LKSleepNet: An explainable lightweight model for sleep staging based on large kernel convolution

Zhe Liu
School of Artificial Intelligence and Computer Science
Jiangnan University
Wuxi, China
6233112031@stu.jiangnan.edu.cn

Jinlong Yang*
School of Artificial Intelligence and Computer Science
Jiangnan University
Wuxi, China
jlyang@jiangnan.edu.cn

***Abstract*—Automatic sleep staging is essential for diagnosing and treatment of sleep disorders. However, most convolutional based methods rely on empirically designed architectures and overlook the intrinsic time-frequency structure of EEG signals, resulting in elevated computational complexity, limited generalization and insufficient clinical interpretability that restrict both edge deployment and clinical adoption. Addressing these challenges, we propose an explainable and lightweight network LKSleepNet that integrates physiological priors into its architecture. The Fourier Transform Convolution (FTConv) and Large Kernel Convolution (LKConv) jointly capture global time-frequency patterns by combining transfer-domain filtering with large receptive-field modeling. The lightweight 2D temporal folding module (LKTimes) provides efficient contextual representation by folding temporal information into a compact 2D feature map. The Prior-Guided Spectral Attention (PGSA) module actively extracts stage-discriminative spectral patterns by generating learnable attention masks. The spectral constraint loss regularizes feature learning to ensure physiological consistency with AASM guidelines, effectively supervising the model to focus on valid frequency bands. Furthermore, we employ Grad-CAM and Conditional Random Fields (CRF) to visualize backbone features, revealing the contributions of different frequency bands to each sleep stage. Experiments on three public datasets SleepEDF-20, SleepEDF-78 and SHHS demonstrates that LKSleepNet outperforms state-of-the-art methods, achieving macro F1-scores of 79.4%, 76.4%, and 79.6%, and overall accuracies of 85.8%, 83.2%, and 86.3%, respectively. Experimental results demonstrate the capable of LKSleepNet for robust and interpretable sleep staging on edge devices.**

Keywords—Automatic sleep staging, Single-channel EEG, Large-kernel convolution network, Lightweight module, Explainable artificial intelligence

I. INTRODUCTION

Sleep occupies approximately one-third of human life [1], and disturbances or deprivation are closely associated with cardiometabolic risks such as hypertension, obesity, and cardiovascular diseases [2]-[3]. Accurate monitoring and characterization of sleep macrostructure are essential for advancing neuroscientific research and improving the diagnosis and treatment of sleep disorders. In clinical practice, sleep assessment primarily relies on polysomnography (PSG), which records multimodal physiological signals including electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG) and electrocardiogram (ECG) [4]. According to established guidelines, PSG recordings are manually segmented into sleep stages following the American Academy of Sleep Medicine (AASM) manual [5]. However, manual scoring is labor-intensive and time-consuming, underscoring the need for reliable automated sleep staging.

Deep learning has emerged as a powerful tool for automatic sleep staging. Convolutional neural networks (CNNs) serving as the backbone for extracting time-invariant features from EEG signals. Inspired by the Transformer-based architectures, recent studies have sought to enhance convolutional models by incorporating global contextual modeling and expanding the effective receptive field (ERF) [6]-[8]. Despite these advances, many existing approaches continue to emphasize increasingly complex architectures for better performance. Meanwhile, lightweight methods such as TinySleepNet [9], AttnSleepNet [10], SAN [11] and LWSleepNet [12] offer promising computational efficiency but rely on time-domain feature extraction, leaving the rich spectral structure of EEG signals insufficiently exploited.

Frequency-domain representations contain discriminative information for sleep staging. Prior studies have utilized short-time Fourier transform (STFT), fast Fourier transform (FFT) and related techniques to construct time–frequency features, achieving competitive performance using only spectral inputs [13]-[17]. However, standard convolutional kernels are inherently optimized for local spatial patterns and are not well aligned with the structured nature of frequency spectra, limiting their ability to extract fine-grained spectral features. This limitation indicates that frequency-domain information is essential for accurate sleep staging.

To capture both time-invariant and temporal dependencies, hybrid CNN-RNN architectures have also been explored [18]-[21]. Within these hybrid models, the CNN component is primarily responsible for learning time-invariant patterns from the raw signal. To capture long-term contextual dependencies, convolution-based models commonly expand the effective receptive field (ERF) by stacking multiple layers. However, this process tends to generate highly abstract feature representations that obscure stage-relevant patterns.

