encoding to make the input data have positional information. Relative position encoding is shown by the following formulas:

$$PE_{(pos,2i)} = \sin\frac{pos}{10000^{\frac{2i}{d}}}, \tag{12}$$

$$PE_{(pos,2i+1)} = \cos\frac{pos}{10000^{\frac{2i}{d}}}, \tag{13}$$

where PE stands for position coding, and pos represents the location of each embedded feature, d refers to the dimension of the vector, i denotes the index of the current embedding feature. The $\sin(\cdot)$ function is used in the even position and the $\cos(\cdot)$ function is used in the odd position.

3.4. Spatial position relationship encoder

In order to enable the model to learn spatial information between EEG features, we design a capsule network module. In the capsule network module as shown in Fig. 6, first, to capture the potential spatial location relationship between features in feature maps, we convert an EEG feature map into multiple capsules using convolutional layers with 128 convolution kernels and kernel size of 1×1. This operation allows us to determine the number of capsules (i.e., 128). Meanwhile, we set the length of the capsules to 8. Second, we use a dynamic routing mechanism as shown in Fig. 7 to better capture the spatial information between different channels. The calculation steps of the dynamic routing mechanism are as follows:

First, we do an affine transformation of the capsule to encode the position information, that is, multiply the ith capsule $u_i (i = 1, 2, \dots, M)$ by a learnable parameter matrix $W_{ij} (j = 1, 2, \dots, L)$. M refers to the number of capsules. L means the length of output capsules. The formula is below:

$$\hat{u}_{j|i} = W_{ij}u_i, \tag{14}$$

where $\hat{u}_{j|i}$ is the vector transformed by affine.

Second, we use a dynamic routing mechanism to process vectors after affine transformation. The specific formula is as follows:

$$s_j = \sum_i^N c_{ij}\hat{u}_{j|i}, \tag{15}$$

where c_{ij} is the routing weight between the ith input capsule and the jth output capsule. It is taken from trainable parameter b_{ij}. The formula can be expressed as follows:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \tag{16}$$

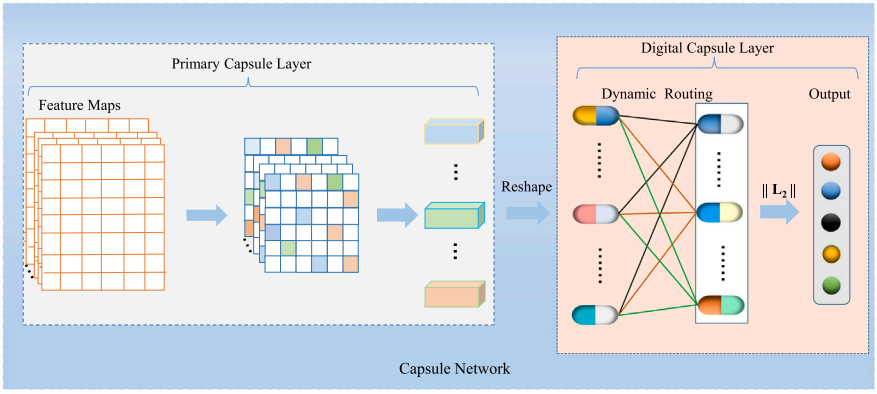

Fig. 6. Architecture of capsule network.

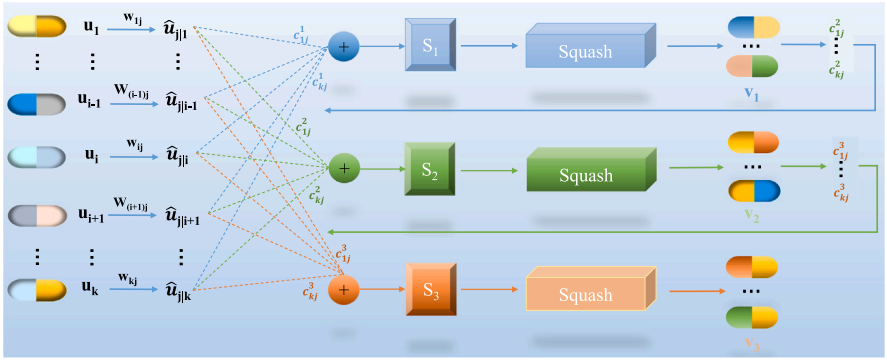

Fig. 7. Dynamic routing mechanism. The number of routing is set to 3.

At last, we utilize the nonlinear function "squashing" to ensure that the length of output vector V_j can be compressed between 0 and 1 while keeping the direction of the vector constant. The formula is expressed as follows:

$$v_j = \frac{\|s_j\|_2^2}{1+\|s_j\|_2^2}\frac{s_j}{\|s_j\|_2}, \tag{17}$$

where s_j denotes the jth capsule vector after weighted summation, $\|s_j\|_2$ represents the output of s_j after ℓ_2-norm.

In addition, the initial value of trainable parameter $b_{ij} = 0$, and it is able to be optimally updated by the dynamic routing mechanism. The formula is as follows:

$$b_{ij} \longleftarrow b_{ij} + v_j \cdot \hat{u}_{j|i}. \tag{18}$$

If v_j and $\hat{u}_{j|i}$ produce a large product, which indicates that there is a strong connection between the feature $\hat{u}_{j|i}$ and the vector v_j, then the weight of the capsule vector will be increased in the next routing. Notably, the greater weight of the capsule vector indicates that the capsule can better represent a certain sleep stage.

