SleepFC: Feature Pyramid and Cross-Scale Context Learning for Sleep Staging

Wei Li, Teng Liu, Baoguo Xu, *Member, IEEE*, and Aiguo Song, *Senior Member, IEEE*

***Abstract*— Automated sleep staging is essential to assess sleep quality and treat sleep disorders, so the issue of electroencephalography (EEG)-based sleep staging has gained extensive research interests. However, the following difficulties exist in this issue: 1) how to effectively learn the intrinsic features of salient waves from single-channel EEG signals; 2) how to learn and capture the useful information of sleep stage transition rules; 3) how to address the class imbalance problem of sleep stages. To handle these problems in sleep staging, we propose a novel method named SleepFC. This method comprises convolutional feature pyramid network (CFPN), cross-scale temporal context learning (CSTCL), and class adaptive fine-tuning loss function (CAFTLF) based classification network. CFPN learns the multi-scale features from salient waves of EEG signals. CSTCL extracts the informative multi-scale transition rules between sleep stages. CAFTLF-based classification network handles the class imbalance problem. Extensive experiments on three public benchmark datasets demonstrate the superiority of SleepFC over the state-of-the-art approaches. Particularly, SleepFC has a significant performance advantage in recognizing the N1 sleep stage, which is challenging to distinguish.**

***Index Terms*— Sleep staging, convolutional feature pyramid network, cross-scale temporal context learning, class adaptive fine-tuning loss function, electroencephalography.**

I. Introduction

SLEEP is important for humans [1]. Different sleep stages, such as non-rapid eye movement (NREM) and rapid eye movement (REM), are essential for memory consolidation, attention improvement, emotion regulation, and so forth [2], [3]. Accurately classifying the sleep stages is indispensable for comprehending how the sleep impacts human physical and mental health. However, manual sleep staging heavily relies on the knowledge and labor of sleep experts. The laboring process is empirical and time-consuming [4], [5]. By contrast, automatic sleep staging is promising to enhance the accuracy and efficiency of sleep analysis [6], [7].

Sleep staging refers to distinguishing the stages of human sleep. Sleep specialists generally categorize the sleep stages based on polysomnography (PSG), which consists of EEG, electrooculogram (EOG), electromyogram (EMG), and electrocardiogram (ECG) [8]. This paper focuses on single-channel EEG for sleep staging. Compared with PSG or multi-channel EEG, single-channel EEG holds great practical significance, because it is quite convenient and efficient to collect only one sort of signals via single one channel. Besides, technological improvement of sleep staging based on single-channel EEG is very helpful for enhancing the performance of sleep staging using multi-channel EEG as well as PSD. According to the American Academy of Sleep Medicine (AASM) criteria, PSG data can be divided into WAKE, REM, and NREM. NREM can further be classified into N1, N2 and N3 stages. In different sleep stages, the EEG signals display different waveforms, amplitudes, and spectra [9]. For instance, the salient waves of REM stage are sawtooth waves, but the salient waves of N2 stage are sleep spindles or K-complexes [10]. Capturing the characteristics of signal wave patterns can be beneficial for sleep stage classification. Moreover, sleep transition rules are also informative to distinguish sleep stages, especially those between the neighboring sleep stages, such as W−N1−N1−W−N1−N1, N2−N2−N3−N2−N3, N2−N2−REM, etc.

Many researchers have recommended deep learning for sleep staging based on EEG. Typical methods include convolutional neural network (CNN) [11], [12], convolutional recurrent neural network (CRNN) [13], [14], fully convolutional network (FCN) [15], etc. Early methodology relies on the one-to-one scheme in which an EEG epoch corresponds to one sleep stage [16]. Generally, EEG signal waves of different sleep stages display distinctive temporal and spectral characteristics. For example, K-complexes occur approximately every 1.0-1.7 minutes, but alpha rhythm undergoes periodic oscillations with a frequency range of 8 to 12 Hz. Therefore, multi-scale feature extraction plays an important role in sleep staging, because it can capture the different characteristics of salient EEG waves. Eldele et al. [17] designed two parallel CNNs, which utilize small and large filters to learn the representations from EEG saline waves for classifying sleep stages. Wang et al. [18] employed the attention mechanism and the multi-scale convolution to extract the salient wave features from EEG to classify sleep stages. Although CNN models have shown inspiring performance in sleep stage classification, their one-to-one scheme ignores the important sleep transition rules between neighboring sleep stages.

In recent years, both many-to-one and sequence-to-sequence schemes, which rely on multiple EEG epochs for sleep staging, have attracted increasing research interests [11], [12], [14].

Manuscript received 27 December 2023; revised 12 April 2024 and 15 May 2024; accepted 17 May 2024. Date of publication 28 May 2024; date of current version 19 June 2024. This work was supported by the Basic Research Project of Leading Technology of Jiangsu Province under Grant BK20192004. *(Corresponding author: Wei Li.)*

The authors are with the School of Instrument Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: li-wei@seu.edu.cn).

Digital Object Identifier 10.1109/TNSRE.2024.3406383

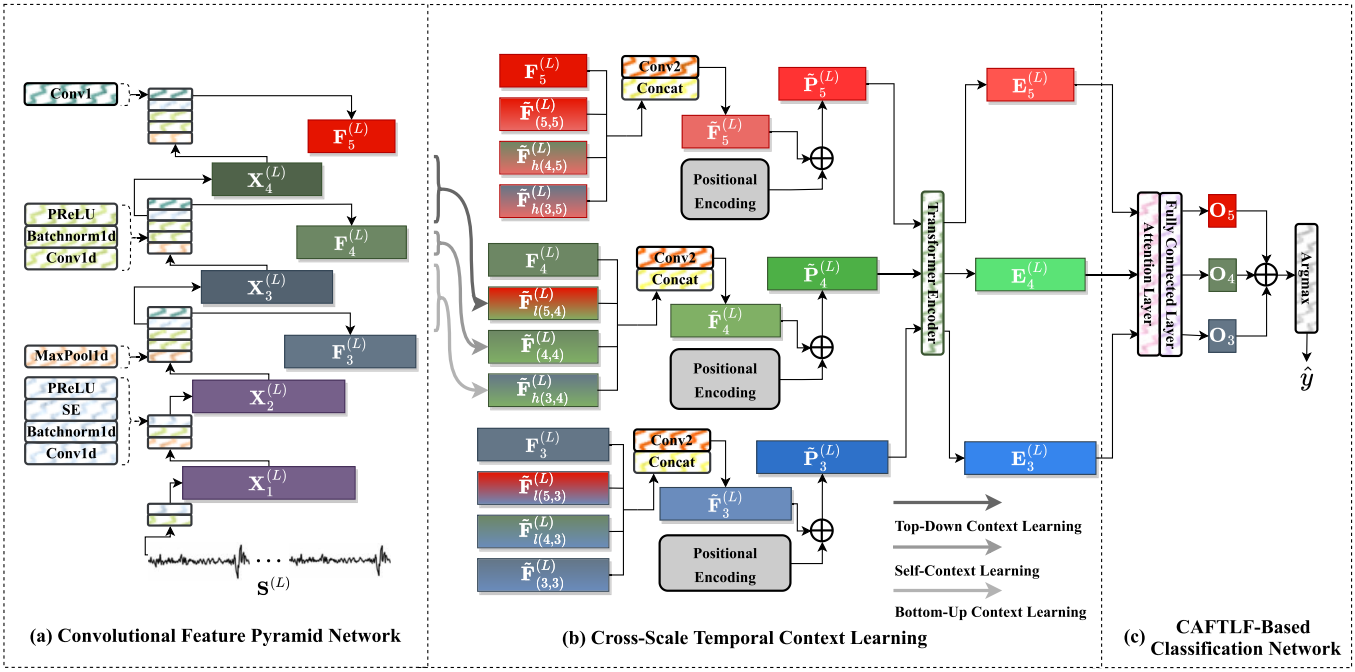

Fig. 1. Overall architecture of SleepFC. At first, CFPN extracts the multi-scale features of salient waves from successive EEG epochs. Then, CSTCL learns to capture sleep stage transition rules from the extracted multi-scale features. In more detail, CSTCL fuses the multi-scale features by SCT, TDCL and BUCL, and encodes the temporal context information of fused features via Transformer encoder. At last, CAFTLF-based classification network distinguishes the imbalanced classes of sleep stages.

These two schemes take into account the transition patterns of neighboring sleep stages and thus achieves encouraging performances [19]. Dong et al. [20] put forward a rectifier neural network to learn the hierarchical features from EEG epochs and adopted long short-term memory (LSTM) to recognize sleep stages. Seo et al. [21] brought forward intra- and inter-epoch temporal context network (IITNet), which is composed of a deep residual network and two layers of bi-directional LSTM (BiLSTM), to extract the time-invariant features from single-channel EEG epochs and learn the sleep transition rules for distinguishing sleep stages. Phan et al. [22] came up with SleepTransformer which extracts the intra-epoch features from each 30-second EEG epoch and learns the inter-epoch temporal representation from these epoch-wise features to separate sleep stages.