In this work, we propose the large-kernel convolutional sleep staging model LKSleepNet to address above challenges. This framework is designed for efficient and explainable sleep staging. The FTConv module performs spectral analysis to extract band-specific features directly. The large-kernel backbone captures discriminative waveform features. The temporal modeling module LKTimes folds linear features into 2D feature maps to extract inter-period and intra-period dependencies. The embedded PGSA layer actively highlights stage-relevant spectral patterns. Furthermore, we propose a spectral constraint loss to promote consistency with AASM physiological priors during training. Validations on Sleep-EDF-20, Sleep-EDF-78 and SHHS datasets show that LKSleepNet outperforms state-of-the-art methods.

The contributions are summarized as follows.

- The proposed LKSleepNet is an explainable and lightweight model that employs modern large kernel convolution module for the automatic sleep staging task.
- We propose a Prior-Guided Spectral Attention (PGSA) layer and a Spectral Constraint loss, which guide the model to align its attention with AASM spectral priors.
- We propose a Fourier-transform-based convolutional module that partitions the EEG signal into distinct frequency bands and extracts band-specific features directly in the frequency domain.
- We present a method that transforms 1D contextual features into a 2D shape by incorporating frequency domain information. Additionally, we introduce 2D large kernel convolution block to extract both inter- and intra-period features.
- We explain the performance of our model by the CRF and Grad-CAM method to reveal correlations between the large kernel convolution feature and the discriminative waveform pattern at each stage.

The rest of this article is organized as follows: Section II describes the details of proposed model architecture. Section III presents the experiments and feature explanations. The conclusions are given in Section IV. The repository of our work is: https://github.com/LiNiaaaa/LKSleepNet.git.

II. Proposed Methods

In this section, we introduce the proposed LKSleepNet for sleep staging based on single-channel EEG signals. Our model classifies EEG samples into sleep stages, one input epoch $\mathrm{X} = \{\mathrm{x}_0, \mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{N}\}$ contains N signal samples, where $x_i \in \mathbb{R}^{E_s \times F_s}$ denotes each sample has sampling rate of F_s and lasting E_s seconds. The corresponding ground-truth label that model determines is expressed as $y \in \{0,1\}^{N_c}$, where $y^{(N)}$ denotes the one-hot encoded label corresponding to the five sleep stages, N_c denotes the number of classes.

The overall architecture of LKSleepNet is shown in Fig. 1. It consists of four modules: FTConv, LKConv, the SE-Embed module and the LKTimes module. The FTConv module extracts frequency-domain features guided by sleep-related prior knowledge of EEG signal. The LKConv backbone extracts discriminative waveform patterns from these band specific features. The SE Embed module down samples the feature tensor and enhances the channel wise representations. The LKTimes module folds the 1D features into a 2D tensor based on sleep related rhythmic periodicity, allowing a large kernel convolution to capture spatiotemporal dependencies. The resulting representations are then passed to the classification block to produce the final predictions. The following subsections describe each module in detail.

A. Fourier transform convolution

The proposed FTConv module aims to preprocess the EEG signal in the frequency domain. Guided by the physiological prior from AASM manual. We conduct convolution operations and frequency-band partitioning directly in the spectral space to obtain efficient and clinically meaningful features. The structure of FTConv is illustrated in Fig. 2, and its overall workflow is summarized as follows.

- *Frequency patch*: Let $\Omega = \{\delta, \theta, \alpha, \sigma, \beta\}$ denote the set of physiological frequency bands. We first transform the input epoch X_s into its frequency representation $X_{\mathcal{F}}$ via FFT, then partition it into band-specific components $X_{\mathcal{F}}^{(b)}$ for each $b \in \Omega$ using a spectral mask as (1):

$$X_{\mathcal{F}}^{(b)}(\omega) = \begin{cases} X_{\mathcal{F}}(\omega), & \omega \in [\omega_{start}^{(b)}, \omega_{end}^{(b)}] \\ 0, & otherwise \end{cases} \tag{1}$$

  we set the bandwidth to 4Hz for δ through σ bands, and 16Hz for the β band, effectively filtering out frequencies above 50Hz.