3.5. Classification

We use the ℓ_2-norm to process the capsules of the dynamic routing mechanism and take the output as the classification result of our model.

3.6. WCE loss

Due to the EEG data having the problem of class imbalance, we choose the weighted cross-entropy (WCE) loss function to calculate the loss of our model. The specific description is as follows:

$$L_{WCE}(\theta) = -\frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N} w_k y_i^k \log(\hat{y}_i^k) + \frac{\lambda}{2}\|\theta\|_2^2, \tag{19}$$

Table 3
Data introduction.

Dataset	W	N1	N2	N3	REM	Total
Sleep-EDF-20	8285 19.6%	2804 6.6%	17799 42.1%	5703 13.5%	7717 18.2%	42308
Sleep-EDF-78	65951 33.7%	21522 11.1%	69132 35.4%	13039 6.7%	25835 13.2%	195479
SHHS	46319 41.3%	10304 3.2%	142125 43.7%	60153 18.5%	65953 20.3%	324854

$$w_k = \frac{N}{N_k}, \tag{20}$$

where y_i^k means the true label and $\hat{y}_i^k$ denotes the label of the prediction, N is the total sample size and K is a quantity of categories, N_k is a quantity of samples in category k. To prevent overfitting, we adopt the ℓ_2-norm regularization term after WCE loss. λ is a hyperparameter.

4. Experiments

In this portion, first, we explain the three datasets. Then we illustrate the input to our model. Subsequently, we show the experimental details. At last, we present the results of our proposed CTCNet assessment.

4.1. Data sets

As shown in Table 3, in this paper, we use the three datasets: Sleep-EDF-20, Sleep-EDF-78, and SHHS.

4.1.1. Sleep-EDF-20 dataset

Sleep-EDF-20 dataset is acquired from PhysioBank [49]. It consists of 39 PSG records from 20 healthy adult Caucasians between the ages

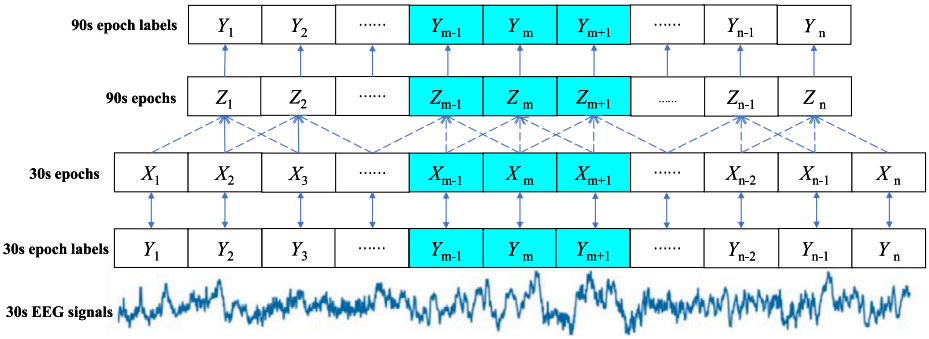

Fig. 8. The Input to our model.

of 25 and 34. In this dataset, everyone's data records contained two nights except subject 13 (who had only one night's record). Sleep-EDF-20 dataset is composed of two parts: the Sleep Cassette (SC) group and the Sleep Telemetry (ST) group. We use data from the Sleep Cassette segment.

Each PSG data is segmented into 30s segments and it contains various signals: Dual-channel EEG signals, namely, Fpz-Cz channel and Pz-Cz channel, one horizontal EOG signal, one EMG signal, and one oro-nasal respiration signal. In addition, EEG and EOG have a sampling frequency of 100 Hz. Sleep specialists manually classify sleep stages into the following categories based on the R&K standard [10] : W, N1, N2, N3, N4, REM, MOVEMENT, and UNKNOWN. Following prior researches [21,50], we choose EEG signals with a sampling rate of 100 Hz for the Fpz-Cz channel.

4.1.2. Sleep-EDF-78 dataset

Sleep-EDF-78 dataset includes 153 PSG records from 78 subjects between the ages of 25 and 101. Except for 13, 36, and 52 subjects who have only one night of recording, the rest have two full records. Apart from these differences, Sleep-EDF-78 is exactly the same as Sleep-EDF-20.

4.1.3. SHHS dataset

SHHS is a multi-center cohort study. The subjects selected for this study all have a variety of medical conditions, such as lung disease, cardiovascular disease, and so on. For the successful implementation of our experiments, subjects who meet our requirements are selected following the previous research in [51]. Finally, we select 329 subjects. Notably, we adopt the C4-A1 channel with a sampling rate of 125 Hz.

In this experiment, we follow these steps to preprocess data that belongs to three datasets:

(1) According to the studies in [23,24], we use only 30-min EEG data before and after sleep on the three datasets.
(2) We remove all MOVEMENT stages and UNKNOWN stages.
(3) We expand the N3 stages by merging the N4 stages into the N3. Finally, the sleep is separated into 5 stages, namely, W, N1, N2, N3, REM.