Nevertheless, because humans have different sleep durations at different stages, the number of signal samples in each sleep stage is usually unequal. Therefore, we need to address such a class imbalance problem for sleep staging [12], [23]. Recently, some studies suggest using data augmentation to balance the class distribution for sleep datasets [24], [25]. Data augmentation approaches usually generate the synthetic samples of minority classes from existing samples at the expense of computational time. Other studies recommend applying cost-sensitive learning to penalize the misclassification of minority classes, which, however, will sacrifice the classification rate on the majority classes as a cost [17].

In this paper, we propose a novel and effective method named SleepFC for sleep staging based on single-channel EEG. The main contributions of SleepFC are summarized as follows.

1) The proposed SleepFC has a new architecture, which consists of convolutional feature pyramid network (CFPN), cross-scale temporal context learning (CSTCL), and class adaptive fine-tuning loss function (CAFTLF) based classification network, as illustrated in Fig. 1.
2) In SleepFC, CFPN takes charge of learning a feature pyramid of salient waves; CSTCL is responsible for capturing the multi-scale sleep transition rules between successive sleep stages; CAFTLF-based classification network plays the role in resolving the class imbalance problem for sleep staging, without causing extra computational expense or compromising the classification rate on the majority classes.
3) Extensive experiments on three public benchmark datasets demonstrate the superiority of SleepFC over the related state-of-the-arts for sleep staging based on single-channel EEG.

II. Method

As shown in Fig. 1, the proposed SleepFC is comprised of three components: CFPN, CSTCL, and CAFTFL-based classification network. The algorithmic procedures of SleepFC are briefly described as follows. At first, CFPN learns the multi-scale features of salient waves from successive EEG epochs. Then, CSTCL captures the sleep stage transition rules from the multi-scale features. At last, CAFTLF-based classification network predicts the sleep stages whilst tackling the class imbalance problem.

A. Preliminary

We denote L successive single-channel EEG epochs sampled at F Hz as $\mathbf{S}^{(L)} \in \mathbb{R}^{T \cdot F \cdot L \times C}$, where T is the number of seconds of EEG epoch duration and C is the number of EEG

channels, we recommend $T = 30$ and $F = 100$ in our work, following the general research on the issue of EEG-based sleep staging [6], [15], [26], [27]. Besides, we denote the one-hot encoding label of an EEG epoch as $y_s^k \in \{0, 1\}^k$, which corresponds to the true label y_s. Here, we set $k = 5$, following the five-stage sleep classification in the AASM criteria [28].

B. Convolutional Feature Pyramid Network

To characterize the intrinsic features of salient waves from EEG signals, CFPN learns the feature pyramid by means of convolutional blocks, max-pooling layers, and convolutional layers.

The feature pyramid consists of three feature maps $\left\{\mathbf{F}_3^{(L)}, \mathbf{F}_4^{(L)}, \mathbf{F}_5^{(L)}\right\}$, where $\mathbf{F}_i^{(L)} \in \mathbb{R}^{d_{t,i} \times d_c}$, $d_{t,i}$ denotes the temporal dimension of the i-th feature map ($i = 3, 4, 5$), and d_c denotes the channel dimension of feature maps. CFPN involves five convolutional blocks, four max-pooling layers, and three convolutional layers, all of which are designed for unifying the channel dimension of feature maps. Each of the first two convolutional blocks contains two 1-D convolutional layers, two 1-D batch normalization layers, and two parametric rectified linear units (PReLU) [29]; each of the last three convolutional blocks contains three 1-D convolutional layers, three 1-D batch normalization layers, and three PReLUs. In each convolutional block, all the convolution layers have the same kernel size. Besides, a squeeze-and-excitation module is positioned before the last PReLU of every convolutional block. The squeeze-and-excitation module can adaptively recalibrate the channel-wise feature responses by explicitly exploring the inter-dependencies among feature channels [30]. A max-pooling layer is placed between every two convolutional blocks to decrease the temporal dimension of feature maps. Moreover, a 1-D convolutional layer with a kernel size of 1 is put after each of the last three convolutional blocks, for reducing and unifying the channel dimension of feature maps as d_c.

C. Cross-Scale Temporal Context Learning

To learn the EEG features for sleep staging, CSTCL captures the multi-scale sleep transition rules by integrating three context learning approaches and one transformer encoder.

The sleep transition rules have multi-scale characteristics according to the AASM criteria (*i.e.*, short scale: N2–REM; middle scale: N3–N1–N1–N3; long scale: N2–N1–N1–W–N1–W–W; here, "–" means the sleep stage transiting from one to another) [31]. CSTCL learns to capture the multi-scale sleep transition rules from feature pyramid. Specifically, CSTCL contains top-down context learning, self-context learning, bottom-up context learning, and Transformer encoder. As the input of CSTCL, the feature pyramid $\left\{\mathbf{F}_3^{(L)}, \mathbf{F}_4^{(L)}, \mathbf{F}_5^{(L)}\right\}$ contains both fine-grained feature map $\mathbf{F}_l \in \mathbb{R}^{d_{t,l} \times d_c}$ at the low level and coarse-grained feature map $\mathbf{F}_h \in \mathbb{R}^{d_{t,h} \times d_c}$ at the high level. In the feature pyramid $\left\{\mathbf{F}_3^{(L)}, \mathbf{F}_4^{(L)}, \mathbf{F}_5^{(L)}\right\}$, every feature map $\mathbf{F}_i^{(L)}$ consists of a sequence of feature vectors $\left[\mathbf{f}_{i,1}^{(L)}, \mathbf{f}_{i,2}^{(L)}, \ldots, \mathbf{f}_{i,T}^{(L)}\right]^\top$ along the temporal dimension of $\mathbf{F}_i^{(L)}$.

1) Self-Context Learning: Self-context learning (SCL) extracts the features of salient waves from EEG in different sleep stages, and learns the contextual relationship along the temporal dimension of these features. The output $\tilde{\mathbf{F}}$ of SCL has the same size as its input $\mathbf{F}$. Firstly, we use M different learnable matrices to project the input $\mathbf{F}$ to M pairs of query and key. Then, we perform a convolutional operation on the input feature map $\mathbf{F}$ to obtain the value $\mathbf{V}$. Next, we calculate the similarity score between each pair of $\mathbf{Q}_j$ and $\mathbf{K}_j$, which are the j-th pair of query and key, by using mixture of softmaxes (MoS) [32]. The MoS-based normalization is formulated as

$$\begin{aligned} \mathbf{W}_S &= \sum_{j=1}^{M} \pi_j \sigma_1\left(\frac{\mathbf{Q}_j \mathbf{K}_j^\top}{\sqrt{d_k}}\right), \\ [\pi_1, \pi_2, \ldots, \pi_M] &= \sigma_2(\mathrm{w}_{\mathrm{mos},j}^\top \bar{\mathbf{K}}), \\ \mathbf{Q}_j &= \mathbf{K}_j = f_{\mathrm{QKS},j}(\mathbf{F}), \\ \mathbf{V} &= f_{\mathrm{VS}}(\mathbf{F}), \end{aligned} \tag{1}$$

where M denotes the number of learnable linear projection matrices; d_k denotes the channel dimension of $\mathbf{K}$; π_j denotes the j-th aggregation weight; $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ are the Softmax functions; $\mathrm{w}_{\mathrm{mos},j}$ denotes the learnable linear projection vector for normalization; $\bar{\mathbf{K}}$ denotes the arithmetic mean of $\mathbf{K}$ along the temporal dimension; $f_{\mathrm{VS}}(\cdot)$ represents the convolutional operation, and $f_{\mathrm{QKS},j}(\cdot)$ the learnable linear projection matrice. Based on the MoS-based normalization, we can obtain the output feature map $\tilde{\mathbf{F}}$ of SCL as

$$\tilde{\mathbf{F}} = \mathrm{BN}(\mathbf{W}_S \mathbf{V}) + \mathbf{F}, \tag{2}$$

where $\mathrm{BN}(\cdot)$ indicates the batch normalization.

2) Top-Down Context Learning: Top-down context learning (TDCL) adopts a top-down attention mechanism. This mechanism fuses the global information of high-level feature map $\mathbf{F}_h$ and the local information of low-level feature map $\mathbf{F}_l$ together.

The method pipeline of TDCL is briefly described in the following. First, we apply three convolutional layers $f_{\mathrm{QT}}(\cdot)$, $f_{\mathrm{KT}}(\cdot)$ and $f_{\mathrm{VT}}(\cdot)$ with a kernel size of 1 to reduce the channel dimension of $\mathbf{F}_l$ and $\mathbf{F}_h$ to $d_c/2$, thus generating $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. After this process, the temporal dimension of $\mathbf{Q}$ is still $d_{t,l}$, and that of $\mathbf{K}$ and $\mathbf{V}$ is still $d_{t,h}$.