- *Convolution calculation process*: The frequency representation of the convolution kernel is also divided into five groups according to the frequency patch method. we perform convolution via spectral multiplication. The intermediate spectral feature $Z^{(b)}$ is obtained by the Hadamard product of the masked signal $X_{\mathrm{F}}^{(b)}$ and the kernel $W_{\mathrm{F}}^{(b)}$. This ensures that features are extracted strictly from the target physiological bands. The process is described in (2):

$$Z^{(b)} = X_{\mathrm{F}}^{(b)} \odot W_{\mathrm{F}}^{(b)} \tag{2}$$

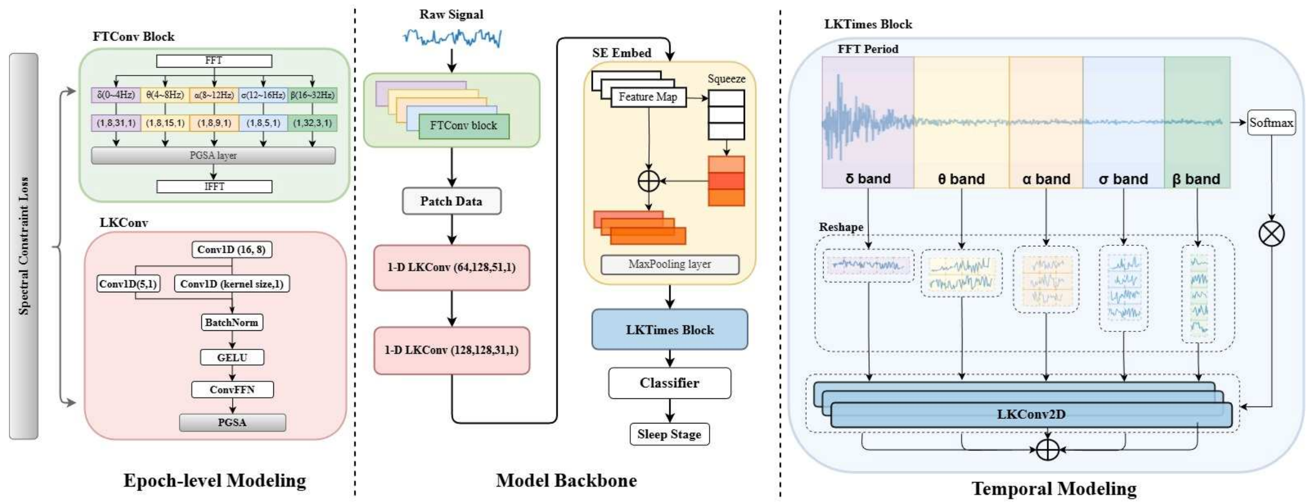


Fig. 1. The overall structure of the LKSleepNet.

- *Prior-Guided Spectral Attention*: The PGSA module is designed to suppress residual intra-band artifacts. It generates a learnable mask M via Sigmoid function σ to refine the intermediate feature $Z^{(b)}$ as:

$$M = \sigma\left(\mathcal{F}_{\text{Conv}}\left(Z^{(b)}\right)\right) \tag{3}$$

where the refined spectral feature is obtained by element-wise multiplication. Finally, the enhanced representations are transformed back to the time domain via Inverse FFT and concatenated to obtain the final output X_{FTConv} as (4), where $\mathcal{F}^{-1}$ denotes the IFFT process.

$$X_{FTConv} = \text{Concat}_{b\in\Omega}\left(\mathcal{F}^{-1}\left(M \odot Z^{(b)}\right)\right) \tag{4}$$

The output of the FTConv is serving as a preprocessing step of the EEG signal.

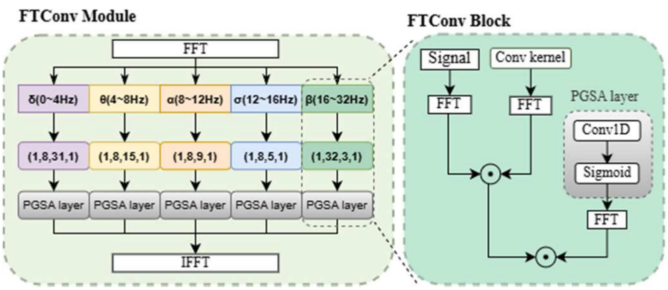


Fig. 2. The structure of the FTConv block.

B. Large kernel convolution

The LKConv backbone is designed to transform the frequency-domain representations from the FTConv module into high-level discriminative feature vectors. To achieve a large Effective Receptive Field (ERF) essential for capturing global EEG patterns without excessive network depth, we employ a depth-wise separable convolution architecture [22], which is widely adopted in lightweight models for its computational efficiency [23]-[27].