4.2. The input to our model

In previous work, most researchers have used a single 30s epoch as input. Although this approach is direct and convenient, it does not take into account the correlation and dependence between adjacent epochs. Studies have shown that the classification of sleep stages depends not only on current epoch but also on the epochs in which it is adjacent [20,52]. So we take three consecutive 30s epochs: X_{m-1}, X_m, X_{m+1} $(m = 2, 3 \ldots, n)$ (i.e., 90s epoch) as the input. Besides, we use the X_m-mapped label as the label for the 90s epoch. Details are illustrated in Fig. 8.

4.3. Implementation details

We use version 1.5 of PyTorch and the processor of an NVIDIA GTX3060 GPU to train the CTCNet model. Subsequently, we decide to introduce our experiments as follows:

4.3.1. Baselines

In this experiment, we adopted six baselines: DeepSleepNet [23], SleepEEGNet [53], ResnetLSTM [54], MultitaskCNN [20], SeqSleepNet [55] and AttnSleepNet [56]. Then, we give them a brief introduction as follows:

(1) DeepSleepNet [23] combined CNN and Bi-LSTM to perform sleep stage classification.
(2) SleepEEGNet [53] combined CNN with the multi-head attention mechanism to complete the sleep stage classification task.
(3) ResnetLSTM [54] used a ResNet to perform feature extraction of EEG signals and then used LSTM to classify the extracted features.
(4) MultitaskCNN [20] first converted EEG signals into images, then used CNN to identify sleep stages.
(5) AttnSleepNet [56] combined CNN with attention mechanism to classify EEG signals.
(6) SeqSleepNet [55] converted EEG signals into images and then used RNN to classify sleep stages.

4.3.2. Experimental setup

In this paper, for a fair comparison with other models, we set the k value for k-fold cross-validation to 20 [23,56]. The division of the datasets is able to affect the performance of the models, so in this article, we divide the datasets as shown in Table 4. We select different subjects as the test set in each fold and randomly select some subjects as the training set and the remainder as the validation set, which ensure that all subjects can be fully utilized, avoiding model performance degradation caused by poor data set division. Since our context input is three consecutive epochs (90s epoch), according to different sampling rates, the amount of data for one epoch of Sleep-EDF is 9000 (30 s × 100 Hz × 3) and the amount of data for one epoch of SHHS is 11 250 (30 s × 125 Hz × 3). In our experiments, we take 128 epochs as input to our model, and if the remaining data is less than 128, all of them are used as input to the model. It allows the data to be fully utilized.

To prevent overfitting during training, we employ some solutions. The first solution is L2 regularization. We add it to the loss function and set the regularization rate (λ) to 10^{-3}. Another method we used is dropout (dropout rate = 0.5). In addition, we adopt an adaptive average pooling layer behind the Transformer module to reduce the amount of model parameters. We train our model with random initialization and use the cosine scheme to adjust the learning rate of the model. We use the SGD optimizer and set its learning rate to 0.005, weight decay to 0.006, and momentum to 0.99. The training epochs in our experiment are set to 200.

Table 4
The details of the division of the three datasets.

Dataset	The number of subjects	The type of fold	Training set	Validation set	Test set	Array shape
Sleep-EDF-20	20	20-fold	17	2	1	$128 \times 1 \times 9000$
Sleep-EDF-78	78	20-fold	72	4	2	$128 \times 1 \times 9000$
SHHS	329	20-fold	310	12	7	$128 \times 1 \times 11\,250$

Table 5
CTCNet confusion matrix obtained on Fpz-Cz channel of Sleep-EDF-20.

Predicted						Pre-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	7467	467	107	39	205	94.3	90.2	92.2	93.8
N1	309	1722	458	60	255	55.3	60.0	57.6	74.8
N2	103	369	15 884	763	680	91.5	88.2	89.8	89.9
N3	13	5	331	5332	22	86.6	93.5	89.6	96.1
REM	60	517	605	4	6531	81.1	84.6	82.8	91.8

Table 6
CTCNet confusion matrix obtained on Fpz-Cz channel of Sleep-EDF-78.

Predicted						Pre-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	59 510	4781	429	54	1177	94.8	90.2	92.5	94.2
N1	1806	13 144	3925	79	2568	48.1	61.1	53.8	75.2
N2	363	5541	59 003	1335	2890	88.3	85.4	86.8	89.5
N3	39	73	1621	11 282	24	87.9	86.6	87.3	92.7
REM	639	3832	1898	91	19 375	74.5	75.1	74.8	84.9

Table 7
CTCNet confusion matrix obtained on C4-A1 channel of SHHS.

Predicted						Pre-class metrics			
	W	N1	N2	N3	REM	PR	RE	F1	GM
W	39 877	2386	1500	434	2122	89.4	84.8	87.1	91.3
N1	1067	4299	1445	239	3254	30.8	36.9	33.6	59.8
N2	1983	3002	124 773	3858	8509	91.2	87.8	89.5	90.6
N3	416	6	5502	53 881	348	92.1	89.6	90.9	93.9
REM	1406	3273	4111	302	56 861	80.7	86.3	83.5	85.3

