Biomedical Signal Processing and Control 86 (2023) 105351

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

EEG-based sleep staging via self-attention based capsule network with Bi-LSTM model

Jin Chen [a], Zhihui Han [a], Heyuan Qiao [a], Chang Li [a], Hu Peng [a,b,*]

[a] Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China
[b] Anhui Province Key Laboratory of Measuring Theory and Precision Instrument, School of Instrument Science and Optoelectronics Engineering, Hefei University of Technology, Hefei 230009, Anhui, China

ARTICLE INFO

Keywords:
Deep learning
Single-channel EEG
Sleep staging
Long short-term memory
Capsule network
Self-attention routing

ABSTRACT

Sleep staging via electroencephalogram is essential for determining the quality of sleep. Manual sleep staging is expensive and time-consuming. Recently, many deep learning-based sleep staging methods are demonstrated to outperform traditional methods. However, most methods do not fully exploit the temporal correlation between features of electroencephalogram signals. In this paper, we propose a self-attention routing-based capsule network with bi-directional long short-term memory model to extract more discriminative features from electroencephalogram signals and improve the accuracy of sleep staging. First, a convolutional neural network is used to extract salient features from the electroencephalogram signal. Second, to learn the transition rules between different sleep epochs, a bi-directional long short-term memory is used to capture the temporal dependence between the encoded electroencephalogram signals. Finally, to fully explore the temporal correlation between the features from the electroencephalogram signals, a self-attention routing-based capsule network is utilized to recode the importance based on the intrinsic temporal similarity of electroencephalogram signals. We evaluated our model by two different single-channel electroencephalogram signals (i.e., Fpz-Cz and Pz-Oz electroencephalogram channels) from two public sleep datasets, named Sleep-EDF-39 and Sleep-EDF-153. Our overall accuracies on the Sleep-EDF-39 and Sleep-EDF-153 datasets are 85.8% and 83.4%, with a kappa of 0.8 and 0.77, respectively. The results show that our proposed method achieves the state-of-the-art level of sleep staging using a single-channel electroencephalogram and offers the possibility of widespread application of capsule networks for sleep staging.

1. Introduction

Sleep occupies a third of a person's life, and the quality of sleep affects one's mental and physical health [1,2]. Therefore, it is necessary to evaluate sleep quality. Some sleep specialists perform sleep staging using analog electrical signals recorded in a polysomnogram (PSG) [3], which is recorded by sensors attached to the epidermis of the brain. The PSG consists of an electroencephalogram (EEG), an electrooculogram (EOG), an electromyogram (EMG), and an electrocardiogram (ECG). Regarding the subject's comfort during sleep monitoring, single-channel EEG signals are increasingly preferred for sleep staging. The PSG is divided into 30-s epochs, the sleep expert manually classifies each of these epochs into distinct stages based on decision rules developed by Rechtschaffen and Kales (R&K) [4] and the American Academy of Sleep Medicine (AASM) [5]. In clinical practice, manual sleep staging is very labor-intensive and costly. Therefore, it is crucial to conduct studies on automatic sleep staging.

Numerous studies have been conducted using various automated sleep staging methods, which can be divided into two categories: traditional machine learning-based methods [6–9], and deep learning-based methods [10–31]. Traditional machine learning-based methods typically involve manually extracted features and sleep stage classification. EEG signals are typically extracted using manual feature extraction to extract temporal or frequency domain features. Then conventional machine learning algorithms such as support vector machines (SVM) [6], random forests (RF) [7], decision tree [9] and hidden Markov model (HMM) [8] will be used to train the model for sleep stage classification based on the manually crafted features. Although these methods achieve acceptable performance, they require manual engineering and prior knowledge of expertise in the relevant field.

Recently, deep learning has attracted a great deal of interest in numerous research fields such as computer vision [32], speech recognition and synthesis [33], object detection [34], life science [35], and

* Corresponding author at: Department of Biomedical Engineering, Hefei University of Technology, Hefei 230009, China.
E-mail addresses: jinchen@mail.hfut.edu.cn (J. Chen), hanzh@hfut.edu.cn (Z. Han), qiao_heyuan@hfut.edu.cn (H. Qiao), changli@hfut.edu.cn (C. Li), hpeng@hfut.edu.cn (H. Peng).

https://doi.org/10.1016/j.bspc.2023.105351
Received 18 July 2022; Received in revised form 20 July 2023; Accepted 1 August 2023
Available online 16 August 2023

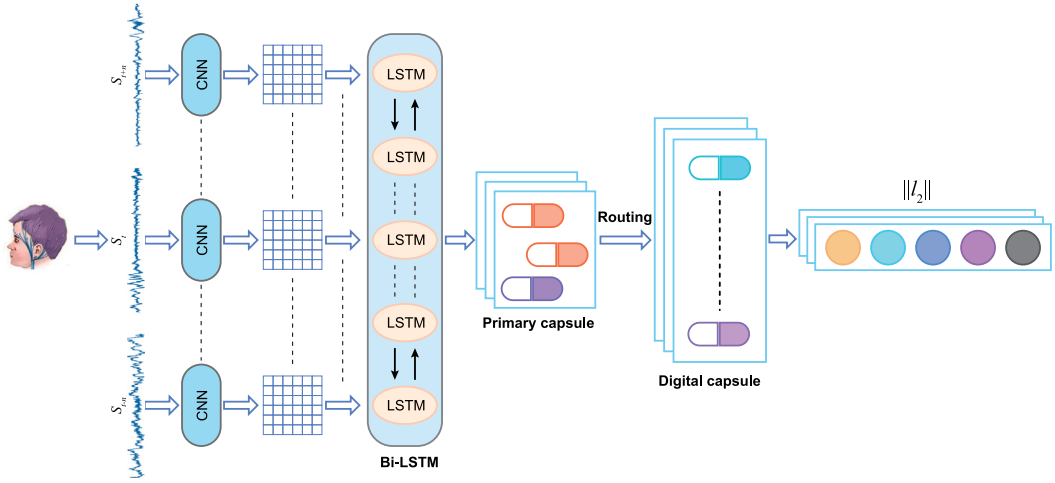

Fig. 1. Overall structure of the proposed SleepCapsNet model for automatic sleep stage classification.

natural language processing [36]. Compared to conventional machine learning models, deep learning-based methods can extract features from scratch [37] and show stellar performance. Numerous studies have proposed various deep learning-based methods for implementing the task of sleep staging. These deep learning-based methods can be divided into two categories according to the composition of the network: convolutional neural networks (CNNs), and recurrent neural networks (RNNs). CNNs are most commonly used to extract sleep-related waveform features. Some researchers have successfully applied CNNs to sleep staging tasks [10–12,15,30,38]. For instance, Tsinalis et al. [10] used one-dimensional convolution to extract features and maximum pooling layer to remove redundant information. Sors et al. [11] proposed a 14-layer deep convolutional neural network (CNN) for sleep staging. Perslev et al. [15] proposed an encoder–decoder structure CNN for sleep staging. Fiorillo et al. [12] proposed a multi-scale CNN for extracting features of different scale to perform sleep staging. Although these approaches using CNNs initially achieved reasonable performance, they lacked the modeling of temporal dependencies. There is a certain temporal correlation between EEG signals, which can be learned by using RNNs instead of using CNNs. For instance, Michielli et al. [39] proposed a cascaded recurrent neural network (RNN) architecture to perform sleep staging. But they still needed manually extracted features as initial input. Some methods combined CNN with RNN by using CNN to extract features and RNN to learn temporal dependencies [16,17,29,40]. For instance, Supratak et al. [16] proposed a CNN and RNN based model by using the CNN to obtain the 30-s epoch feature representation and the long short-term memory (LSTM) to learn the transition rules between epochs. Out of an initial exploration of sleep staging, their network is costly and requires a two-step training process. They have also developed TinySleepNet [29], a faster and more effective variant. Several methods combined CNN with attention mechanisms by using CNN to extract features and attention mechanisms to learn the temporal dependence between features [23,25–27]. For instance, Eldele et al. [23] combined CNN with channel attention and used an attention mechanism-based module for quantifying the interdependence within features. However, their method is based on short sequences and cannot learn transition rules for sequences of lengths more than 3. Qu et al. [26] proposed a residual model by using CNN to extract features and attention mechanisms to study the transition rules between epochs. But the length of the sequence they need to reference is very long. In addition, Mousavi et al. [20] proposed an encoder–decoder structure that combined RNN with attention mechanisms to learn the most relevant parts of the input sequence. Several methods used different views of the EEG signal for sleep staging [21,22,28]. For instance, Phan et al. [28] proposed a method that used both the original signal and the time–frequency images to perform sleep staging from two distinct perspectives. To solve the overfitting issue caused by the different convergence rates of different viewpoints during training, they employed different learning rates for different network branches and achieved satisfactory performance.