Next, we calculate the dot product between $\mathbf{Q}$ and $\mathbf{K}^\top$ and take the normalization operation to produce the attention score. Then, we multiply the attention score by $\mathbf{V}$ to yield a new feature map. Finally, we utilize a convolutional layer with the kernel size of 1 and the stride size of 1 to increase the channel dimension of this feature map to d_c and keep the temporal dimension as $d_{t,l}$, so that $\tilde{\mathbf{F}}_l \in \mathbb{R}^{d_{t,l} \times d_c}$. The above process can be formulated as follows:

$$\begin{aligned} \tilde{\mathbf{F}}_l &= \mathrm{Conv}_T\left(\frac{\mathbf{Q}\mathbf{K}^\top}{d_{t,h}}\mathbf{V}\right), \\ \mathbf{Q} &= f_{\mathrm{QT}}(\mathbf{F}_l), \\ \mathbf{K} &= f_{\mathrm{KT}}(\mathbf{F}_h), \\ \mathbf{V} &= f_{\mathrm{VT}}(\mathbf{F}_h), \end{aligned} \tag{3}$$

where the size of output $\tilde{\mathbf{F}}_l$ is the same as the input $\mathbf{F}_l$.

3) Bottom-Up Context Learning: Bottom-up context learning (BUCL) fuses the local information of $\mathbf{F}_l$ into $\mathbf{F}_h$. Specifically, $\mathbf{F}_h$ is linearly projected to $\mathbf{Q}$, and $\mathbf{F}_l$ is linearly projected to $\mathbf{K}$ and $\mathbf{V}$:

$$\begin{aligned}\mathbf{Q} &= f_{\mathrm{QB}}(\mathbf{F}_h),\\ \mathbf{K} &= f_{\mathrm{KB}}(\mathbf{F}_l),\\ \mathbf{V} &= f_{\mathrm{VB}}(\mathbf{F}_l),\end{aligned} \tag{4}$$

where $f_{\mathrm{QB}}(\cdot)$, $f_{\mathrm{KB}}(\cdot)$ and $f_{\mathrm{VB}}(\cdot)$ are the learnable matrices.

Next, we process the low-level feature map $\mathbf{K}$ by the channel-wise attention operation:

$$\mathbf{w}_c = \mathrm{ReLU}(\mathrm{GAP}(\mathbf{K})), \tag{5}$$

where the weight $\mathbf{w}_c$ of channel-wise attention is computed by global average pooling (GAP) and ReLU.

Then, we calculate the Hadamard product between $\mathbf{W}_c$ and $\mathbf{Q}$ by

$$\begin{aligned}\tilde{\mathbf{F}}_h &= \mathrm{ReLU}(\mathbf{Q} \odot \mathbf{W}_c + \mathbf{V}),\\ \mathbf{W}_c &= [\mathbf{w}_c, \mathbf{w}_c, \ldots, \mathbf{w}_c]_{1\times d_k},\end{aligned} \tag{6}$$

where d_k denotes the temporal dimension of $\mathbf{K}$, and $\odot$ represents the Hadamard product.

4) Transformer Encoder: By performing SCT, TDCT and BUCL on the feature pyramid $\left\{\mathbf{F}_3^{(L)}, \mathbf{F}_4^{(L)}, \mathbf{F}_5^{(L)}\right\}$, we can obtain three feature sets, each of which contains four feature maps. Then, we concatenate the four feature maps of each set along the channel dimension. To reduce the channel dimension, we process the concatenated feature maps by a convolutional layer ConV$(\cdot)$ to yield the feature maps $\tilde{\mathbf{F}}_3^{(L)}$, $\tilde{\mathbf{F}}_4^{(L)}$ and $\tilde{\mathbf{F}}_5^{(L)}$:

$$\begin{aligned}\tilde{\mathbf{F}}_3^{(L)} &= \mathrm{ConV}(\mathrm{Concat}(\mathbf{F}_3^{(L)}, \tilde{\mathbf{F}}_{l(5,3)}^{(L)}, \tilde{\mathbf{F}}_{l(4,3)}^{(L)}, \tilde{\mathbf{F}}_{(3,3)}^{(L)})),\\ \tilde{\mathbf{F}}_4^{(L)} &= \mathrm{ConV}(\mathrm{Concat}(\mathbf{F}_4^{(L)}, \tilde{\mathbf{F}}_{l(5,4)}^{(L)}, \tilde{\mathbf{F}}_{(4,4)}^{(L)}, \tilde{\mathbf{F}}_{h(3,4)}^{(L)})),\\ \tilde{\mathbf{F}}_5^{(L)} &= \mathrm{ConV}(\mathrm{Concat}(\mathbf{F}_5^{(L)}, \tilde{\mathbf{F}}_{(5,5)}^{(L)}, \tilde{\mathbf{F}}_{h(4,5)}^{(L)}, \tilde{\mathbf{F}}_{h(3,5)}^{(L)})),\end{aligned} \tag{7}$$

where Concat$(\cdot)$ represents the concatenation operation along the channel dimension of the four feature maps in each set, the feature map $\tilde{\mathbf{F}}_i^{(L)}$ consists of a sequence of feature vectors $\left[\tilde{\mathbf{f}}_{i,1}^{(L)}, \tilde{\mathbf{f}}_{i,2}^{(L)}, \ldots, \tilde{\mathbf{f}}_{i,T}^{(L)}\right]^\top$, and T denotes the temporal dimension of $\tilde{\mathbf{F}}_i^{(L)}$.

Finally, we encode the context information of temporal sequence $\left[\tilde{\mathbf{f}}_{i,1}^{(L)}, \tilde{\mathbf{f}}_{i,2}^{(L)}, \ldots, \tilde{\mathbf{f}}_{i,T}^{(L)}\right]$ by Transformer encoder. In the positional encoder of Transformer, we adopt the sine and cosine functions to incorporate the order information of feature vectors $\left[\tilde{\mathbf{f}}_{i,1}^{(L)}, \tilde{\mathbf{f}}_{i,2}^{(L)}, \ldots, \tilde{\mathbf{f}}_{i,T}^{(L)}\right]$ into

$$\begin{aligned}\mathbf{E}_i^{(L)} &= \mathrm{TransformerEncoder}(\tilde{\mathbf{P}}_i^{(L)}),\\ \tilde{\mathbf{P}}_i^{(L)} &= \tilde{\mathbf{F}}_i^{(L)} + \mathbf{P}_i^{(L)},\end{aligned} \tag{8}$$

where $\mathbf{P}_i^{(L)}$ denotes the positional encoding matrix; $\mathbf{E}_i^{(L)}$ denotes the encoded feature map of the i-th feature map $\tilde{\mathbf{P}}_i^{(L)}$; TransformerEncoder$(\cdot)$ represents the encoder component of Transformer. Because of the large number of parameters in Transformer, we reduce the hidden dimension d_{FF} of the feed-forward network. Besides, we retain the original number of attention heads N_h and encoder layers N_e in Transformer [33]. The parameter settings will be detailed in Section III-B.

D. CAFTFL-Based Classification Network

CAFTFL-based classification network predicts the sleep stages whilst handling the class imbalance via the attention mechanism and the loss function CAFTLF in a two-stage training process.

In CAFTFL-based classification network, we fuse the encoded feature map $\mathbf{E}_i^{(L)} = \left[\mathbf{e}_{i,1}^{(L)}, \mathbf{e}_{i,2}^{(L)}, \ldots, \mathbf{e}_{i,T}^{(L)}\right]^\top$ into an attention vector $\tilde{\mathbf{e}}_i$ by the attention layer. More concretely, we combine the feature vectors $\left[\mathbf{e}_{i,1}^{(L)}, \mathbf{e}_{i,2}^{(L)}, \ldots, \mathbf{e}_{i,T}^{(L)}\right]$ via the weighted sum:

$$\tilde{\mathbf{e}}_i = \sum_{t=1}^{T} \alpha_{i,t}\mathbf{a}_{i,t}, \tag{9}$$

where $\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,t}$ denote the attention weights which can be learned by an attention layer:

$$\begin{aligned}\mathbf{a}_{i,t} &= \tanh\left(\mathbf{W}\mathbf{e}_{i,t}^{(L)} + \mathbf{b}\right),\\ \alpha_{i,t} &= \frac{\exp\left(\mathbf{a}_{i,t}^\top \mathbf{w}_\alpha\right)}{\sum_{t=1}^{T}\exp\left(\mathbf{a}_{i,t}^\top \mathbf{w}_\alpha\right)},\end{aligned} \tag{10}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the learnable weight matrix and bias, respectively; $\mathbf{w}_\alpha$ is the trainable weight vector.

Finally, the i-th feature vector $\tilde{\mathbf{e}}_i$ passes through a fully connected layer to yield the i-th output logit $\mathbf{O}_i$, and thus the sleep stage can be predicted via

$$\hat{y} = \mathrm{argmax}\left(\sum_{i\in\{3,4,5\}} \mathbf{O}_i\right), \tag{11}$$

where $\hat{y}$ denotes the predicted sleep stage.