4.3.3. Evaluation metrics

In this paper, we adopt several common metrics to evaluate the above models: the accuracy (ACC), macro-averaged F1-score (MF1), Cohen Kappa (k) [57], and the macro-averaged G-mean (MGm). MGM and MF1 are used to better assess various models that are on imbalanced datasets [58]. In addition, to objectively and intuitively display the advantages and disadvantages of the models, we also adopt the true positive (TP), false positive (FP), true negative (TN), false negative (FN), precision (PR), recall (RE) and specificity (SP). The formulas for ACC, MF1, k, and MGm are able to be described as follows:

$$Acc = \frac{\sum_{i=1}^{N} TP_i}{S}, \tag{21}$$

$$MF1 = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \times PR_i \times RE_i}{PR_i + RE_i}, \tag{22}$$

$$k = \frac{P_o - P_e}{1 - P_e}, \tag{23}$$

$$MGm = \frac{1}{N}\sum_{i=1}^{N}\sqrt{SP_i \times RE_i}, \tag{24}$$

where i represents stage, S denotes the all sample size and N refers to sleep stages. The formulas for remainder evaluation metrics are the following:

$$PR_i = \frac{TP_i}{TP_i + FP_i}, \tag{25}$$

$$RE_i = \frac{TP_i}{TP_i + FN_i}, \tag{26}$$

$$P_o = ACC, \tag{27}$$

$$P_e = \frac{\sum_{i=1}^{N}(TP_i + FP_i) \times (TP_i + FN_i)}{S^2}, \tag{28}$$

$$SP_i = \frac{TN_i}{TN_i + FP_i}. \tag{29}$$

4.4. Results and comparison

4.4.1. Scoring performance of CTCNet

We use four metrics to evaluate our model: per-class precision (PR), per-class recall (RE), per-class F1-score (F1), and per-class G-mean (GM). There are three confusion matrixes in Tables 5, 6, and 7.

Each row of the three confusion matrixes means the sample size of the experts' classification, and each column denotes the sample size predicted by our model.

4.4.2. Comparison with state-of-the-art methods

We compare our model with six baselines on the three datasets using ACC, MF1, k, and MGm. As you can see from Table 8, CTCNet performs best on many evaluation metrics. CTCNet implements the best MF1 and MGm on Sleep-EDF-20 and Sleep-EDF-78. It shows that our model is helpful in solving the class imbalance problem. Our proposed model shows great performance on Sleep-EDF-20 compared to the other six models. Its classification accuracy is 0.6% higher than AttenSleep. On Sleep-EDF-78, CTCNet had slightly lower accuracy than AttenSleep and SeqSleepNet, with the highest F1 score at 53.8%, 86.8%, and 87.1% in N1, N2, and N3, respectively. On SHHS, our model gets the best F1 score in N2, N3, at 89.5%, 90.9%. However, the F1 score is worse in N1, as shown in Table 7, CTCNet is inclined to misclassify N1 as N2 and N3.

4.4.3. Ablation study

Our model is roughly made up of five parts: MSCNN, ACFR, MSDCB, a Transformer module, and a capsule network module. To gain a deeper understanding of the role played by each part of our model, we design the following 5 sets of experiments on three datasets.

(1) experiment 1: MSCNN
(2) experiment 2: MSCNN + ACFR
(3) experiment 3: MSCNN + ACFR + MSDCB
(4) experiment 4: MSCNN + ACFR + MSDCB + Transformer
(5) experiment 5: CTCNet (our model)

We can draw four conclusions from the ablation experiments as shown in Figs. 9–11. First, by comparing the results of experiment 1 and experiment 2, it is shown that ACFR discovers the relationship between channel-wise features in feature maps generated by MSCNN and explicitly recalibrates features to promote useful features and features that are less inhibitive. Second, comparing the results of experiment 2 and experiment 3, it can be seen that MSDCB expands the receptive fields of the model by using dilated convolution, which enables the model to enhance its ability to various frequency components and complex features. Then, according to the results of experiment 3 and experiment 4, it is proved that Transformer's multi-head attention mechanism makes it easy to learn global temporal context features. Capturing

Table 8
Comparison results between CTCNet and six baselines.

Dataset	Per-class F1-score						Overall metrics		
	Method	W	N1	N2	N3	REM	Accuracy	MF1	k	MGm
Sleep-EDF-20	DeepSleepNet	86.7	45.5	85.1	83.3	82.6	81.9	76.6	0.76	86.9
	SleepEEGNet	89.4	44.4	84.7	84.6	79.6	81.5	76.6	0.75	85.3
	RestnetLSTM	86.5	28.4	87.7	89.8	76.2	82.5	73.7	0.76	81.8
	MultitaskCNN	87.9	33.5	87.5	85.8	79.0	83.1	75.0	0.77	83.1
	AttnSleep	90.3	47.9	89.8	89.0	85.0	85.6	80.9	0.80	88.2
	SeqSleepNet	87.7	43.8	88.2	86.5	84.0	84.6	78.0	0.79	85.3
	CTCNet (*ours*)	92.2	57.6	89.8	89.6	82.8	86.2	82.5	0.82	89.3
Sleep-EDF-78	DeepSleepNet	90.9	45.5	79.2	72.7	71.1	77.8	71.8	0.70	81.6
	SleepEEGNet	89.9	42.1	75.2	70.4	70.6	74.2	69.6	0.66	82.3
	RestnetLSTM	90.7	34.7	83.6	80.9	67.0	78.9	71.4	0.71	80.8
	MultitaskCNN	90.9	39.7	83.2	76.6	73.5	79.6	72.8	0.72	82.5
	AttnSleep	92.6	47.4	85.5	83.7	81.5	82.9	78.1	0.77	85.6
	SeqSleepNet	91.8	46.0	85.0	77.5	81.0	82.6	76.3	0.76	84.3
	CTCNet (*ours*)	92.5	53.8	86.8	87.3	74.8	82.5	79.1	0.78	87.3
SHHS	DeepSleepNet	85.4	40.5	82.5	79.3	81.9	81.0	73.9	0.73	82.6
	SleepEEGNet	81.3	34.4	73.4	75.9	77.0	73.9	68.4	0.65	82.7
	RestnetLSTM	85.1	9.4	86.3	87.0	79.1	83.3	69.4	0.76	76.4
	MultitaskCNN	82.2	25.7	83.9	83.3	81.1	81.4	71.2	0.74	80.4
	AttnSleep	88.3	46.3	88.7	87.6	87.4	86.6	79.7	0.81	87.9
	SeqSleepNet	84.2	47.3	87.2	85.4	88.6	85.6	78.5	0.80	85.4
	CTCNet (*ours*)	87.1	33.6	89.5	90.9	83.5	85.7	77.2	0.81	85.3