The input length of the signal affects the learning of transition rules between epochs by deep learning-based methods, which is important for the sleep staging task. In terms of signal input length, sleep staging that utilizes separate epoch signals is unable to detect the reference information between epochs, thereby diminishing the accuracy. Some studies used a 90-s input signal to determine the intermediate sleep stages [12,23,24,30], and others used a sequence-to-sequence model. These two distinct input methods result in distinct network structures and different costs. Due to the overlap of training signals, the use of 90-s signals as inputs can increase the time for training the model. The sequence-to-sequence model requires a long sequence to learn the time dependence. Current methods for learning the transition rules between sleep epochs by referencing the temporal context rely primarily on recurrent neural networks and attention mechanisms. The use of attention mechanisms enables the capture of longer time dependence, however, it loses information about the position of the elements in the sequence. RNNs can capture temporal dependencies between feature channels, however, the intrinsic temporal relationships between features could not be fully explored.

In this paper, we propose a self-attention routing-based capsule network named SleepCapsNet with bi-directional long short-term memory (Bi-LSTM) model to perform sleep staging work, which can better explore the intrinsic temporal correlations between EEG signals. EEG signal contains characteristic waveforms associated with sleep stages, thus, the convolutional layer of SleepCapsNet can explore the salient features of EEG signals. The intrinsic time relationship between the epochs of each sequence affects the sleep category of each epoch, so the epochs contain temporal relationships with each other. Thus, the Bi-LSTM network of SleepCapsNet can capture this temporal dependence between different epochs. Due to the inherent recursiveness of Bi-LSTM networks, there is a time-based hierarchy of EEG signals encoded as high-level features. Capsule networks can act as a classifier while capturing temporal or spatial dependencies. Therefore, using capsule networks can better extract the intrinsic temporal relationships of two-dimensional features encoded by EEG signals. Additionally, to solve the problem of difficult training due to the routing mechanism of the capsule network, we employ a new attention-based routing mechanism that drastically reduces the number of network parameters.

The following is a general summary of the contributions of this paper:

Table 1
Methods and their advantages and disadvantages.

Methods	Architecture	Input sequence length	Sequence encoder	Advantages	Disadvantages
Tsinalis [10]	CNN	1	–	No need to extract features manually.	It cannot learn about time-dependent relationships.
Sors [11]	CNN	4	–	No need to extract features manually.	It cannot learn about time-dependent relationships.
Fiorillo [12]	Multi-scale CNN	3	–	It can extract features at different scales.	Additional data enhancement techniques are needed to improve performance
Perslev [15]	Encoder–decoder-CNN	35	CNN	It can extract more deeper features.	It cannot learn about time-dependent relationships.
Zhou30[30]	Inception-CNN	3	–	It has a wider and flexible structure.	Parallelized structure adds higher cost.
Michielli [39]	LSTM	1	–	It can learn internal time dependencies.	Manual feature extraction is required first.
Supratak [16]	CNN, LSTM	25	Bi-LSTM	It can learn the transition rules between epochs.	It requires two steps of training and is difficult to train.
Supratak [29]	CNN, LSTM	20	LSTM	It has a more efficient structure.	It needs to apply data augmentation in the training.
Eldete [23]	CNN, Attention	3	Attention	Parallelizability while learning time dependency.	It cannot learn temporal dependencies of sequences of length greater than 3 and requires parametrization of the loss function.
Qu [26]	CNN, Attention	30	Attention	It can learn the time dependence between long sequences.	It requires reference to a long temporal context and does not allow for adequate learning of temporal dependencies.
Mousavi [20]	CNN, LSTM, Attention	20	Bi-LSTM	It can notice the relevant parts of the sequence.	It cannot fully learn the time dependence relationship.

Table 2
Sample distribution of sleep stage categories for each data set and number of subjects.

Dataset	N_s	W	N1	N2	N3	REM	Total
Sleep-EDF-39	20	10197	2804	17799	5703	7717	44220
Sleep-EDF-153	78	69824	21522	69132	13039	25835	199352

- We propose a self-attention routing-based capsule network with bi-directional long short-term memory model for EEG-based sleep staging. First, an efficient convolutional neural network is used to extract salient features from the EEG signal. Second, the Bi-LSTM can capture the intrinsic time dependence between epochs and thus learn the transition rules between epochs. Third, the innovative structure of the capsule networks (CapsNet) based on self-attention routing can enhance the learning of time-dependent relationships among features extracted from EEG signals and act as a classifier.
- Our method is compared with the state-of-the-art methods in the Sleep-EDFx dataset. The Fpz-Cz channel has an accuracy of 85.8% and a kappa of 0.80 on the Sleep-EDF-39 dataset. The Pz-Oz channel has an accuracy of 82.9% and a kappa of 0.76 on the Sleep-EDF-39 dataset. The Fpz-Cz channel has an accuracy of 83.4% and a kappa of 0.77 on the Sleep-EDF-153 dataset. The Pz-Oz channel has an accuracy of 79.1% and a kappa of 0.70 on the Sleep-EDF-153 dataset. The results show that SleepCapsNet achieves better or at least comparable performance to the state-of-the-art methods and demonstrate the effectiveness of our method (see Tables 1 and 2).

2. Propose methods

2.1. Overview of SleepCapsNet

Fig. 1 depicts the overall structure of our SleepCapsNet model which has four primary components, (1) feature extraction; (2) temporal context encoder; (3) capsule network; (4) classification.

First, the subject's original EEG signal is divided into 30-s epochs, and then each 30-s epoch's signal is extracted by CNN for feature extraction. The extracted features are fed into the temporal context encoder to learn the temporal relationship between different epochs, and the output of the features by the temporal context encoder contains the features of each corresponding epoch and the corresponding temporal dependencies. The feature output from Bi-LSTM is fed into the self-attention routing-based Capsule Network to enhance the learning of temporal dependent relationships. Finally, in the classification stage, the capsule network outputs capsules containing high-dimensional features during the classification stage, and the l2 norm of the vector is used to represent the probability that each class exists.