In the classification network, we employ a piecewise loss function to address the class imbalance problem of sleep stages. The training process of SleepFC consists of two stages. In the first stage, we utilize the standard multi-class cross-entropy [34] as the loss function; in the second stage, we devise the loss function CAFTLF as

$$\mathcal{L}_{\mathrm{CAFTLF}} = -\frac{1}{S}\sum_{\{i\in 3,4,5\}}\sum_{s=1}^{S}\sum_{k=1}^{K} w_k y_{i,s}^k \log\left(\hat{y}_{i,s}^k\right), \tag{12}$$

$$w_k = \begin{cases} 1 + \mu_{(y_s,\hat{y}_s)} \cdot \max\left(1, \log\left(\dfrac{S}{S_k}\right)\right), & \text{if } y_s \neq \hat{y}_s\\ 1, & \text{if } y_s = \hat{y}_s,\end{cases} \tag{13}$$

where $\hat{y}_s^k$ denotes the predicted probability of the s-th sample belonging to class k; S denotes the total number of samples, and K denotes the total number of classes; w_k denotes the weight assigned to class k; S_k denotes the number of samples belonging to class k; $\mu_{(y_s,\hat{y}_s)}$ is an adjustable parameter indicating the distinctness of the class.

When training SleepFC, an early stopping technique is employed to reduce the overfitting risk and enhance the generalization performance. In the training process, once the

TABLE I
DATASET CHARACTERISTICS AND EVALUATION PROTOCOLS

Datasets	Number of Subjects	Cross-Validation		Used Channel	Class Distribution (Normal font – Class Size; Italic Font – Class Size Ratio)					
		Number of Folds	Training/Validation/Testing Sets Every Fold		WAKE	N1	N2	N3	REM	Total
SleepEDF-20	20	20-Fold	15/4/1 Subjects	Fpz-Cz	8285 *19.61%*	2804 *6.63%*	17799 *42.07%*	5703 *13.48%*	7717 *18.23%*	42308
SleepEDF-78	78	10-Fold	63/7/8 Subjects	Fpz-Cz	65501 *33.58%*	21522 *11.03%*	69132 *35.40%*	13039 *6.68%*	25835 *13.21%*	195029
ISRUC-S3	10	10-Fold	7/2/1 Subjects	C4-A1	1676 *19.53%*	1215 *14.13%*	2616 *30.43%*	2016 *23.47%*	1066 *12.43%*	8589

validation loss stops decreasing not less than a certain number of training iterations (*i.e.*, the early stopping patience ϕ_1), the first stage of training ends and the second stage of training starts. In the second training stage, if the validation accuracy ceases to increase not less than a certain number of training iterations (*i.e.*, the early stopping patience ϕ_2), which indicates that the trained model can no longer be improved, then the training process ends. In the second training stage, the class weight w_k is influenced by two factors: first, the number of samples in each class; second, the classification rate for each class in the first training stage. So, CAFTLF can depress the over-belief of classification network in the classification rate while neglecting the class size, which is thereby conducive to overcoming the imbalanced classification problem in sleep staging. In addition, the hyperparameters for training SleepFC will be detailed in Section III-B.

III. EXPERIMENTS

A. Datasets

We evaluate our proposed method, SleepFC, on three public benchmark datasets: SleepEDF-20 [26], SleepEDF-78 [27], and ISRUC-S3 [35], whose critical characteristics have been summarized in Table I.

SleepEDF-20: SleepEDF-20 is comprised of 10 male subjects and 10 female subjects aged from 25 to 34 years old without sleep disorders. Two consecutive nights of PSG recordings were collected from them, except that one recording of subject 13 was lost due to device failure. Based on the Rechtschaffen and Kales criteria [6], sleep experts manually annotated the PSG sleep periods in 30-second sleep epochs and categorized the sleep epochs into eight classes: MOVEMENT, UNKNOWN, WAKE, N1, N2, N3, N4, and REM.

SleepEDF-78: SleepEDF-78 is the Sleep-EDF Expanded dataset (version 2013), consisting of 78 healthy subjects aged from 25 to 101. Each subject underwent two consecutive nights of PSG sleep recordings, except for subjects 13, 36 and 52, whose one recording was lost due to device failure. Every sleep epoch was categorized into the same eight classes as SleepEDF-20.

ISRUC-S3: ISRUC-S3 contains the PSG recordings collected from 10 healthy subjects (9 males and 1 female). The recordings of ISRUC-S3 lasted continually for 8 hours with a sampling frequency of 200 Hz. Each recording includes 6 EEG channels, 1 ECG channel, 3 EMG channels, and 2 EOG channels. According to the criteria of AASM, sleep experts categorized these PSG signals into five sleep stages: WAKE, N1, N2, N3, and REM.

TABLE II
EVALUATION OF FEATURE EXTRACTION METHOD FEATURE PYRAMID IN SLEEPFC

Backbones	Representations	SleepEDF-78		
		ACC (%)	MF1 (%)	κ
U-Time [16]	Original Features	82.7	76.0	0.761
	Feature Pyramid	83.0	77.4	0.766
XSleepNet [6]	Original Features	82.8	76.5	0.761
	Feature Pyramid	83.2	77.5	0.769
AttnSleep [17]	Original Features	78.8	69.8	0.704
	Feature Pyramid	79.0	70.4	0.706
SleepFC	$\mathbf{F}_5^{(L)}$	83.2	77.3	0.767
	Feature Pyramid	**83.6**	**77.8**	**0.772**

B. Experimental Settings

For each dataset, we use one single channel of original EEG, except for ISRUC-S3 whose signals are downsampled at the frequency of 100 Hz. In experiments, we use the Fpz-Cz channel of EEG from SleepEDF-20 and SleepEDF-78, and the C4-A1 channel of EEG from ISRUC-S3 for method evaluation. The MOVEMENT class refers to the physical activity during sleep. There are also the movement artifacts that cannot be scored in both beginning and end of the recording from each subject. These noisy parts of each recording are labeled as UNKNOWN [36]. Because these two classes don't represent any specific sleep stage, we exclude them before experiments [6], [31], [37]. Moreover, according to the AASM criteria, we merge N3 and N4 stages into N3 for classification [6], [17], [38], [39]. Besides, we keep 30 minutes of the WAKE periods before and after the sleep period as the WAKE stage [40].

We follow the universally-used evaluation protocols for method evaluation [6], [17], [22]. The evaluation protocols on different datasets have been described in Table I. It is worth mentioning that, in experiments, the validation set is randomly selected from the training set, which is independent of the testing set. Besides, we adopt three metrics to evaluate the method performance: accuracy (ACC), macro F1-score (MF1), and Cohen's Kappa (κ) [6], [22], [36].

The parameter settings of SleepFC are given in the following. L is set as 10, which means that one current and nine previous adjacent EEG epochs are used as the input data of SleepFC. In each convolutional block of CFPN, for every convolutional layer, the kernel size is set as 3, the stride size is set as 1, and the padding size is set as 1; for every max-pooling layer, the kernel size is set as 5, and the stride size is set as 5. In CFPN, the output channel number d_{cr} of the convolutional layer is set as 128. In CSTCL, all the outputs

TABLE III

CONFUSION MATRICES OF SLEEPFC FOR SLEEP STAGE CLASSIFICATION ON SLEEPEDF-20, SLEEPEDF-78 AND ISRUC-S3 (IN EACH CONFUSION MATRIX, THE ROW STANDS FOR THE GROUND-TRUTH LABELS, AND THE COLUMN STANDS FOR THE PREDICTED CLASSES; THE ABOVE VALUE INDICATES THE CLASSIFICATION RATE ON EACH CLASS, AND THE BELOW VALUE INDICATES THE NUMBER OF PREDICTED SAMPLES IN EACH CLASS)

Datasets	Sleep-EDF-20					Sleep-EDF-78					ISRUC-S3				
	W	N1	N2	N3	R	W	N1	N2	N3	R	W	N1	N2	N3	R
W	89.82% 7126	5.70% 452	1.54% 122	0.78% 62	2.17% 172	93.52% 60387	4.78% 3084	0.86% 556	0.05% 35	0.79% 512	88.59% 1405	8.39% 133	2.65% 42	0.25% 4	0.13% 2
N1	16.62% 466	47.97% 1345	22.68% 636	0.43% 12	12.30% 345	17.75% 3821	45.65% 9825	28.19% 6068	0.24% 51	8.16% 1757	12.18% 148	55.88% 679	18.11% 220	0.00% 0	13.83% 168
N2	0.61% 108	1.67% 298	90.65% 16134	4.38% 780	2.69% 479	0.67% 461	4.80% 3317	88.76% 61362	2.63% 1821	3.14% 2171	1.45% 38	7.61% 199	81.61% 2135	7.76% 203	1.57% 41
N3	0.28% 16	0.02% 1	10.66% 608	89.04% 5078	0.00% 0	0.18% 24	0.09% 12	20.41% 2661	78.76% 10269	0.56% 73	0.15% 3	0.10% 2	14.83% 299	84.72% 1708	0.20% 4
R	0.67% 52	2.27% 175	9.29% 717	0.54% 42	87.22% 6731	1.27% 327	5.08% 1312	10.28% 2657	0.02% 4	83.36% 21535	0.66% 7	13.41% 143	8.44% 90	0.66% 7	76.83% 819

of SCT, TDCL and BUCL have the same channel dimension $d_{cr} = 128$. In SCT, the number of mixture models in MoS is set as 2. In Transformer encoder, the number of heads is set as $N_h = 8$, and the number of encoder layers is set as $N_e = 6$. The hidden dimension of the feed-forward network d_{FF} is set as 128. Besides, SleepFC is trained using the Adam optimizer [41] with $\eta = 5 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. In training, the mini-batch size of SleepFC is set as 32. To mitigate overfitting, the L2-weight regularization with a coefficient of 1×10^{-6} is adopted for SleepFC.