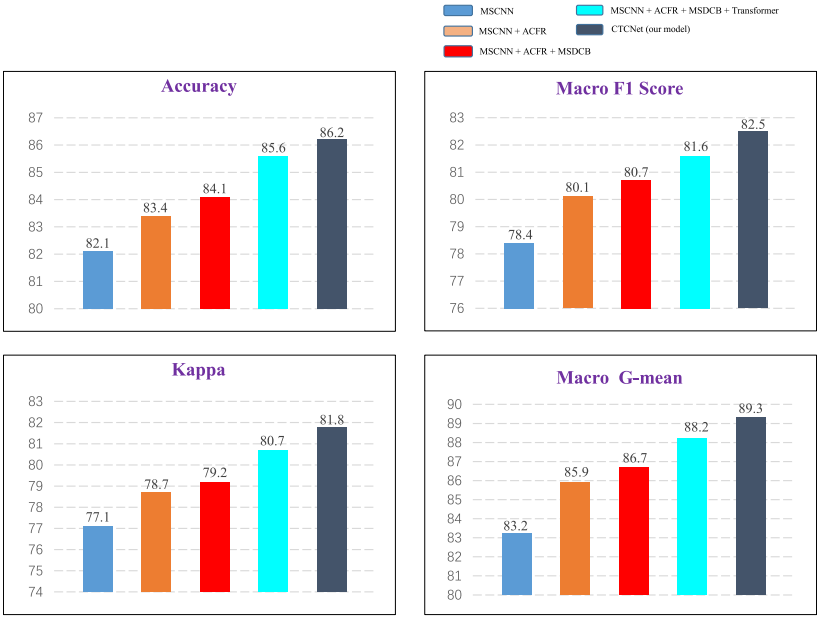

Fig. 9. Ablation study on Sleep-EDF-20.

time dependencies with Transformer goes a long way in enhancing our model's performance. At last, from experiments 4 and 5, we know that using capsule network to extract spatial position relationships between EEG features of different channels is able to be helpful in enhancing the classification ability of our model.

5. Discussions

EEG signals are complex, and they are also nonlinear and non-stationary from the waveform of the signals [59]. Owing to CNN performing well in feature extraction, it is first used to classify sleep stages. After many experiments, CNN has been shown to be useful in improving the accuracy of sleep stage classification. It is found that the convolution kernel only focuses on a small area of the input data, which allows CNN to capture local features of the data well. But this also makes it difficult for CNN to extract global temporal context characteristics of EEG signals. So the researchers adopt RNN into the field of EEG. RNN can learn temporal context information well. However, it lacks the ability to process data in parallel, leading to low computational efficiency. The structural characteristics of Transformer make it possible to process input data in parallel, which greatly improves the training efficiency. In addition, EEG features usually include multiple attributes such as frequency and time, and due to CNN use of scalar neurons, it cannot acquire spatial relationships between these features. Therefore, we propose a novel model called CTCNet.

Our model is made up of four main parts: (1) local feature extraction, (2) global temporal context encoder, (3) Spatial position relationship encoder, and (4) classification. To extract features, first, we use convolutional blocks with different convolution kernel sizes to extract EEG signals of different frequencies, subsequently, adaptive channel feature recalibration (ACFR) to obtain important channel features, and finally, multi-scale dilated convolution block (MSDCB) to extract complex EEG features. In order to obtain global temporal context information of EEG features, we used Transformer as the second part of

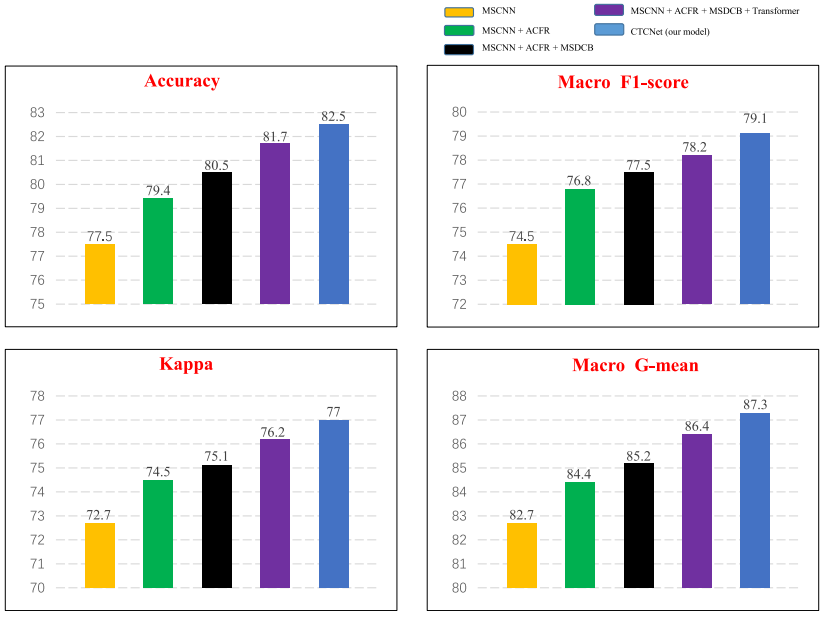

Fig. 10. Ablation study on Sleep-EDF-78.