2.2. Feature extraction

Various sleep stages exhibit distinct waveform differences. For instance, the low-amplitude mixed-frequency (LAMF) wave occurs during Stage 1 (N1), and the K-complex predominately occurs during Stage 2 (N2) of NREM sleep, along with sleep spindles, which are the two distinguishing characteristics of this stage, and sawtooth waves are the only typical EEG pattern of REM sleep [41]. Fig. 2 depicts the structure of the module used for feature extraction. Inspired by [29], we design a four-layer 1D convolutional neural network. The size of the convolutional kernel in the first layer is equal to half of the sampling rate (Fs), the filter numbers are 64, and the step size is $[\frac{Fs}{16}]$. The first convolutional layer is followed by the maximum pooling layer with step size and pooling kernel of 8 to capture essential features. Afterward, we apply three layers of convolution with the same parameters, including

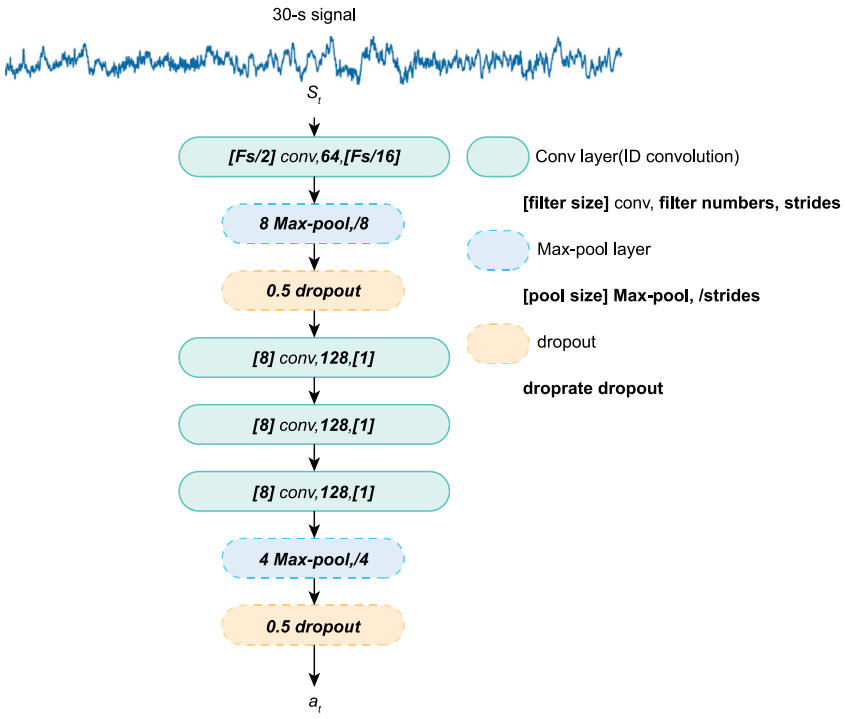

Fig. 2. Convolutional neural networks for feature extraction. Stacked convolutions can merge the field of view and make the network deeper to facilitate the extraction of deeper features.

a convolutional kernel size of 8 and filter numbers of 128. Each convolutional layer is followed by a batch normalization layer [42] and a Relu activation [43] function to accelerate the convergence of the model. Stacked convolutional layers facilitate the merging of the field of view, thereby capturing larger-scale features. Finally, we use the maximum pooling layer with step size and pooling kernel of 4 to capture essential features. We refer to previous studies when choosing network parameters [16,29]. Formally, it extracts the tth characteristics from the tth EEG epoch S_t as follows:

$$a_t = \mathrm{CNN}_{\theta_{shared}}(S_t), \tag{1}$$

where $\mathrm{CNN}_{\theta_{shared}}$ represents the CNNs that transforms from a single-EEG epoch signal into a feature vector, θ_{shared} means the trainable parameters of the CNNs, S_t denotes the tth-epoch input signal.

2.3. Temporal context encoder

There is a certain temporal dependency between sleep epochs in a sequence. To better learn the transition rules between epochs, we use a bi-directional long short-term memory(Bi-LSTM) to learn the temporal dependence between epochs.

LSTM networks are designed to learn the long-term dependencies of time series. The LSTM network consists of gates with different functions and different operations. The LSTM unit and its internal structure are presented in Fig. 3. The forget gate determines what information may be discarded or retained, with the following expression:

$$f_t = \sigma(W_f \odot [h_{t-1}, x_t] + b_f). \tag{2}$$

The purpose of updating the input gate and output gate is to determine whether to replace the memory unit with the candidate value and generate the activated portion of the current time step. It can also be stated as:

$$i_t = \sigma(W_i \odot [h_{t-1}, x_t] + b_i), \tag{3}$$

$$o_t = \sigma(W_o \odot [h_{t-1}, x_t] + b_o). \tag{4}$$

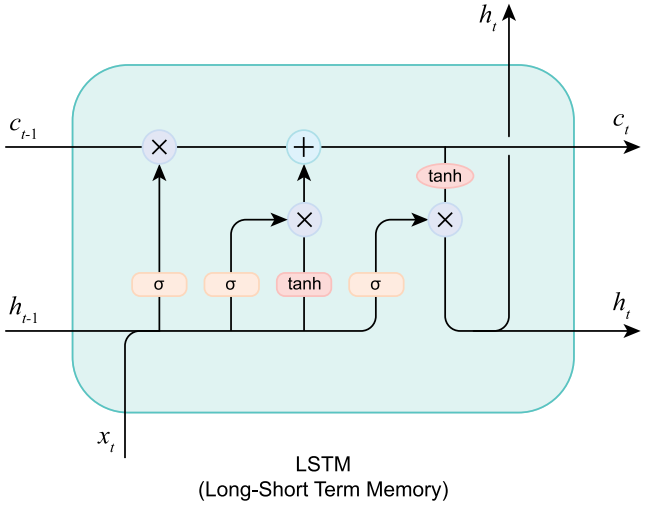

Fig. 3. The cell structure of LSTM.

The following equations govern the behavior of the LSTM unit:

$$\hat{C}_t = tanh(W_C \odot [h_{t-1}, x_t] + b_C), \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t, \tag{6}$$

$$h_t = o_t * tanh(C_t), \tag{7}$$

where σ represents the sigmoid activation function, $tanh$ represents the tangent activation function. W_f, W_i, W_o denote the weight matrices, x_t is the input vector, h_{t-1} denotes the past hidden state. b_n is bias.

A bidirectional LSTM, consisting of a forward LSTM for learning the preceding temporal dependencies and a backward LSTM for learning the subsequent temporal dependencies, performs better than a unidirectional LSTM.

$$out_t = Concat(\overrightarrow{h_t}, \overleftarrow{h_t}), \tag{8}$$

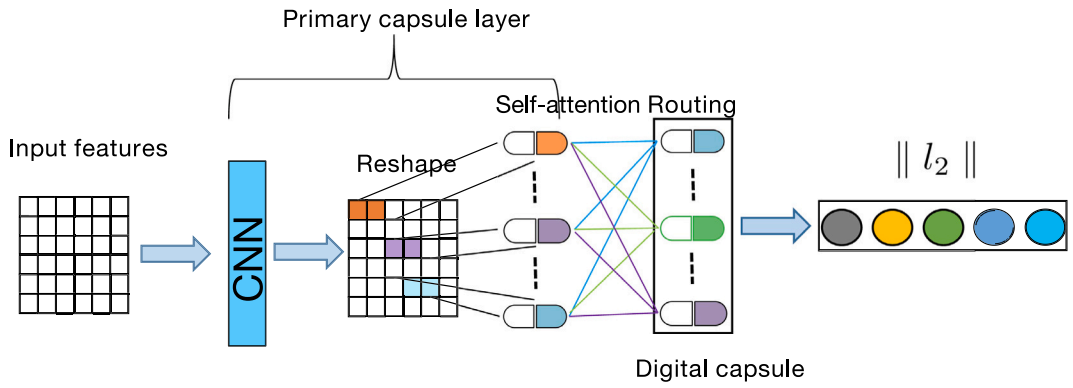

Fig. 4. Overview of the proposed capsule network.

where $\overrightarrow{h_t}$ denotes the current state in the forward direction, and $\overleftarrow{h_t}$ denotes the current state in the backward direction.

2.4. Capsule network

To further enhance the learning of temporal dependent relationships, we use a capsule network to learn the local and global relationships while the capsule network can also act as a classifier.

Although the combination of convolutional layers with pooling layers has yielded positive results in many fields, the extensive use of pooling layers can lead to the omission of crucial details, thereby reducing the network's generation. Moreover, convolutional neural networks are incapable of learning the spatial or temporal location relationships between features. In 2017, Sabour et al. [44] proposed a capsule network as an alternative to convolutional neural networks. Features are transformed into capsules that have size and direction in the capsule network. Lower level capsules can be passed to higher-level capsules through a dynamic routing mechanism. With this mechanism in place, location information is retained and transmitted.