As illustrated in Fig. 3, the proposed backbone consists of two stacked large-kernel blocks. Each block comprises a large-kernel Depthwise Convolution (DWConv) for spatial feature extraction, followed by a Convolutional Feed-Forward Network (ConvFFN) for channel mixing and feature refinement. Batch Normalization [28] and GELU activation are applied within each block.

Specifically, the large kernel size of the first block is set to 51 (with 64 filters) to capture long-range dependencies, the second block employs a kernel size of 31 to extract finer-grained features, enabling multi-scale representation extracting. The max-pooling layer is inserted between blocks to reduce the feature dimension. Within the ConvFFN, we adopt a symmetric inverted bottleneck structure with an expansion ratio of 4, a Squeeze-and-Excitation (SE) module is integrated to adaptively recalibrate channel-wise feature importance and dropout is used after each ConvFFN block to prevent overfitting. Finally, we deploy a PGSA layer at the end of the backbone for further suppress redundancy in the extracted waveform features. It generates a temporal attention mask to refine the epoch-level representations before they are fed into the subsequent temporal modeling module.

C. Squeeze and excitation feature embedding

To improve the expressive of the LKConv features and reducing the size of the feature tensor, the residual squeeze-and-excitation (SE) [29] block is implemented after the LKConv block. First, put large kernel convolution feature X_{LKConv} into a convolution block that consists of a Conv1d, a BN and a ReLU layer, then implement an SE block to select features of each channel that most distinctive for the classification, the channel reduction rate is set to 16. After the two steps above, the feature dimension of each channel is down-sampled by the global maximum pooling module to obtain the final output of the embedding layer. The entire feature embedding process is presented by (5), where $MaxPool$ denotes the global maximum pooling operation.

$$X_E = MaxPool\left(SE\left(X_{LKConv}\right)\right) \tag{5}$$

Finally, the embedded feature vector X_E represents each epoch of the EEG signal, serve as the input to the contexture modeling module for temporal feature extraction.

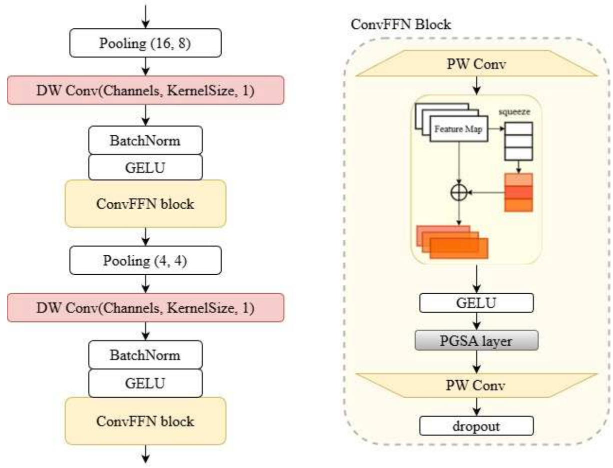


Fig. 3. The overall structure of the LKConv block.

D. Temporal information modeling

Temporal features are essential for accurate sleep staging. Inspired by the TimesNet [30]. We employ a robust approach LKTimes to efficiently extract both intra- and inter-period variations by converting 1D time series features into 2D tensors.

The extraction process relies on spectral frequency analysis. We first apply the FFT to get the frequency of the embedded feature X_E and calculate the magnitude to obtain the amplitude spectrum A as (6).

$$A = Amp\left(\mathcal{F}(X_E)\right). \tag{6}$$

The derived spectrum A is utilized to identify dominant periodicities across five sleep-related frequency bands (e.g., $\delta, \theta, \alpha, \sigma, \beta$). For each band, the frequency index i_b^* with the maximum amplitude is selected to compute the corresponding period length P_b by dividing it into the total feature length L. The selection and calculation process are illustrated in (7).

$$P_b = \left\lfloor \frac{L}{i_b^*} \right\rfloor + \beta_b,\ i_b^* = \arg\max_{i \in I_b} A\,(i),\ b \in \Omega \tag{7}$$

Based on the determined period, the input feature is padded with zeros along the temporal dimension, then fold it into a 2D spatial tensor $\tilde{x}^{(b)}$ to fully represent temporal variations. A 2D large kernel convolutional network processes these maps to extract intra-period and inter-period information. As shown in Fig. 4, the network begins with a small-kernel convolution block for short-term feature extraction. A spatially separated LKConv block follows with two parallel convolutional layers. These layers utilize kernel sizes of (1, 21) and (21, 1). A ConvFFN block with a ratio of 4 completes the extraction. The output feature map $X_t^{(b)}$ is generated as defined in (8).