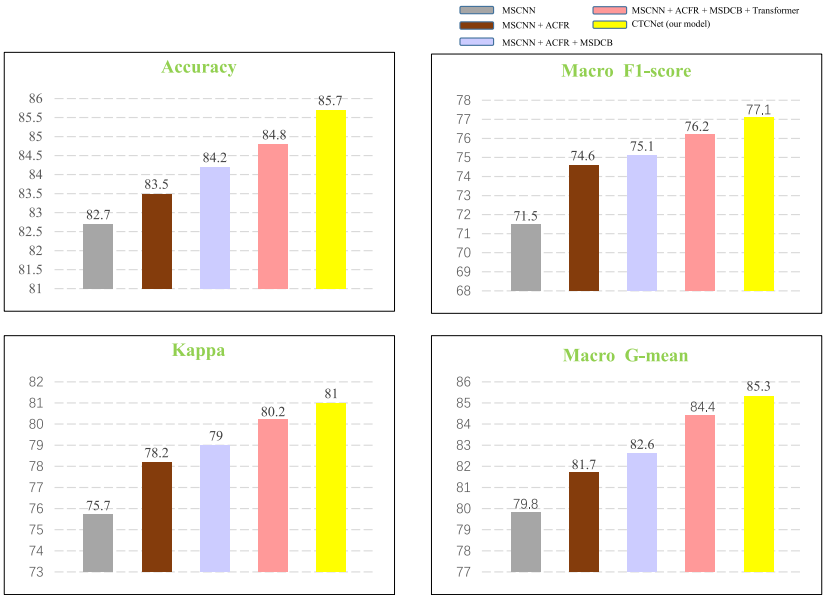

Fig. 11. Ablation study on SHHS.

the model. The Transformer module can better obtain the global time context information of EEG features than RNN and can process data in parallel, which greatly improves computational efficiency. The third part is made up of a capsule network, which can well obtain spatial position relationship information in EEG features due to its unique dynamic routing mechanism. It is worth highlighting we also creatively use the capsule network as a classifier for our model.

The experimental results from Tables 5, 6, 7, and 8 reveal that compared with the six models, all evaluation results of CTCNet are optimal on Sleep-edf-20, and on Sleep-edf-78 and SHHS the classification accuracy of CTCNet is slightly lower than AttnSleepNet, but the rest of the evaluation results are the best, especially the recognition accuracy of the N1 stage is much higher than other models. This indicates that our model performs well in handling class-imbalanced datasets. However, there is a decrease in the recognition accuracy of REM. It may be due to we overfocus on the least classified N1 and neglect the others. In addition, the subjective judgment of experts has led to inconsistent scoring results. Nevertheless, our model achieves an average Kappa of 0.8 on these three publicly available datasets. Generally speaking, our model can achieve comparable or better performance than state-of-the-art deep learning methods on the three datasets.

6. Conclusion

We propose a novel structure called CTCNet which uses single-channel raw EEG signals. The contribution of CTCNet is to combine CNN, Transformer, and Capsule Network. CNN has advantages in the extraction of local features, Transformer captures global time context information well and capsule network is capable of extracting spatial position relationships between features. Besides, we use capsule network as a classifier to deal with the final results. The results show that CTCNet reaches an advanced level under different evaluation matrices.

Besides, it demonstrates the reliability of our model. In follow-up studies, we may convert our research focus to models with multi-modal physiological signals as input.

Funding

This work is supported by the National Natural Science Foundation of China (Grants 62071165, 41901350, 32271431, 32150017), and the Fundamental Research Funds for the Central Universities of China (Grant PA2023IISL0095).

CRediT authorship contribution statement

Weijie Zhang: Writing – original draft, Software, Methodology, Conceptualization. **Chang Li:** Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Hu Peng:** Visualization, Formal analysis, Data curation. **Heyuan Qiao:** Writing – review & editing, Validation. **Xun Chen:** Supervision, Project administration.

Declaration of competing interest

We would like to confirm that all authors were fully involved in the study and preparation of the manuscript. None of this work has been previously published, or been pending publication in another journal, or been under review in any other journal. None of the authors have a conflict of interest.

Data availability

The authors do not have permission to share data.

References

[1] B. Van Alphen, E.R. Semenza, M. Yap, B. Van Swinderen, R. Allada, A deep sleep stage in drosophila with a functional role in waste clearance, Sci. Adv. 7 (4) (2021) eabc2999.

[2] S.A. Keenan, Chapter 3 an overview of polysomnography, in: C. Guilleminault (Ed.), Handbook of Clinical Neurophysiology, Vol. 6, Elsevier, 2005, pp. 33–50, http://dx.doi.org/10.1016/S1567-4231(09)70028-0, URL https://www.sciencedirect.com/science/article/pii/S1567423109700280.