Fig. 4 depicts the components of the proposed capsule network. In the primary capsule layer, the features first pass through a convolutional layer. Then, features are transformed into capsules, each of the primary capsules is equivalent to a vector with direction and size. Through a routing mechanism, lower-level capsules are passed to higher-level capsules in the digital capsule layer. Each capsule represents a local feature, and the capsule network can quantify the importance of each local feature to determine the final output, which cannot be learned by traditional convolutional neural networks.

2.4.1. Primary capsule

We design a primary capsule layer composed of 1D convolutional layers with 256 convolutional kernels and a convolutional kernel size of 1. The use of a 1D convolutional layer of size 1 is also more effective for merging the features from the output of the temporal context encoder, after which the features are transformed into multiple primary capsules. The reason for choosing a convolution kernel size of 1 is to increase the nonlinearity by combining the activation function behind it while being able to guarantee that the resolution of features does not change. The shape of each primary capsule, d_l, is set to 8. The choice of d_l affects the number of primary capsules and the representation of features. After our tests, the best results are obtained when d_l is chosen at 8. Each capsule contains not only the size information but also the location information for the feature.

2.4.2. Squashing activation

The nonlinear "squashing" activation function guarantees that the size of the capsule vector v_j is between 0 and 1. The following expression describes the squashing activation function:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} * \frac{s_j}{\|s_j\|}, \tag{9}$$

where s_j denotes capsule vector before using the activation function, $\|s_j\|$ represents the l2 norm of the capsule vector s_j.

2.4.3. Self-attention routing

Since the inception of capsule networks, many routing mechanisms have been proposed [44,45], however, they are either difficult to train or fail to take into account the similarity between capsules. Especially on the sleep staging task, the high sampling rate and the time dependence between epochs cause the sleep staging task to be more difficult to train compared to other EEG signal classification tasks. Thus, we use the efficient self-attention routing mechanism proposed in [46] to conclude the process of transforming primary capsules into higher-level capsules.

Fig. 5 shows the overall architecture of self-attention routing. Each capsule $U^i_{d_{l+1}}$ in layer $l+1$ is determined by all capsules from the layer l. Since the layer of $l+1$ has n_{l+1} capsules, we consider that each capsule in layer of $l+1$ corresponds to a space. We expect to learn n_{l+1} sets of matrices W that can map the capsules of the lth layer to n_{l+1} spaces from different perspectives, while keeping the same dimensionality as the output capsules:

$$Space^i_{n_l} = U_{n_l,d_l} \times W^i_{n_l,d_l,d_{l+1}}, \tag{10}$$

where U_{n_l,d_l} represents all capsules in the layer of l, there are n_l capsules in the layer of l, and the dimension of each capsule is d_l. $Space^i_{n_l}$ represents the subspace transformed by the matrix $W^i_{n_l,d_l,d_{l+1}}$. There are n_l capsules in the subspace. There are n_{l+1} capsules in the layer of $l+1$, and the dimension of each capsule is d_{l+1}. $W^i_{n_l,d_l,d_{l+1}}$ represents the ith learnable matrix.

Each space $Space^i_{l+1}$ corresponds to the ith output capsule, so there are n_{l+1} spaces and n_{l+1} matrices. Each subspace contains attributes with the category represented by the corresponding high-level capsule, and these local attributes together constitute this subspace. Referring to the self-attention mechanism proposed in [47], we reassign importance based on the similarity of the capsules in the space as follows:

$$A^i_{(n_l,n_l)} = \frac{Space^i_{n_l} \times Space^{Ti}_{n_l}}{\sqrt{d_l}}, \tag{11}$$

where $A^i_{(n_l,n_l)}$ are similarity of the capsules in the space.

To obtain the coupling coefficient for each capsule in each subspace according to the calculated similarity:

$$C^i_{(n_l,j)} = \frac{\exp A^i_{(n_l,j)}}{\sum_{j=1}^{n_l} \exp A^i_{(n_l,j)}} + B^i_{(n_l,j)}, \tag{12}$$

where $B^i_{(n_l,j)}$ denotes learnable bias.

Based on the coupling coefficient $C^i_{(n_l,n_l)}$, we can obtain the weighted sum of all capsules in the subspace.

$$U^i_{d_{l+1}} = Sum(C^i_{(n_l,n_l)} \times Space^i_{n_l}). \tag{13}$$

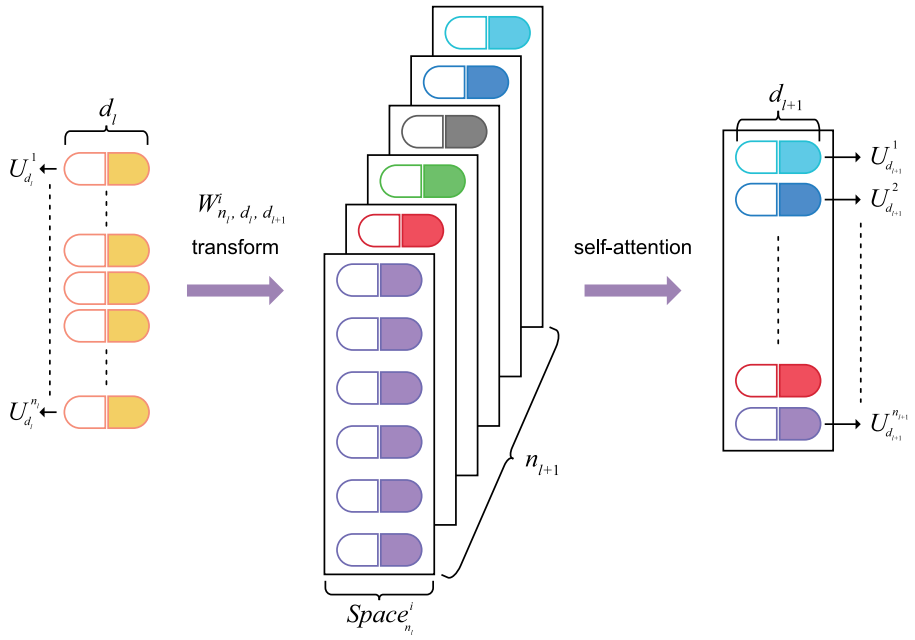

Fig. 5. The capsules in layer l are viewed through different perspectives, and then the importance of the capsules is assigned according to the intrinsic self-attentive mechanism. The ith matrix $W_{n_l,d_l,d_{l+1}}^i$ maps the capsule from the lth layer to the ith subspace $Space_{n_l}^i$. Each subspace represents the feature space with the category represented by the corresponding output capsule.

Each subspace reassigns weights based on the similarity of the capsules in the subspace. And based on these weights and the capsules in the subspace, the subspace completes the routing from the low-level capsules to the higher-level capsules. From the self-attentive mechanism of routing, we can see that the output capsule of each subspace reflects the commonality of all capsules in the subspace.

2.5. Classification

Through the capsule network, we obtain 5 capsules. Each capsule represents the characteristics of the corresponding sleep stage ($W, N1, N2, N3, R$), and by taking l2 norm for each capsule, we can determine the probability that each sleep stage exists.

2.5.1. MFE loss

Since our sleep staging task is an unbalanced distribution classification task, training the network with a balanced distribution loss function will result in errors. The use of a balanced distribution loss function will increase the probability that the network will identify the signal as the largest number of categories in the training set. To solve this issue, we employ the mean false error loss function described below, which does not require parameter tuning:

$$l_{MFE} = \frac{1}{M}\sum_{k=1}^{K}\frac{1}{M_k}\sum_{i=1}^{M}(y_i^k - \hat{y}_i^k)^2, \tag{14}$$

where y_i^k is the actual label for ith sample and $\hat{y}_i^k$ is the predicted probability of ith sample for the class k, M is the total number of samples and K is the number of classes. M_k is the number of samples in class k. The MFE loss function compels the network to prioritize categories with a small sample size, thereby mitigating the issue of unbalanced distribution due to the sample size. In addition, since we cannot precisely determine the distribution of the validation set, the loss function has a priori and there is no need to adjust the hyperparameters.