On SleepEDF-20 and SleepEDF-78, SleepFC is evaluated on the validation set every 500 training iterations (*i.e.*, the validation period $\psi = 500$); and on ISRUC-S3, SleepFC is evaluated with $\psi = 150$. At the same time, the validation loss is also monitored for early stopping. The first stage of training focuses on minimizing the validation loss. If $\phi_1 = 20$, the first training stage ends and the second training stage starts. The second stage of training turns to maximize the validation accuracy, $\eta = 1 \times 10^{-4}$ and $\phi_2 = 10$.

C. Feature Evaluation

To evaluate the performance gain brought by CFPN, we compare the feature extraction components of SleepFC, U-Time, XSleepNet, and SleepTransformer with and without the feature pyramid method on SleepEDF-78. This comparison is carried out under the condition where the subsequent components of the feature extraction components are utilizing Transformer Encoder and the CAFTFL-based classification network of SleepFC.

U-Time [15] is a fully convolutional network for sleep staging. U-Time has an encoder-decoder structure for feature extraction, and the encoder is used for feature extraction and the decoder for times series segmentation. In our experiments, we only utilized the encoder component to extract EEG features directly from the raw EEG signal.

XSleepNet [6] is a sequence-to-sequence bidirectional RNN for sleep staging. XSleepNet is composed of two network streams: one for processing raw signals and the other for processing time-frequency images. In our experiments, we only use the former stream to extract features, considering its suitability for EEG.

AttnSleep [17] is an attention-based deep learning approach for sleep staging using single-channel EEG. The feature extraction component of AttnSleep is a multi-resolution convolutional neural network (MRCNN), which is bifurcated into two distinct branches. The low-resolution branch extracts low-frequency features, and the high-resolution branch extracts high-frequency features. The features from the two branches are then concatenated as the extracted features.

From Table II, we can see that, with feature pyramid, the overall ACC, MF1, and κ performances of all the evaluated methods consistently rise. These results not only validate the competency of CFPN in SleepFC, but also verify the compatibility of feature pyramid with all the compared networks for sleep staging.

D. Method Comparison

Table III has visualized the confusion matrices of SleepFC for sleep stage classification on SleepEDF-20, SleepEDF-78 and ISRUC-S3. From these confusion matrices, we can observe that the class imbalance problem has a big influence on the performance of SleepFC. Specifically, it is indeed the easiest case to identify the sleep stage W, which belongs to the majority class in the long-tailed distribution, on all the datasets, while, it is also the hardest case to classify the sleep stage N1, which belongs to the minority one at the other end of such a class distribution.

Moreover, we compare our proposed SleepFC with the state-of-the-art approaches in Table IV. We directly report the results of the methods with the input setting of $L = 10$ in their original papers, including IITNet [21] and SleepEEGNet [13]. For those methods having a different setting of L, we also evaluate them based on the input data of $L = 10$ for fairness. In particular, we implement AttnSleep [17], Multi-Task CNN [42], TinySleepNet [37], XSleepNet [6] and U-Time [15] using public-available codes, and reproduce DeepSleepNet [36], ResnetLSTM [43], SleepFCN [44], Single-Stream XSleepNet [6], SleepTransformer [22], TSA-Net [39], MNN [20] and SeqSleepNet [10] by ourselves.

From Table IV, we can see that SleepFC performs the best for sleep staging in terms of ACC, MF1 and κ on the whole. In greater detail, SleepFC achieves the remarkable F1-Scores performances on N1, N2 and REM. Besides, the results of SleepFC on WAKE and N3 are also relatively encouraging.

TABLE IV
COMPARISON OF SLEEPFC WITH RELATED STATE-OF-THE-ARTS FOR SLEEP STAGING

Datasets	Methods	L	Evaluation Metrics			F1-Score (%) Every Class				
			ACC (%)	MF1 (%)	κ	WAKE	N1	N2	N3	REM
SleepEDF-20	AttnSleep† [17]	1	84.4	78.1	0.790	89.7	42.6	88.8	**90.2**	79.0
	AttnSleep [17]	10	85.5	80.8	0.801	90.2	51.8	88.3	88.9	84.9
	DeepSleepNet† [36]	1	82.0	76.9	0.760	84.7	46.6	85.9	84.8	82.4
	DeepSleepNet [36]	10	79.1	72.9	0.713	81.2	40.4	82.9	80.0	79.9
	IITNet† [21]	10	83.9	77.6	0.780	87.7	43.4	87.7	86.7	82.5
	Multi-Task CNN† [42]	3	80.9	73.6	0.731	81.7	40.4	87.1	73.4	85.2
	Multi-Task CNN [42]	10	79.4	71.5	0.710	81.7	33.8	85.1	73.2	84.0
	ResnetLSTM [43]	1	82.8	77.3	0.765	88.8	44.0	87.9	88.4	77.3
		10	85.6	81.4	0.807	90.3	52.6	88.5	87.5	85.7
	SleepEEGNet† [14]	10	84.3	79.7	0.790	89.2	52.2	86.8	85.1	85.0
	SleepFCN [44]	1	82.9	78.3	0.771	88.9	47.2	87.4	89.1	79.1
		10	84.2	80.4	0.788	89.0	**54.6**	86.8	85.5	86.0
	TinySleepNet† [37]	15	85.4	80.5	0.800	90.1	51.4	88.5	88.3	84.3
	TinySleepNet [37]	10	82.7	77.3	0.762	85.7	47.8	87.2	83.9	81.8
	XSleepNet† [6]	20	86.3	80.6	0.813	**92.2**	51.8	88.0	86.8	83.9
	XSleepNet* [6]	10	83.9	79.7	0.781	86.9	51.5	87.6	87.8	84.5
	XSleepNet [6]	10	84.9	79.4	0.794	89.4	48.1	88.5	87.6	83.4
	SleepFC (Ours)	10	**86.8**	**81.5**	**0.818**	90.8	53.0	**89.6**	87.0	**87.2**
SleepEDF-78	AttnSleep† [17]	1	81.3	75.1	0.740	92.0	42.0	85.0	**82.1**	74.2
	AttnSleep [17]	10	82.9	78.1	0.765	92.2	**51.3**	85.5	80.5	81.1
	DeepSleepNet [36]	10	76.0	66.7	0.664	89.2	42.2	78.6	54.2	69.2
	SleepEEGNet† [14]	10	80.0	73.6	0.730	91.7	44.1	82.5	73.5	76.1
	SleepTransformer† [22]	21	81.4	74.3	0.743	91.7	40.4	84.3	77.9	77.2
	SleepTransformer [22]	10	78.7	69.5	0.700	90.3	34.2	82.3	68.7	72.2
	TinySleepNet† [37]	15	83.1	78.1	0.770	92.8	51.0	85.3	81.1	80.3
	TinySleepNet [37]	10	81.9	76.3	0.751	91.9	49.8	84.3	76.9	78.8
	TSA-Net [39]	1	79.4	72.4	0.714	90.6	36.3	83.6	80.6	70.8
		10	80.4	75.2	0.732	89.9	45.1	83.7	78.5	78.6
	U-Time† [16]	35	81.3	76.3	0.745	92.0	51.0	83.5	74.6	80.2
	U-Time [16]	10	78.4	73.1	0.705	89.4	43.5	81.5	72.6	78.4
	XSleepNet† [6]	20	84.0	77.9	0.778	**93.3**	49.9	86.0	78.7	81.8
	XSleepNet* [6]	10	79.6	74.3	0.723	91.5	50.1	82.1	74.1	73.5
	XSleepNet [6]	10	82.7	76.8	0.762	92.3	47.5	85.4	79.1	79.9
	SleepFC (Ours)	10	**84.2**	**78.8**	**0.781**	93.2	50.3	**86.2**	81.4	**83.0**
ISRUC-S3	AttnSleep [17]	10	73.9	71.2	0.663	83.4	43.7	74.0	85.7	69.3
	DeepSleepNet [36]	10	62.8	61.0	0.541	83.7	46.7	40.2	77.3	56.9
	MNN [20]	5	74.0	71.7	0.667	84.7	47.0	73.1	83.4	70.5
		10	72.6	69.4	0.648	82.8	41.6	72.6	84.1	66.1
	SeqSleepNet [10]	30	74.0	71.4	0.664	84.1	49.8	72.7	85.0	65.3
		10	75.7	73.3	0.687	86.3	51.9	75.7	86.6	65.9
	XSleepNet* [6]	10	70.4	68.4	0.621	82.2	45.2	70.0	81.2	63.4
	XSleepNet [6]	10	71.0	69.5	0.631	85.1	48.5	69.0	83.6	61.0
	SleepFC (Ours)	10	**79.4**	**77.8**	**0.734**	**88.2**	**57.3**	**79.0**	**86.7**	**78.0**

Notes: 'XSleepNet*' represents the variant of XSleepNet with single one raw-signal stream; '†' indicates that the L setting and the corresponding results are directly derived from the original paper of this method.