$$X_t^{(b)} = \mathcal{F}_{\text{Conv}}\left(\tilde{x}^{(b)}\right), \quad \tilde{x}^{(b)} \in \mathbb{R}^{B \times \frac{L_b}{P_b} \times P_b}. \tag{8}$$

The feature maps generated for each frequency band are adaptively aggregated by the importance of each frequency band. The amplitude of each band is used to estimate each feature importance. The maximum amplitude value w_b is normalized through a softmax operation to obtain the weight α_b. The process is summarized as (9), where the $X_{\mathcal{T}}$ denotes the 1D feature generated by the LKTimes module.

$$X_{\mathcal{T}} = \sum_{b=1}^{5} \alpha_b\ \mathcal{F}_{\text{Flat}}\left(X_t^{(b)}\right),\ \alpha_b = \frac{\exp(w_b)}{\sum_{k=1}^{5}\exp(w_k)}. \tag{9}$$

Finally, the highly structured 1D tensor $X_{\mathcal{T}}$ provides the discriminative feature used for classification.

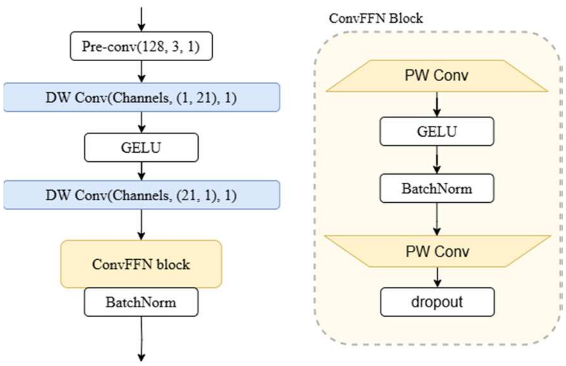


Fig. 4. The overall structure of the 2D LKConv block.

E. Spectral Constraint Loss

To guide the masks towards valid sleep frequencies, we propose a loss function based on the AASM manual. We explicitly constrain the Power Spectral Density (PSD) of the backbone features X_{LKConv} during training. The PSD is computed as $S(\omega) = |\,\mathcal{F}(X_{LKConv})\,|^2$. We define target frequency bands Ω_{y_i}corresponding to the ground-truth sleep stage y_i. The spectral constraint loss L_{spec} penalizes energy distribution outside these physiological bands as (10).

$$L_{\text{spec}} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\sum_{f\in\Omega_{y_i}} S_i(f)}{\sum_{f\in\Omega} S_i(f)+\epsilon}\right) \tag{10}$$

The total loss is $L_{total} = L_{ce} + \lambda L_{spec}$, where L_{ce} is the cross-entropy loss, the λ of L_{spec} is set to 1.35.

III. Experiments

A. Dataset and Evaluation metrics

We evaluate our work on three public datasets, namely Sleep-EDF-20, Sleep-EDF-78 and SHHS. For each dataset, we only use single EEG channel to train and test our model.

SleepEDF-20: Obtained from PhysioBank, this dataset includes 39 PSG recordings from 20 healthy subjects aged 25 to 101. Except for subject 13 who has only one night of data, all subjects contribute two nights of recordings. Each recording is segmented into 30 second epochs. For this study we use only the Fpz-Cz EEG channel, sampled at 100 Hz.

SleepEDF-78: This dataset is an expansion of Sleep-EDF-20, containing 153 recordings from 78 subjects. Most participants have two-night records, with the exception of subjects 13, 36, and 52. The preprocessing steps and channel selection are the same as in SleeEDF-20.

SHHS: The SHHS [31] is a large-scale cohort study investigating the cardiovascular consequences of sleep-disordered breathing. Following the protocol established in [10], we select a subset of 329 subjects from the original 6,441. For our experiments, we utilize the C4-A1 EEG channel, sampled at 125 Hz.

Sleep stages are manually split into the following categories: W, N1, N2, N3, N4, REM, MOVEMENT, and UNKNOWN. We remove MOVEMENT and UNKNOWN stages, merge “N4” and “N3” into “N3” following the AASM manual. According to the researches before, we only use 30 minutes before and after sleep of each record for prediction.