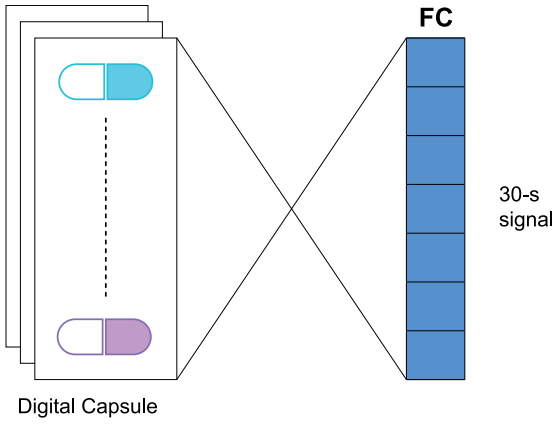

Fig. 6. Reconstruction regularizer.

2.5.2. Reconstruction regularizer and total loss

As depicted in Fig. 6, we place a fully connected layer at the end of the network to reconstruct the original input signal to prevent the network from becoming overfitted. The Euclidean distance between the reconstructed signal and the original signal is used to calculate the reconstruction loss. We multiply the regularized loss by a factor γ and add it to the loss function. Therefore, the total loss:

$$L_{total} = l_{MFE} + \gamma \times loss_{Recon}. \tag{15}$$

3. Experimental results

3.1. Datasets

The Sleep-EDFx dataset [48] was utilized to evaluate our model. In Sleep-EDFx, we used 153 PSG recordings from the study of age effects in healthy subjects (SC) which is collected from 78 participants.

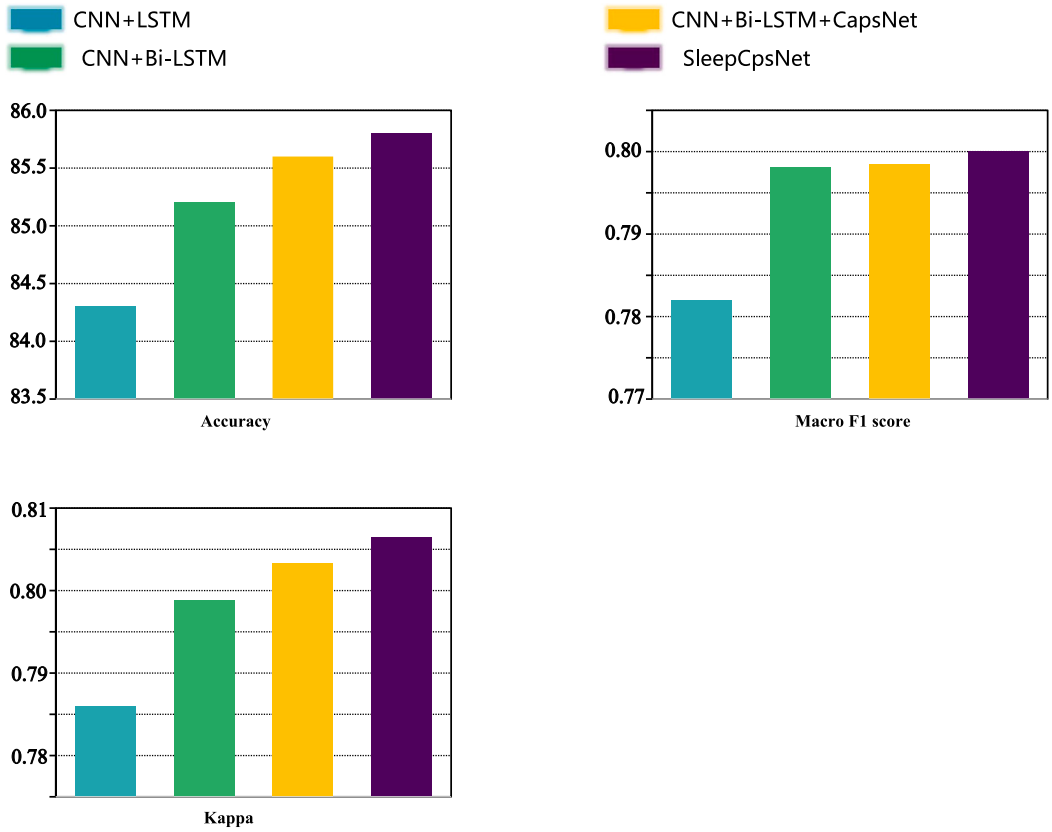

Fig. 7. Ablation study conducted on Sleep-EDF-39 dataset.

To compare with existing methods, we divided Sleep-EDFx into two datasets, Sleep-EDF-39 and Sleep-EDF-153. Sleep-EDF-39 contains the first 20 subjects of SC, excluding the 13th subject who has only one night of data, and the other 19 persons who have two nights of data. There are 78 subjects in Sleep-EDF-153. To demonstrate the wide applicability of our model to different channels, we conduct experiments on the Fpz-Cz channel, the Pz-Oz channel, and the Eog channel, respectively. We chose these specific channels because the Sleep-EDFx dataset only records the EEG signals of these specific channels.

3.2. Experimental settings

The k-fold cross-validation scheme was utilized to evaluate our model. To ensure a fairer comparison of the experimental results, the k values in this paper were set to be the same as those studies [16,20,23,29,31]. For Sleep-EDF-39, k equals 20, which means that we utilize data from 19 subjects for the training set and data from 1 subject for the test set. For Sleep-EDF-153, k equals 10. The remaining 10 percent of the training set was used for validation in each fold. The number of mini-batches used for training was 20, and the length of each sequence was 15. For updating the model's weights, the Adam optimizer's parameters lr, β_1, and β_2 were set to 10^{-4}, 0.90 and 0.999, respectively. For regularization, the L2 weight decay was applied in the loss function as a regularization term whose weight was 10^{-3}. The reconstruction loss in the capsule network was also applied in the loss function whose weight was set to 10^{-4}. The drop rate of the dropout layer [49] in the CNN part was set to 0.5. The maximum number of training epochs for the network was set to 200. If the validation set loss did not decrease within 20 iterations, the network would cease training and save the weight with the lowest validation loss as the optimized weight.

Tensorflow [50] was utilized as the backend for our experiments, and an NVIDIA GeForce GTX 1070 graphics card with 8 GB of video memory was used to accelerate the training.

3.3. Evaluation metrics

In the work, we evaluated the performance of our model using per-class precision (PR), per-class recall (RE), per-class F1-score (F1), macro-averaging F1-score (MF1), and overall accuracy (ACC), and Cohen's Kappa coefficient (k). PR reflects the proportion of true positive samples among the positive cases determined by the classifier. RE reflects the proportion of all samples that are truly positive cases that are determined to be positive by the classifier. F1 is a combined evaluation based on PR and RE. When F1 is high, it can indicate that the test method is more effective. MF1 is the average of F1 of all classes, which reflects the overall performance of the model. Cohen's Kappa coefficient (k) is used to measure the agreement of two different sleep staging results, which corresponds to the results of our model and the labels developed by the sleep experts. The formula for the above evaluation indices is as follows:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{16}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{17}$$

$$F1 = \frac{2 \times Recall \times Precision}{Precision + Recall}, \tag{18}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

$$Kappa = \frac{Accuracy - P_e}{1 - P_e}, \tag{20}$$

$$MF1 = \frac{1}{K} \sum_{i=1}^{K} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}, \tag{21}$$

where TP represents the true positive, TN denotes the true negative, FP means the false positive, FN is the false negative and P_e is the hypothetical probability of chance agreement. i represents the class i.

Table 3
The confusion matrix of the proposed model applied on Fpz-Cz channel from the Sleep-EDF-39 dataset.

	W	N1	Prediction		R	Per-class metrics		
			N2	N3		Precision	Recall	F1
W	9442	435	173	20	127	91.8	92.6	92.2
N1	432	1309	684	13	366	47.6	46.7	47.1
N2	268	540	15852	541	598	87.3	89.1	88.2
N3	42	9	649	5002	1	89.7	87.7	88.7
R	101	457	804	0	6355	85.3	82.4	83.8

Table 4
The confusion matrix of the proposed model applied on Pz-Oz channel from the Sleep-EDF-39 dataset.