Actually, the sleep stage N1 is a challenging minority class, which only accounts for 5% – 15% of the total sleep time. Even so, SleepFC still obtains a relatively high F1-score on N1. These results readily demonstrate the ability of SleepFC to deal with the class imbalance problem in sleep staging.

Furthermore, we measure the model size of SleepFC in Table V. Although the performance advantage of SleepFC over XSleepNet is not so obvious as the compared approaches, yet SleepFC has smaller parameter amount and requires fewer EEG epochs.

E. Model Ablation

We carry out ablation study to validate the rationality and effectivity of the key components CFPN, CSTCL and

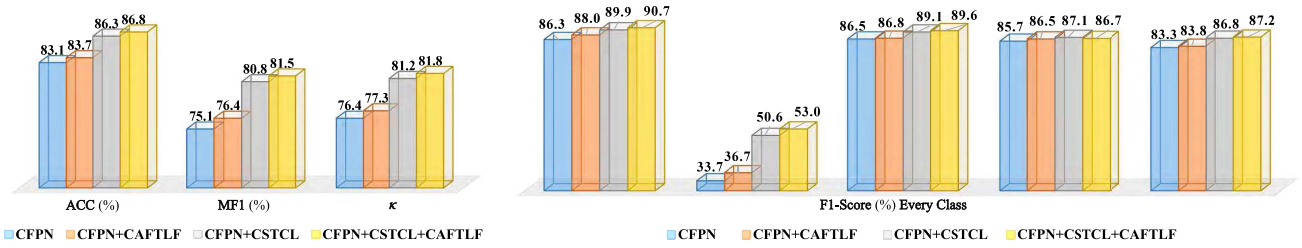

Fig. 2. Ablation of SleepFC on SleepEDF-20.

TABLE V
MODEL SIZE OF SLEEPFC

Models	L	Input Data	Parameter Amount
SeqSleepNet [10]	30	Spectrograms	1.37×10^5
SleepTransformer [22]	21	Spectrograms	3.70×10^6
XSleepNet [6]	20	Raw Signals and Spectrograms	5.74×10^6
SleepFC	10	Raw Signals	3.68×10^6

CAFTLF in SleepFC on SleepEDF-20. The following four experiments are conducted:

1) Ablation on CFPN: CFPN and CAFTLF-disabled classification network with the first training stage.

2) Ablation on CFPN+CAFTLF: CFPN and CAFTLF-based classification network with the two-stage training.

3) Ablation on CFPN+CSTCL: CFPN, CSTCL and CAFTLF-disabled classification network with the first training stage.

4) Ablation on CFPN+CSTCL+CAFTLF: CFPN, CSTCL and CAFTLF-based classification network with the two-stage training, *i.e.*, SleepFC.

From Fig. 2, we can see that CSTCL avails SleepFC of capturing the informative multi-scale transition rules between sleep stages, thus boosting the performance of SleepFC. These results reveal the value of this context learning component in SleepFC for sleep staging. By comparing CFPN and CFPN+CAFTLF as well as comparing CFPN+CAFTLF and CFPN+CSTCL+CAFTLF, we can find that CAFTLF-based classification network not only can enhance the overall ACC, MF1, and κ performances of SleepFC, but also can significantly improve its F1-Score for N1 classification in spite of the severe class imbalance problem. Such results confirm that CAFTLF enables SleepFC to attach importance to the minority class but without compromising its performance on the majority classes.

F. Sensitivity Analysis

1) Evaluation on the Number of EEG Epochs: We evaluate the influence of the number of EEG epochs, denoted as L, on the performance of SleepFC, by adjusting L as 1, 2, 5, 10 and 20. Our results under three different evaluation metrics, as illustrated in Fig. 3, reveal that SleepFC achieves its peak performance on SleepEDF-20, SleepEDF-78 and ISRUC-S3 when L is set as 10. By contrast, both increase and decrease of L result in a performance decline of SleepFC. This is mainly because the lower values of L cannot offer sufficient

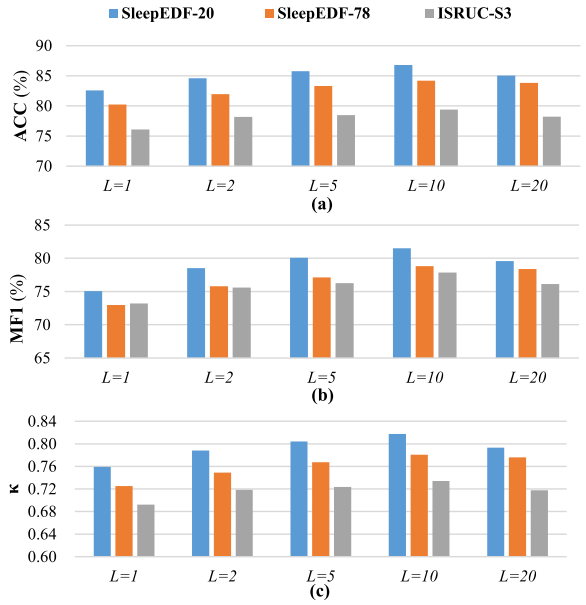

Fig. 3. Evaluation on the number of EEG epochs as the input for SleepFC using SleepEDF-20, SleepEDF-78 and ISRUC-S3: (a) the results of SleepFC under ACC; (b) the results of SleepFC under MF1; (c) the results of SleepFC under κ.

temporal context information for CSTCL of SleepFC to learn discriminative feature maps $\left\{\mathbf{E}_3^{(L)}, \mathbf{E}_4^{(L)}, \mathbf{E}_5^{(L)}\right\}$ from the feature pyramid, while the higher values of L will involve much redundant and noisy information to harm the discriminability of learned feature maps $\left\{\mathbf{E}_3^{(L)}, \mathbf{E}_4^{(L)}, \mathbf{E}_5^{(L)}\right\}$. As a compromise, $L = 10$ is the relatively best choice for SleepFC.

2) Evaluation on the Convolution Kernel Size of CFPN: We evaluate the influence of the convolution kernel size of CFPN, denoted as K, on the performance of SleepFC, by adjusting K from 1 to 9. By observing the results of SleepFC under three different evaluation metrics on ISRUC-S3 in Fig. 4, we can find that the performance of SleepFC fluctuates with the increase of K, resulting in more than one peak. The reason to explain this phenomenon is as follows. For a sleep stage, the larger convolution kernel size enables CFPN to encode the rich and varied information, thus being beneficial for CFPN to learn more robust features at the expense of discriminability; the smaller convolution kernel size enables CFPN to encode the detailed and typical information, thus being conducive to CFPN to learn more discriminative features at the cost of robustness. To ensure a good generalization performance, CFPN should balance both discriminability and robustness

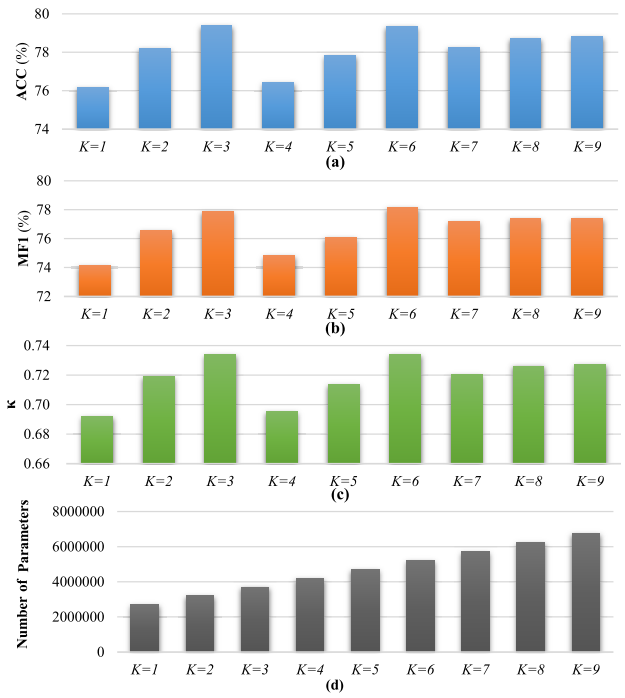

Fig. 4. Evaluation on the convolution kernel size of CFPN in SleepFC using ISRUC-S3: (a) the results of SleepFC under ACC; (b) the results of SleepFC under MF1; (c) the results of SleepFC under κ; (d) the number of parameters in SleepFC with different size of convolution kernels.