[3] C. Li, Y. Zhao, R. Song, X. Liu, R. Qian, X. Chen, Patient-specific seizure prediction from electroencephalogram signal via multi-channel feedback capsule network, IEEE Trans. Cogn. Dev. Syst. 15 (3) (2023) 1360–1370.

[4] X. Chen, C. Li, A. Liu, M.J. McKeown, R. Qian, Z.J. Wang, Toward open-world electroencephalogram decoding via deep learning: A comprehensive survey, IEEE Signal Process. Mag. 39 (2) (2022) 117–134.

[5] M. Xu, X. Xiao, Y. Wang, H. Qi, T.-P. Jung, D. Ming, A brain–computer interface based on miniature-event-related potentials induced by very small lateral visual stimuli, IEEE Trans. Biomed. Eng. 65 (5) (2018) 1166–1175.

[6] C. Li, X. Huang, R. Song, R. Qian, X. Liu, X. Chen, EEG-based seizure prediction via transformer guided CNN, Measurement 203 (2022) 111948, http://dx.doi.org/10.1016/j.measurement.2022.111948, URL https://www.sciencedirect.com/science/article/pii/S0263224122011447.

[7] C. Li, C. Shao, R. Song, G. Xu, X. Liu, R. Qian, X. Chen, Spatio-temporal MLP network for seizure prediction using EEG signals, Measurement 206 (2023) 112278.

[8] T. Chen, S. Ju, F. Ren, M. Fan, Y. Gu, EEG emotion recognition model based on the LIBSVM classifier, Measurement 164 (2020) 108047.

[9] R.B. Berry, R. Brooks, C.E. Gamaldo, S.M. Harding, C. Marcus, B.V. Vaughn, et al., The AASM manual for the scoring of sleep and associated events, Rules Terminol. Tech. Specif. Darien Ill. Am. Acad. Sleep Med. 176 (2012) 2012.

[10] E.A. Wolpert, A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects., Arch. Gen. Psychiatry 20 (2) (1969) 246–247.

[11] T. Sudhakar, G.H. Krishnan, N. Krishnamoorthy, B. Janney, M. Pradeepa, J. Raghavi, Sleep disorder diagnosis using EEG based deep learning techniques, in: 2021 Seventh International Conference on Bio Signals, Images, and Instrumentation (ICBSII), IEEE, 2021, pp. 1–4.

[12] A. Malhotra, M. Younes, S.T. Kuna, R. Benca, C.A. Kushida, J. Walsh, A. Hanlon, B. Staley, A.I. Pack, G.W. Pien, Performance of an automated polysomnography scoring system versus computer-assisted manual scoring, Sleep 36 (4) (2013) 573–582.

[13] T. Chen, C. Guestrin, XGBoost, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, http://dx.doi.org/10.1145/2939672.2939785.

[14] G. Zhu, Y. Li, P. Wen, Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal, IEEE J. Biomed. Health Inform. 18 (6) (2014) 1813–1821.

[15] S. Seifpour, H. Niknazar, M. Mikaeili, A.M. Nasrabadi, A new automatic sleep staging system based on statistical behavior of local extrema using single channel EEG signal, Expert Syst. Appl. 104 (2018) 277–293.

[16] P. Memar, F. Faradji, A novel multi-class EEG-based sleep stage classification system, IEEE Trans. Neural Syst. Rehabil. Eng. 26 (1) (2017) 84–95.

[17] X. Li, L. Cui, S. Tao, J. Chen, X. Zhang, G.-Q. Zhang, Hyclasss: a hybrid classifier for automatic sleep stage scoring, IEEE J. Biomed. Health Inform. 22 (2) (2017) 375–385.

[18] S.A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Syst. Man Cybern. (4) (1976) 325–327.

[19] S. Bhattacharyya, A. Khasnobish, S. Chatterjee, A. Konar, D. Tibarewala, Performance analysis of LDA, QDA and KNN algorithms in left-right limb movement classification from EEG data, in: 2010 International Conference on Systems in Medicine and Biology, IEEE, 2010, pp. 126–131.

[20] H. Phan, F. Andreotti, N. Cooray, O.Y. Chén, M. De Vos, Joint classification and prediction CNN framework for automatic sleep stage classification, IEEE Trans. Biomed. Eng. 66 (5) (2018) 1285–1296.

[21] O. Tsinalis, P.M. Matthews, Y. Guo, S. Zafeiriou, Automatic sleep stage scoring with single-channel EEG using convolutional neural networks, 2016, arXiv preprint arXiv:1610.01683.

[22] H. Phan, F. Andreotti, N. Cooray, O.Y. Chén, M.D. Vos, Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018, pp. 1452–1455, http://dx.doi.org/10.1109/EMBC.2018.8512480.

[23] A. Supratak, H. Dong, C. Wu, Y. Guo, DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (11) (2017) 1998–2008.

[24] H. Phan, K. Mikkelsen, O.Y. Chén, P. Koch, A. Mertins, M. De Vos, Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification, IEEE Trans. Biomed. Eng. 69 (8) (2022) 2456–2467.

[25] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[26] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).