	W	N1	Prediction		R	Per-class metrics		
			N2	N3		Precision	Recall	F1
W	9191	407	140	16	443	88.4	90.1	89.3
N1	656	1033	665	26	424	48.6	36.8	41.9
N2	203	372	15646	832	746	85.1	87.9	86.5
N3	17	9	1158	4517	2	83.7	79.2	81.4
R	326	306	784	5	6296	79.6	81.6	80.6

Table 5
The confusion matrix of the proposed model applied on EOG channel from the Sleep-EDF-39 dataset.

	W	N1	Prediction		R	Per-class metrics		
			N2	N3		Precision	Recall	F1
W	8692	409	573	80	443	84.2	85.2	84.7
N1	626	855	953	10	360	45.7	30.5	36.6
N2	558	475	14999	965	802	80.6	84.3	82.4
N3	80	37	1177	4403	6	80.7	77.2	78.9
R	361	95	910	0	6351	79.8	82.3	81.0

Table 6
The confusion matrix of the proposed model applied on Fpz-Cz channel from the Sleep-EDF-153 dataset.

	W	N1	Prediction		R	Per-class metrics		
			N2	N3		Precision	Recall	F1
W	65312	3118	566	44	762	92.2	93.6	92.9
N1	3904	8724	6589	113	2199	56.2	43.2	48.8
N2	752	3031	60682	1958	2724	83.3	87.8	85.5
N3	38	5	2631	10321	44	82.1	79.2	80.6
R	798	1423	2349	129	21136	78.7	81.8	80.2

3.4. Classification performance

Tables 3, 4, 5 and 6 presents the confusion matrices of the proposed model applied on the Fpz-Cz channel, the Pz-Oz channel, and the Eog channel, respectively. The diagonal elements of the confusion matrix represent the agreement with the labels classified by the experts, while the right side of the table represents the evaluation parameters for each category. We used PR, RE, and F1 to measure the performance of the model in each category separately. The higher evaluation parameters indicate better performance. In terms of the channels used, the Fpz-Cz channel has the best performance, whereas the Eog channel has the worst performance. For each category, the best performance is in the W and N2 phases, while the worst performance is in the N1 phase, which is easily confused with other sleep phases. This is primarily due to the small amount of data in the N1 phase and the similarity of the N1 phase to other sleep phases, which makes the N1 phase more difficult to distinguish from other sleep phases.

3.5. Ablation study

To represent the function of each component of the model, we conducted ablation experiments. Here, we will compare the model's performance with and without the part-capsule network. Our model for comparison consists of four base models as follows:

- **CNN + LSTM**, containing a CNN and a unidirectional LSTM.
- **CNN + Bi-LSTM**, containing a CNN and a bi-directional LSTM.
- **CNN + Bi-LSTM + CapsNet**, containing a CNN and a bi-directional LSTM and capsule network without reconstruction loss.
- **SleepCapsNet**, the proposed model, containing a CNN and a bi-directional LSTM and capsule network with reconstruction loss.

We evaluate the performance of different structures using the macro-averaging F1-score (MF1), the overall accuracy (ACC), and Cohen's Kappa coefficient. These evaluation metrics can reflect the performance of the model in general. Fig. 7 depicts the performance histograms of various models with varying evaluation parameters. The bidirectional LSTM performs better than the unidirectional LSTM model, primarily because the unidirectional LSTM model can only capture the temporal dependencies in one direction, whereas the bidirectional LSTM model captures temporal dependencies in both directions, which is advantageous for learning the transition rules in the sleep stage.

Table 7
Comparison among SleepCapsNet and state-of-the-art methods. The best values on each dataset are highlighted in bold.

Methods	Dataset	Channel	CV	Test Epochs	Overall performance		Per-class F1 score				
					Accuracy	Kappa(%)	W	N1	N2	N3	R
DeepSleepNet	Sleep-EDF-39	Fpz-Cz	20-fold CV	41950	82.0	76.0	84.7	46.6	85.9	84.8	82.4
SleepEEGNet				42308	84.26	79.0	89.19	**52.19**	86.77	85.13	**85.02**
TinySleepNet				44220	85.4	80.0	90.1	51.4	88.5	88.3	84.3
CCRRSleepNet				41950	84.29	78.0	89.01	51.73	87.25	88.2	82.86
AttnNet				42308	85.6	80.0	90.3	47.9	**89.8**	**89.0**	85.0
Our Model				44220	**85.84**	**80.64**	**92.2**	47.1	88.2	88.7	83.8
DeepSleepNet	Sleep-EDF-39	Pz-Oz	20-fold CV	41950	79.8	72	88.1	37	82.7	77.3	80.3
CCRRSleepNet				41950	80.31	73	86.01	41.54	84.87	80.97	79.56
SleepEEGNet				42308	82.83	77	**90.27**	**44.64**	85.74	**81.55**	**82.88**
Our Model				44220	**82.9**	76.6	89.3	41.9	**86.5**	81.4	80.6
EOGNet	Sleep-EDF-39	EOG	20-fold CV	41950	76.3	69.3	81.6	34.9	81.6	72.5	75.7
Our Model				44220	**79.8**	**72.2**	**84.7**	**36.6**	**82.4**	**78.9**	**81.0**
SleepEEGNet	Sleep-EDF-153	Fpz-Cz	10-fold Cv	222479	80.03	73.0	91.72	44.05	82.49	73.45	76.06
TinySleepNet				199352	83.1	77.0	92.8	51.0	85.3	81.1	80.3
AttnNet				195479	82.9	77.0	92.6	47.4	85.5	83.7	81.5
Our Model				199352	**83.4**	77.1	**92.9**	48.8	**85.5**	80.6	80.2
SleepEEGNet		Pz-Oz		222479	77.56	68.94	90.26	42.21	79.71	74.83	72.19
Our Model				199352	**79.1**	70.5	90.1	**43.3**	**81.3**	68.8	**72.1**

In contrast to the CNN + Bi-LSTM model, the CNN + Bi-LSTM + CapsNet model performs better and is higher in all three evaluation indices, indicating that the addition of a capsule network can improve the Bi-LSTM network's performance. We also compare the effect of reconstruction loss on network performance. The reconstruction loss can improve network performance, thereby reducing overfitting. The optimal performance is achieved by minimizing reconstruction regularization loss.

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

Comparing our model to the state-of-art approach, we evaluate parameters such as the macro-averaging F1-score (MF1), the overall accuracy (ACC), and the Cohen's Kappa coefficient.

Table 7 displays the results obtained using modern techniques. The models we compare include DeepSleepNet [16], SleepEEGNet [20], TinySleepNet [29], CCRRSleepNet [31], AttnNet [23], EOGNet [51]. Our model outperforms other models in terms of accuracy, achieving 85.84% on Sleep-EDF-39 and 83.4% on Sleep-EDF-153. Specifically, our model outperforms other models in terms of W-phase recognition accuracy and generalization performance on large data sets. Using EOG channel signals for staging, our model achieves superior results compared to EOGNet, outperforming EOGNet in every way. Our model is a Seq-to-Seq model that uses multiple consecutive epochs as input and outputs corresponding to the sleep stage for each epoch. While DeepSleepNet employs a two-step training method, first learning the representation for each epoch, and then training the RNN to learn the transition relationship between epochs, our model does not require two-step learning. The first half of our model is comparable to that of TinySleepNet, except for whether the features extracted from the CNN are flattened and fed into the LSTM. Note that TinySleepNet requires mini-batch data augmentation during training. AttnNet is a network that uses multi-head attention to capture temporal dependencies. AttnNet differs from our network in that it achieves the best results by using three epochs as input and outputting the label of the middle epoch. In addition, their loss function is a class-aware loss function that lacks a reasonable explanation and requires hyperparameter tuning for different datasets, whereas our network does not.