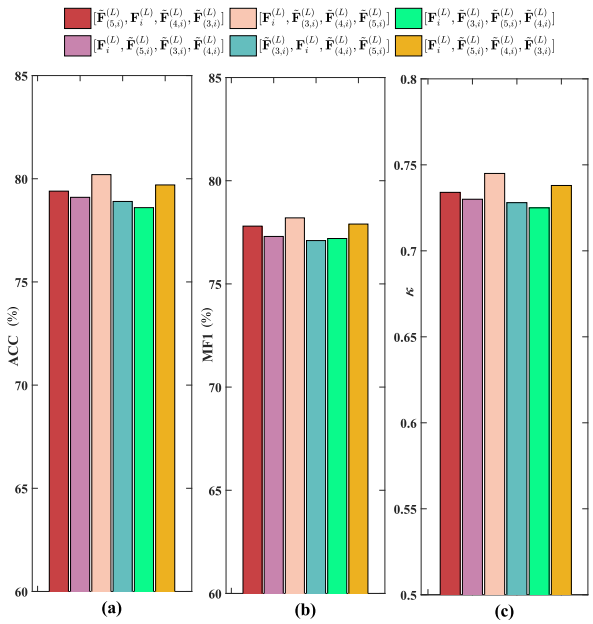

Fig. 5. Evaluation on the in SleepFC using ISRUC-S3: (a) the results of SleepFC under ACC; (b) the results of SleepFC under MF1; (c) the results of SleepFC under κ.

in feature learning. Moreover, the signal data in different sleep stages have different characteristics, commensurate with different kernel sizes of CFPN to learn the features with the strongest generalizability. Therefore, SleepFC exhibits a fluctuating performance as K increases. However, as shown in Fig. 4(d), the larger kernel size also means the more model parameters and computational complexity of SleepFC. Considering this, we recommend $K = 3$ for SleepFC, because this is the smallest kernel size for SleepFC to obtain the relatively highest ACC, MF1, and κ.

3) Evaluation on the Concatenation Order of Feature Maps: We evaluate the concatenation order of feature maps on ISRUC-S3, including $[\tilde{\mathbf{F}}^{(L)}_{(5,i)}, \mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}, \tilde{\mathbf{F}}^{(L)}_{(3,i)}]$, $[\mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(5,i)}, \tilde{\mathbf{F}}^{(L)}_{(3,i)}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}]$, $[\mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(3,i)}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}, \tilde{\mathbf{F}}^{(L)}_{(5,i)}]$, $[\tilde{\mathbf{F}}^{(L)}_{(3,i)}, \mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}, \tilde{\mathbf{F}}^{(L)}_{(5,i)}]$, $[\mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(3,i)}, \tilde{\mathbf{F}}^{(L)}_{(5,i)}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}]$, and $[\mathbf{F}^{(L)}_{i}, \tilde{\mathbf{F}}^{(L)}_{(5,i)}, \tilde{\mathbf{F}}^{(L)}_{(4,i)}, \tilde{\mathbf{F}}^{(L)}_{(3,i)}]$. From Fig. 5, we can see that the concatenation order of feature maps has nearly no influence on the performance of SleepFC under different evaluation metrics on ISRUC-S3. Generally, the concatenation order of feature maps in deep learning will affect the classification performance, so long as the downstream layers learn the deep representation relying on the position of concatenated feature maps [45], [46], [47]. Nevertheless, SleepFC concatenates the four feature maps along the channel dimension instead of the temporal dimension, so the concatenation order has no impact on the process of temporal feature learning. Actually, the minor performance variation of SleepFC for different feature map concatenation is mainly caused by the random initialization of convolution layers after concatenation, which is inevitable and negligible in applications.

4) Evaluation on the Scale of Learned Representation: We evaluate the effectivity of each scale of learned representation output from SleepFC on ISRUC-S3. For convenience, we denote the three scales of learn representations as $\mathbf{O}_3$, $\mathbf{O}_4$, and $\mathbf{O}_5$, which correspond to the output logits of SleepFC, as shown in Fig. 1(c). As reported in Table VI, the results exhibit the performance enhancement of SleepFC with the increment of learned representation scales; when all the three scales of representations $\mathbf{O}_3$, $\mathbf{O}_4$ and $\mathbf{O}_5$ are used together, SleepFC performs the best. Such results validate the effectiveness of the multi-scale representations learned by SleepFC for sleep staging. Further, we can observe the contribution of different scales of learned representations to sleep stage prediction. In greater detail, $\mathbf{O}_3$ is especially effective to classify the WAKE and N3 stages, and $\mathbf{O}_4$ is particularly effective to classify the N1 stage, while $\mathbf{O}_5$ is effective for classifying all the sleep stages on the whole without special superiority in any class. Since different scales of learned representations have different classification advantages, combining these representations together, as done by SleepFC, can well integrate their advantages and hence achieves the best performance for sleep staging.

G. Significance Test

We evaluate the statistical significance of the performance improvement of SleepFC over the three related advanced methods AttnSleep, DeepSleepNet, and XSleepNet by means of the paired Wilcoxon signed-rank test. To be specific, we assess the p-values for ACC improvement, MF1 improvement, and κ improvement of SleepFC in comparison to the three methods. To this end, we set the null hypothesis H_0 as follows: the performance difference between SleepFC and each compared model in the control group is not significant; if the p-value is less than 0.05, H_0 will be rejected. In statistical significance

TABLE VI
EVALUATION ON THE SCALE OF LEARNED REPRESENTATION OUTPUT FROM SLEEPFC

Representation Scales	Evaluation Metrics			F1-Score (%) Every Class				
	ACC (%)	MF1 (%)	κ	WAKE	N1	N2	N3	REM
Representation Scale $\mathbf{O}_3$	78.7	77.0	0.725	87.8	55.6	78.4	86.6	77.5
Representation Scale $\mathbf{O}_4$	78.9	77.5	0.728	86.9	**57.9**	78.4	85.9	77.3
Representation Scale $\mathbf{O}_5$	78.3	76.7	0.711	87.1	55.8	78.2	86.1	77.4
Representation Scales ($\mathbf{O}_3$, $\mathbf{O}_4$)	79.0	77.5	0.729	87.6	57.5	78.5	86.3	77.8
Representation Scales ($\mathbf{O}_3$, $\mathbf{O}_5$)	79.0	77.5	0.729	87.4	57.1	78.4	86.5	77.9
Representation Scales ($\mathbf{O}_4$, $\mathbf{O}_5$)	78.9	77.5	0.728	87.8	57.7	78.2	86.2	77.7
Representation Scales ($\mathbf{O}_3$, $\mathbf{O}_4$, $\mathbf{O}_5$)	**79.4**	**77.8**	**0.734**	**88.2**	57.3	**79.0**	**86.7**	**78.0**

Note: the learned representation is output from the fully connected layer of CAFTLF-based classification network in SleepFC.

TABLE VII
STATISTICAL SIGNIFICANCE TESTS ON THE PERFORMANCE IMPROVEMENT OF SLEEPFC

Datasets	Methods	ACC (%) / Improvement (%) [p-values]	MFI (%) / Improvement (%) [p-values]	κ / Improvement (%) [p-values]
SleepEDF-20	AttnSleep	85.5 (+1.3) [0.00271]	80.8 (+0.7) [0.06372]	0.801 (+0.017) [0.01069]
	DeepSleepNet	79.1 (+7.7) [0.00004]	72.9 (+8.6) [0.00010]	0.713 (+0.105) [0.00006]
	XSleepNet*	83.9 (+2.9) [0.00013]	79.7 (+1.8) [0.16496]	0.781 (+0.037) [0.00032]
	XSleepNet	84.9 (+1.9) [0.01923]	79.4 (+2.1) [0.03684]	0.794 (+0.024) [0.03623]
	SleepFC	86.8 / —	81.5 / —	0.818 / —
SleepEDF-78	AttnSleep	82.9 (+1.3) [0.01367]	78.1 (+0.7) [0.00977]	0.765 (+0.016) [0.01367]
	DeepSleepNet	76.0 (+8.2) [0.00195]	66.7 (+12.1) [0.00195]	0.664 (+0.117) [0.00195]
	XSleepNet*	79.6 (+4.6) [0.00195]	74.3 (+4.5) [0.00195]	0.723 (+0.058) [0.00195]
	XSleepNet	82.7 (+1.5) [0.00977]	76.8 (+2.0) [0.00977]	0.762 (+0.019) [0.00977]
	SleepFC	84.2 / —	78.8 / —	0.781 / —
ISRUC-S3	AttnSleep	73.9 (+5.5) [0.04883]	71.2 (+6.6) [0.04883]	0.663 (+0.071) [0.04883]
	DeepSleepNet	62.8 (+16.6) [0.00391]	61.0 (+16.8) [0.00586]	0.541 (+0.193) [0.00195]
	XSleepNet*	70.4 (+9.0) [0.04883]	68.4 (+9.4) [0.06445]	0.621 (+0.113) [0.04883]
	XSleepNet	71.0 (+8.4) [0.03791]	69.5 (+8.3) [0.04883]	0.631 (+0.103) [0.04883]
	SleepFC	79.4 / —	77.8 / —	0.734 / —

Notes: '+' represents the positive performance improvement brought by SleepFC; 'XSleepNet*' indicates the variant of XSleepNet with single one raw-signal stream.

tests, all the methods adopt the same input of EEG epochs with $L = 10$.