[27] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[29] A.M. Roy, An efficient multi-scale CNN model with intrinsic feature integration for motor imagery EEG subject classification in brain-machine interfaces, Biomed. Signal Process. Control 74 (2022) 103496, http://dx.doi.org/10.1016/j.bspc.2022.103496, URL https://www.sciencedirect.com/science/article/pii/S1746809422000180.

[30] G. Dai, J. Zhou, J. Huang, N. Wang, HS-CNN: a CNN with hybrid convolution scale for EEG motor imagery classification, J. Neural Eng. 17 (1) (2020) 016025.

[31] W. Mao, H. Fathurrahman, Y. Lee, T. Chang, EEG dataset classification using CNN method, in: Journal of Physics: Conference Series, Vol. 1456, IOP Publishing, 2020, 012017.

[32] G. Kong, C. Li, H. Peng, Z. Han, H. Qiao, EEG-based sleep stage classification via neural architecture search, IEEE Trans. Neural Syst. Rehabil. Eng. 31 (2023) 1075–1085.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[34] A.M. Braşoveanu, R. Andonie, Visualizing transformers for nlp: a brief survey, in: 2020 24th International Conference Information Visualisation (IV), IEEE, 2020, pp. 270–279.

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

[36] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[39] J. Sun, X. Wang, K. Zhao, S. Hao, T. Wang, Multi-channel EEG emotion recognition based on parallel transformer and 3D-convolutional neural network, Mathematics 10 (17) (2022) 3131.

[40] Z. Wang, Y. Wang, C. Hu, Z. Yin, Y. Song, Transformers for EEG-based emotion recognition: A hierarchical spatial information learning model, IEEE Sens. J. 22 (5) (2022) 4359–4368.
[41] L. Gong, M. Li, T. Zhang, W. Chen, EEG emotion recognition using attention-based convolutional transformer neural network, Biomed. Signal Process. Control 84 (2023) 104835, http://dx.doi.org/10.1016/j.bspc.2023.104835, URL https://www.sciencedirect.com/science/article/pii/S1746809423002689.
[42] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, Adv. Neural Inf. Process. Syst. 30 (2017).
[43] M.E. Paoletti, J.M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, F. Pla, Capsule networks for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (4) (2018) 2145–2160.
[44] P. Gupta, M.K. Siddiqui, X. Huang, R. Morales-Menendez, H. Panwar, H. Terashima-Marin, M.S. Wajid, COVID-WideNet—A capsule network for COVID-19 detection, Appl. Soft Comput. 122 (2022) 108780.
[45] A. Jaiswal, W. AbdAlmageed, Y. Wu, P. Natarajan, Capsulegan: Generative adversarial capsule network, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.
[46] Y. Liu, Y. Wei, C. Li, J. Cheng, R. Song, X. Chen, Bi-CapsNet: A binary capsule network for EEG-based emotion recognition, IEEE J. Biomed. Health Inf. 27 (3) (2022) 1319–1330.
[47] J. Chen, Z. Han, H. Qiao, C. Li, H. Peng, EEG-based sleep staging via self-attention based capsule network with bi-LSTM model, Biomed. Signal Process. Control 86 (2023) 105351.
[48] Y. Liu, Y. Ding, C. Li, J. Cheng, R. Song, F. Wan, X. Chen, Multi-channel EEG-based emotion recognition via a multi-level features guided capsule network, Comput. Biol. Med. 123 (2020) 103927.
[49] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220.
[50] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, J.-F. Payen, A convolutional neural network for sleep stage scoring from raw single-channel EEG, Biomed. Signal Process. Control 42 (2018) 107–114.
[51] P. Fonseca, N. Den Teuling, X. Long, R.M. Aarts, Cardiorespiratory sleep stage detection using conditional random fields, IEEE J. Biomed. Health Inform. 21 (4) (2016) 956–966.
[52] R.B. Berry, R. Budhiraja, D.J. Gottlieb, D. Gozal, C. Iber, V.K. Kapur, C.L. Marcus, R. Mehra, S. Parthasarathy, S.F. Quan, et al., Rules for scoring respiratory events in sleep: update of the 2007 AASM manual for the scoring of sleep and associated events: deliberations of the sleep apnea definitions task force of the American academy of sleep medicine, J. Clin. Sleep Med. 8 (5) (2012) 597–619.
[53] S. Mousavi, F. Afghah, U.R. Acharya, SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach, PLoS One 14 (5) (2019) e0216456.
[54] Y. Sun, B. Wang, J. Jin, X. Wang, Deep convolutional network method for automatic sleep stage classification based on neurophysiological signals, in: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE, 2018, pp. 1–5.
[55] H. Phan, F. Andreotti, N. Cooray, O.Y. Chén, M. De Vos, SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (3) (2019) 400–410.
[56] E. Eldele, Z. Chen, C. Liu, M. Wu, C.-K. Kwoh, X. Li, C. Guan, An attention-based deep learning approach for sleep stage classification with single-channel EEG, IEEE Trans. Neural Syst. Rehabil. Eng. 29 (2021) 809–818.
[57] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46.
[58] Y. Tang, Y.-Q. Zhang, N.V. Chawla, S. Krasser, SVMs modeling for highly imbalanced classification, IEEE Trans. Syst. Man Cybern. B 39 (1) (2008) 281–288.
[59] D. Li, X. Li, Z. Liang, L.J. Voss, J.W. Sleigh, Multiscale permutation entropy analysis of EEG recordings during sevoflurane anesthesia, J. Neural Eng. 7 (4) (2010) 046010.