4. Discussion

This paper proposes the SleepCapsNet hybrid network for sleep stage classification. This model can be applied to different EEG channels (Fpz-Cz and Pz-Oz) and EOG channels without modifying the model's structure or training algorithm. From the experimental results conducted on each channel, we can see that the use of the Fpz-Cz channel for sleep staging has achieved better performance compared to other channels. This suggests that the Fpz-Cz channel inherently has more sleep-related information. In terms of performance in each category, the best performance is in the W stage and the worst performance is in the N1 stage. The identification of the N1 stage has always been a difficult task for the sleep staging task, on the one hand, the data of the N1 stage is relatively limited, and on the other hand, its identification is inherently difficult and easily confused with other stages. Our proposed model can be used for sleep staging from raw unprocessed single-EEG signals, which is friendly to subjects and people lacking expertise in the sleep field. The reason why we use single-channel EEG signals for sleep staging is that it reduces the subject's measurement discomfort. However, it also increases the risk of inaccurate sleep staging. Single-channel EEG signals are not always present with characteristic sleep-related waveforms, and even contain different levels and types of noisy signals. For instance, in Sleep-EDF-39, subject 11's validation accuracy on the Fpz-Cz channel is only 60%, which is the worst result among the 20-fold cross-validation results. This is primarily due to the presence of high-intensity noisy signals in the validation set, which causes epochs belonging to other categories to be identified as W- or R-periods. Past deep learning methods cannot fully explore the temporal relationships between features. Our model can better captures such intrinsic temporal relationships by introducing a capsule network for more accurate sleep staging. The ablation experiments demonstrate that the capsule network can enhance the performance of the LSTM and that the bidirectional LSTM outperforms the unidirectional LSTM. The Bi-LSTM can take into account temporal reference information from both directions compared to LSTM, thus allowing for more accurate sleep staging. Prior to this, applying capsule networks to sleep staging tasks was very difficult. Conventional routing mechanisms require a very large number of parameters to implement the delivery of capsules, the self-attention routing mechanism we use can fully account for the inherent similarity between capsules and has fewer parameters. The efficient self-attention routing mechanism we use enables the wide application of capsule networks for sleep staging tasks based on EEG signals. In clinical practice, sleep specialists perform sleep staging tasks with reference not only to single-channel EEG signals but also to other channels including the EOG channel. Sleep-related features such as sleep spindle waves are not present in all EEG signals simultaneously, and eye movements and electromyographic signals are also important for sleep staging discrimination. In addition, scoring styles vary among specialists. There are distinctions between the scoring styles

of various experts, resulting in disparate scoring results [52]. Even so, the overall kappa of our model is over 0.8 on Sleep-EDF-39. In general, our model can achieve comparable or better performance on the Sleep-EDFx dataset than the state-of-art deep learning methods.

The model we propose has a certain practical value. In hospitals, sleep assessments are performed manually by sleep specialists, which is a time-consuming and expensive process. Using our model for sleep staging, the model can be trained on data accumulated in hospitals. Sleep staging can be done in a very short time without manual feature extraction, which facilitates the monitored person to make rapid psychological or physical adjustments based on the results. In particular, our method can reduce infant discomfort in infant sleep monitoring, which helps monitor the development of infants. By combining our model with physical EEG signal acquisition tools, individual sleep monitoring can also be made more convenient.

On the other hand, we only test on publicly available datasets. However, there are more clinical uncertainties, such as individual signal variability and noise artifacts, which can lead to actual classification errors. Putting our model into practical application is still some distance away, and this is a topic we are working on.

5. Conclusion

In this work, we propose a novel model for automated sleep staging. A convolutional neural network is used to extract salient features from the original signal, Bi-LSTM is used to learn temporal dependencies between EEG signals, and a capsule network is used to enhance the learning of temporal dependencies between encoded features. Through ablation experiments, combining the Bi-LSTM with the capsule network improves the learning of transition rules between sleep stages. The self-attention mechanism based-routing significantly reduces the number of parameters of the capsule network, allowing the capsule network to be applied effectively to the sleep staging task. Compared to other models, experimental results on Sleep-EDF demonstrate that our proposed model achieves the state-of-art level. Our method is based on unprocessed single-channel raw data, so in the future, we will focus on multi-channel signals as an input to improve sleep staging precision.

CRediT authorship contribution statement

Jin Chen: Conceptualization, Methodology, Experiments, Writing. **Zhihui Han:** Methodology, Validation, Experiments, Writing. **Heyuan Qiao:** Visualization, Writing – review & editing. **Chang Li:** Visualization, Writing – review & editing. **Hu Peng:** Writing – review & editing, Supervision.

Declaration of competing interest

None of the authors have a conflict of interest.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants 62071165, 62171177, 41901350), the Fundamental Research Funds for the Central Universities of China (Grant PA2023IISL0095).

Appendix. Abbreviations and definitions

1.PSG: polysomnogram

2.EEG: electroencephalogram

3.EOG: electrooculogram

4.EMG: electromyogram

5.ECG: electrocardiogram

6.R&K: Rechtschaffen and Kales

7.AASM: Academy of Sleep Medicine

8.SVM: support vector machines

9.RF: random forests

10.HMM: hidden Markov model

11.CNNS: convolutional neural networks

12.CNN: convolutional neural network

13.RNNs: recurrent neural networks

14.RNN: recurrent neural network

15.LSTM: long short-term memory

16.Bi-LSTM: bi-directional long short-term memory

17.CapsNet: capsule networks

18.LAMF: low-amplitude mixed-frequency

19.SleepCapsNet: our proposed model

References

[1] F.S. Luyster, P.J. Strollo, P.C. Zee, J.K. Walsh, Sleep: A health imperative, Sleep 35 (6) (2012) 727–734, http://dx.doi.org/10.5665/sleep.1846.

[2] D.J. Buysse, Sleep health: can we define it? Does it matter? Sleep 37 (1) (2014) 9–17, http://dx.doi.org/10.5665/sleep.3298.

[3] S.A. Keenan, An overview of polysomnography, Handb. Clin. Neurophysiol. 6 (2005) 33–50, http://dx.doi.org/10.1016/S1567-4231(09)70028-0.

[4] 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, http://dx.doi.org/10.1001/archpsyc.1969.01740140118016.

[5] 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.

[6] E. Alickovic, A. Subasi, Ensemble SVM method for automatic sleep stage classification, IEEE Trans. Instrum. Meas. 67 (6) (2018) 1258–1265, http://dx.doi.org/10.3390/10.1109/TIM.2018.2799059.

[7] 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, http://dx.doi.org/10.1109/TNSRE.2017.2776149.

[8] H. Ghimatgar, K. Kazemi, M.S. Helfroush, A. Aarabi, An automatic single-channel EEG-based sleep stage scoring method based on hidden Markov model, J. Neurosci. Methods 324 (2019) 108320, http://dx.doi.org/10.1016/j.jneumeth.2019.108320.

[9] S.A. Imtiaz, E. Rodriguez-Villegas, Automatic sleep staging using state machine-controlled decision trees, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (EMBC), IEEE, 2015, pp. 378–381, http://dx.doi.org/10.1109/EMBC.2015.7318378.

[10] O. Tsinalis, P.M. Matthews, Y. Guo, S. Zafeiriou, Automatic sleep stage scoring with single-channel EEG using convolutional neural networks, 2016, http://dx.doi.org/10.48550/arXiv.1610.01683, arXiv preprint arXiv:1610.01683.

[11] 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, http://dx.doi.org/10.1016/j.bspc.2017.12.001.

[12] L. Fiorillo, P. Favaro, F.D. Faraci, Deepsleepnet-lite: A simplified automatic sleep stage scoring model with uncertainty estimates, IEEE Trans. Neural Syst. Rehabil. Eng. 29 (2021) 2076–2085, http://dx.doi.org/10.48550/arXiv.2108.10600.

[13] E. Khalili, B.M. Asl, Automatic sleep stage classification using temporal convolutional neural network and new data augmentation technique from raw single-channel EEG, Comput. Methods Programs Biomed. 204 (2021) 106063, http://dx.doi.org/10.1016/j.cmpb.2021.106063.