As recorded in Table VII, in almost all the cases, SleepFC has obvious performance improvements over the compared approaches, and the p-values for these improvements are much lower than the significance level of 0.05. Such results straightforwardly evidence the statistical significance of the performance superiority of SleepFC for the task of sleep staging.

IV. CONCLUSION

In this paper, we have proposed a novel method SleepFC for the issue of single-EEG-based sleep staging. SleepFC not only can effectively extract and fuse the representative features from the salient waves of EEG epochs, but can learn and capture the informative multi-scale sleep transition rules among sleep stages, and also can competently tackle the serious class imbalance problem ever haunting this issue. Experimental results on three public benchmark datasets have demonstrated the superiority of proposed method over the related state-of-the-arts. In future, we will tentatively incorporate an appropriate transfer learning strategy into SleepFC to handle the thorny problem of cross-subject domain gap, so as to further enhance the performance of our model for this issue.

REFERENCES

[1] S. Xu et al., "A review of automated sleep disorder detection," *Comput. Biol. Med.*, vol. 150, pp. 106100(1)–106100(20), Nov. 2022.

[2] R. Lefter, R. O. Cojocariu, A. Ciobica, I.-M. Balmus, I. Mavroudis, and A. Kis, "Interactions between sleep and emotions in humans and animal models," *Medicina*, vol. 58, no. 2, pp. 274(1)–274(17), Feb. 2022.

[3] M. Vandekerckhove and Y.-L. Wang, "Emotion, emotion regulation and sleep: An intimate relationship," *AIMS Neurosci.*, vol. 5, no. 1, pp. 1–17, Dec. 2017.

[4] N. F. Watson and C. R. Fernandez, "Artificial intelligence and sleep: Advancing sleep medicine," *Sleep Med. Rev.*, vol. 59, pp. 101512(1)–101512(14), Oct. 2021.

[5] Z. Liu, A. Alavi, M. Li, and X. Zhang, "Self-supervised contrastive learning for medical time series: A systematic review," *Sensors*, vol. 23, no. 9, pp. 4221(1)–4221(34), May 2023.

[6] H. Phan, O. Y. Chén, M. C. Tran, P. Koch, A. Mertins, and M. De Vos, "XSleepNet: Multi-view sequential model for automatic sleep staging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5903–5915, Sep. 2022.

[7] J. B. Stephansen et al., "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature Commun.*, vol. 9, no. 1, pp. 5229(1)–5229(15), Dec. 2018.

[8] M. M. Rahman, M. I. H. Bhuiyan, and A. R. Hassan, "Sleep stage classification using single-channel EOG," *Comput. Biol. Med.*, vol. 102, pp. 211–220, Nov. 2018.

[9] C. Li, Y. Qi, X. Ding, J. Zhao, T. Sang, and M. Lee, "A deep learning method approach for sleep stage classification with EEG spectrogram," *Int. J. Environ. Res. Public Health*, vol. 19, no. 10, pp. 6322(1)–6322(17), May 2022.

[10] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "SeqSleepNet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 3, pp. 400–410, Mar. 2019.

[11] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 4, pp. 758–769, Apr. 2018.

[12] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 42, pp. 107–114, Apr. 2018.

[13] S. Mousavi, F. Afghah, and U. R. Acharya, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS ONE*, vol. 14, no. 5, pp. e0216456(1)–e0216456(11), May 2019.

[14] C. Sun, C. Chen, W. Li, J. Fan, and W. Chen, "A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 5, pp. 1351–1366, May 2020.

[15] M. Perslev, M. H. Jensen, S. Darkner, P. J. Jennum, and C. Igel, "U-Time: A fully convolutional network for time series segmentation applied to sleep staging," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2019, pp. 4415–4426.

[16] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. D. Vos, "Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Honolulu, HI, USA, Jul. 2018, pp. 1452–1455.

[17] E. Eldele et al., "An attention-based deep learning approach for sleep stage classification with single-channel EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 809–818, 2021.

[18] H. Wang et al., "A novel sleep staging network based on multi-scale dual attention," *Biomed. Signal Process. Control*, vol. 74, pp. 103486(1)–103486(10), Apr. 2022.

[19] N. Decat et al., "Beyond traditional sleep scoring: Massive feature extraction and data-driven clustering of sleep time series," *Sleep Med.*, vol. 98, pp. 39–52, Oct. 2022.

[20] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 324–333, Feb. 2018.

[21] H. Seo, S. Back, S. Lee, D. Park, T. Kim, and K. Lee, "Intra- and inter-epoch temporal context network (IITNet) using sub-epoch features for automatic sleep scoring on raw single-channel EEG," *Biomed. Signal Process. Control*, vol. 61, pp. 102037(1)–102037(11), Aug. 2020.

[22] H. Phan, K. Mikkelsen, O. Y. Chén, P. Koch, A. Mertins, and M. De Vos, "SleepTransformer: Automatic sleep staging with interpretability and uncertainty quantification," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 8, pp. 2456–2467, Aug. 2022.

[23] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2017, pp. 1–6.

[24] C. Sun, J. Fan, C. Chen, W. Li, and W. Chen, "A two-stage neural network for sleep stage classification based on feature learning, sequence learning, and data augmentation," *IEEE Access*, vol. 7, pp. 109386–109397, 2019.

[25] D. Zhou et al., "Alleviating class imbalance problem in automatic sleep stage classification," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 4006612(1)–4006612(12), 2022.

[26] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.

[27] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[28] R. B. Berry 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.*, vol. 8, no. 5, pp. 597–619, Oct. 2012.

[29] K. Sun et al., "High-resolution representations for labeling pixels and regions," 2019, *arXiv:1904.04514*.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[31] Z. Jia, Y. Lin, J. Wang, X. Wang, P. Xie, and Y. Zhang, "SalientSleepNet: Multimodal salient wave detection network for sleep staging," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2614–2620.

[32] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–18.

[33] A. Waswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 2961–2970.

[34] S. Seifpour, H. Niknazar, M. Mikaeili, and A. M. Nasrabadi, "A new automatic sleep staging system based on statistical behavior of local extrema using single channel EEG signal," *Expert Syst. Appl.*, vol. 104, pp. 277–293, Aug. 2018.

[35] B. Jiang, C. Ding, B. Luo, and J. Tang, "Graph-Laplacian PCA: Closed-form solution and robustness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3492–3498.

[36] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 11, pp. 1998–2008, Nov. 2017.

[37] A. Supratak and Y. Guo, "TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel EEG," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Montreal, QC, Canada, Jul. 2020, pp. 641–644.

[38] J. Phyo, W. Ko, E. Jeon, and H.-I. Suk, "TransSleep: Transitioning-aware attention-based deep neural network for sleep staging," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4500–4510, Jul. 2023.

[39] G. Fu, Y. Zhou, P. Gong, P. Wang, W. Shao, and D. Zhang, "A temporal–spectral fused and attention-based deep model for automatic sleep staging," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 1008–1018, 2023.

[40] G.-Q. Zhang et al., "The national sleep research resource: Towards a sleep data commons," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1351–1358, Oct. 2018.

[41] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–13.

[42] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, May 2019.

[43] Y. Sun, B. Wang, J. Jin, and X. Wang, "Deep convolutional network method for automatic sleep stage classification based on neurophysiological signals," in *Proc. Int. Congr. Image Signal Process., Biomed. Eng. Informat.*, Beijing, China: IEEE, Oct. 2018, pp. 1–5.

[44] N. Goshtasbi, R. Boostani, and S. Sanei, "SleepFCN: A fully convolutional deep learning framework for sleep stage classification using single-channel electroencephalograms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2088–2096, 2022.

[45] B. H. D. Koh, C. L. P. Lim, H. Rahimi, W. L. Woo, and B. Gao, "Deep temporal convolution network for time series classification," *Sensors*, vol. 21, no. 2, pp. 603(1)–603(20), Jan. 2021.

[46] G. Zhang, C. Wei, C. Jing, and Y. Wang, "Short-term electrical load forecasting based on time augmented transformer," *Int. J. Comput. Intell. Syst.*, vol. 15, no. 1, pp. 67(1)–67(11), Aug. 2022.

[47] F. Husain, B. Dellen, and C. Torras, "Action recognition based on efficient deep feature learning in the spatio-temporal domain," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 984–991, Jul. 2016.