[14] B. Yang, X. Zhu, Y. Liu, H. Liu, A single-channel EEG based automatic sleep stage classification method leveraging deep one-dimensional convolutional neural network and hidden Markov model, Biomed. Signal Process. Control 68 (2021) 102581, http://dx.doi.org/10.1016/j.bspc.2021.102581.

[15] M. Perslev, M. Jensen, S. Darkner, P.J. Jennum, C. Igel, U-time: A fully convolutional network for time series segmentation applied to sleep staging, Adv. Neural Inf. Process. Syst. 32 (2019) http://dx.doi.org/10.48550/arXiv.1910.11162.

[16] 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, http://dx.doi.org/10.1109/TNSRE.2017.2721116.

[17] H. Nie, S. Tu, L. Xu, Recsleepnet: An automatic sleep staging model based on feature reconstruction, in: 2021 IEEE International Conference on Bioinformatics and Biomedicine, (BIBM), IEEE, 2021, pp. 1458–1461, http://dx.doi.org/10.1109/BIBM52615.2021.9669687.

[18] F. Li, R. Yan, R. Mahini, L. Wei, Z. Wang, K. Mathiak, R. Liu, F. Cong, End-to-end sleep staging using convolutional neural network in raw single-channel EEG, Biomed. Signal Process. Control 63 (2021) 102203, http://dx.doi.org/10.1016/j.bspc.2020.102203.

[19] 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, http://dx.doi.org/10.1109/TBME.2018.2872652.

[20] 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, http://dx.doi.org/10.1371/journal.pone.0216456.

[21] Z. Jia, X. Cai, G. Zheng, J. Wang, Y. Lin, SleepPrintNet: A multivariate multimodal neural network based on physiological time-series for automatic sleep staging, IEEE Trans. Artif. Intell. 1 (3) (2020) 248–257, http://dx.doi.org/10.1109/TAI.2021.3060350.

[22] J. Ye, Q. Xiao, J. Wang, H. Zhang, J. Deng, Y. Lin, Cosleep: A multi-view representation learning framework for self-supervised learning of sleep stage classification, IEEE Signal Process. Lett. 29 (2021) 189–193, http://dx.doi.org/10.1109/LSP.2021.3130826.

[23] 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, http://dx.doi.org/10.1109/TNSRE.2021.3076234.

[24] 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, http://dx.doi.org/10.1109/TNSRE.2019.2896659.

[25] H. Phan, K.B. Mikkelsen, O. Chen, P. Koch, A. Mertins, M. De Vos, Sleep-Transformer: Automatic sleep staging with interpretability and uncertainty quantification, IEEE Trans. Biomed. Eng. (2022) http://dx.doi.org/10.1109/TBME.2022.3147187.

[26] W. Qu, Z. Wang, H. Hong, Z. Chi, D.D. Feng, R. Grunstein, C. Gordon, A residual based attention model for eeg based sleep staging, IEEE J. Biomed. Health Inform. 24 (10) (2020) 2833–2843, http://dx.doi.org/10.1109/JBHI.2020.2978004.

[27] G. Shi, Z. Chen, R. Zhang, A transformer-based spatial-temporal sleep staging model through raw EEG, in: 2021 International Conference on High Performance Big Data and Intelligent Systems, (HPBD&IS), IEEE, 2021, pp. 110–115, http://dx.doi.org/10.1109/HPBDIS53214.2021.9658439.

[28] H. Phan, O.Y. Chén, M.C. Tran, P. Koch, A. Mertins, M. De Vos, XSleepNet: Multi-view sequential model for automatic sleep staging, IEEE Trans. Pattern Anal. Mach. Intell. (2021) http://dx.doi.org/10.1109/TPAMI.2021.3070057.

[29] A. Supratak, Y. Guo, TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel EEG, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, (EMBC), IEEE, 2020, pp. 641–644, http://dx.doi.org/10.1109/EMBC44109.2020.9176741.

[30] D. Zhou, J. Wang, G. Hu, J. Zhang, F. Li, R. Yan, L. Kettunen, Z. Chang, Q. Xu, F. Cong, Singlechannelnet: A model for automatic sleep stage classification with raw single-channel eeg, Biomed. Signal Process. Control 75 (2022) 103592, http://dx.doi.org/10.1016/j.bspc.2022.103592.

[31] W. Neng, J. Lu, L. Xu, Ccrrsleepnet: A hybrid relational inductive biases network for automatic sleep stage classification on raw single-channel eeg, Brain Sci. 11 (4) (2021) 456, http://dx.doi.org/10.3390/brainsci11040456.

[32] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Comput. Intell. Neurosci. 2018 (2018) http://dx.doi.org/10.1155/2018/7068349.

[33] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al., Deep speech: Scaling up end-to-end speech recognition, 2014, http://dx.doi.org/10.48550/arXiv.1412.5567, arXiv preprint arXiv:1412.5567.

[34] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232, http://dx.doi.org/10.48550/arXiv.1807.05511.

[35] B. Ramsundar, P. Eastman, P. Walters, V. Pande, Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more, O'Reilly Media, 2019.

[36] D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2020) 604–624, http://dx.doi.org/10.1109/TNNLS.2020.2979670.

[37] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) 1–38, http://dx.doi.org/10.1145/3285029.

[38] L. Wei, Y. Lin, J. Wang, Y. Ma, Time-frequency convolutional neural network for automatic sleep stage classification based on single-channel EEG, in: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence, (ICTAI), 2017, pp. 88–95, http://dx.doi.org/10.1109/ICTAI.2017.00025.

[39] N. Michielli, U.R. Acharya, F. Molinari, Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals, Comput. Biol. Med. 106 (2019) 71–81, http://dx.doi.org/10.1016/j.compbiomed.2019.01.013.

[40] C. Zhao, J. Li, Y. Guo, SleepContextNet: A temporal context network for automatic sleep staging based single-channel EEG, Comput. Methods Programs Biomed. 220 (2022) 106806, http://dx.doi.org/10.1016/j.cmpb.2022.106806.

[41] M.A. Carskadon, W.C. Dement, et al., Normal human sleep: An overview, Princ. Pract. Sleep Med. 4 (1) (2005) 13–23, http://dx.doi.org/10.1016/j.mcna.2004.01.001.

[42] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456, http://dx.doi.org/10.48550/arXiv.1502.03167.

[43] A.F. Agarap, Deep learning using rectified linear units (relu), 2018, http://dx.doi.org/10.48550/arXiv.1803.08375, arXiv preprint arXiv:1803.08375.

[44] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.48550/arXiv.1710.09829.

[45] G.E. Hinton, S. Sabour, N. Frosst, Matrix capsules with EM routing, in: International Conference on Learning Representations, 2018.

[46] V. Mazzia, F. Salvetti, M. Chiaberge, Efficient-capsnet: Capsule network with self-attention routing, Sci. Rep. 11 (1) (2021) 1–13, http://dx.doi.org/10.1038/s41598-021-93977-0.

[47] 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) http://dx.doi.org/10.48550/arXiv.1706.03762.

[48] 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, http://dx.doi.org/10.1161/01.cir.101.23.e215.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016, http://dx.doi.org/10.48550/arXiv.1603.04467, arXiv preprint arXiv:1603.04467.

[51] J. Fan, C. Sun, M. Long, C. Chen, W. Chen, Eognet: A novel deep learning model for sleep stage classification based on single-channel eog signal, Front. Neurosci. 15 (2021) http://dx.doi.org/10.3389/fnins.2021.573194.

[52] H. Danker-hopfe, P. Anderer, J. Zeitlhofer, M. Boeck, H. Dorn, G. Gruber, E. Heller, E. Loretz, D. Moser, S. Parapatics, et al., Interrater reliability for sleep scoring according to the rechtschaffen & kales and the new AASM standard, J. Sleep Res. 18 (1) (2009) 74–84, http://dx.doi.org/10.1111/j.1365-2869.2008.00700.x.