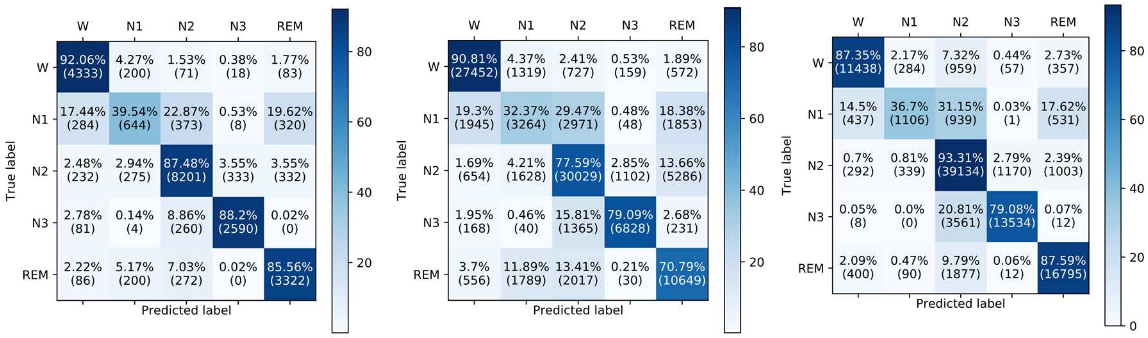
Our metrics are adopted to evaluate the performance of various models for sleep stage classification, namely, the accuracy (Acc), macro-averaged F1-score (MF1), and the Cohen Kappa (κ) [32]. The overall Acc, k and $MF1$ are defined as follows

$$Acc = \frac{\sum TP_i}{M} \tag{11}$$

$$k = \frac{P_o - P_e}{1 - P_e} \tag{12}$$

$$MF1 = \frac{1}{M}\sum_{i=1}^{M} \frac{\sum Precision_i \times Recall_i}{Precision_i + Recall_j} \tag{13}$$

where $Precision_i = \frac{TP}{TP_i + FP_i}$, $Recall_i = \frac{TP}{TP_i + FN_i}$, M is the total number of samples. We also use per-class precision (PR), per-class recall (RE) and per-class F1-score (F1) to evaluate our model. They are calculated as in binary classification by considering one class as the positive class and the other four classes as the negative class.



(a) (b) (c)

Fig. 5. (a) Confusion matrix with sleep-edf-20. (b) Confusion matrix with sleep-edf-78. (c) Confusion matrix with SHHS.

B. Performance of LKSleepNet

Figs. 5 (a)-(c) illustrate the confusion matrices for proposed LKSleepNet, applied to the Fpz-Cz channel of the SleepEDF datasets and the C4-A1 channel of the SHHS dataset. For SleepEDF datasets, the confusion matrix is generated by aggregating all staging results from each fold. Each row of the figure represents the expert-labeled samples, while each column represents the epochs predicted by the model. Notably, stage N1 shows the lowest F1-score and is commonly misclassified as W or N2. In contrast, stage N3 demonstrates the best performance for the SleepEDF-20 dataset. However, its performance declines on the SleepEDF-78 dataset. Most misclassifications across the datasets are concentrated in stage N2, the most prevalent class.

The results presented above further indicate that the majority of low-confidence predictions correspond to the N1 stage, which is frequently misclassified as the W, REM and N2 stages. The difficulty in accurately recognizing the N1 stage can be attributed to its infrequent occurrence and its strong resemblance to the W and N2 stages.

C. Baselines and Experimental Setup

DeepSleepNet [20] exploits a specific tow-branch CNN head with two LSTM layers to classify the sleep stages.

TinySleepNet [11] aims to propose a lightweight model, it use one-branch CNN to extract epoch-level information and combined with a LSTM layer to modeling the contexture feature and classify stages on single EEG channel.

ResnetLSTM [23] uses ResNet to extract features from EEG signals, then classified into sleep stages by the LSTM.

FC-STGCN [32] is a graph neural network which models the spatial-temporal dependencies between the channels. It can be used with multi or single channel inputs.

AttnSleepNet [12] serves as an end-to-end hybrid model that combines a CNN head with an attention mechanism to classify a single EEG channel. For a fair comparison of performance, we broaden the feature vector of the AFR module to contain more temporal information.

SsleepNet [33] integrates a multi-scale CNN feature extractor with a Bi-LSTM–CRF structured module for sleep staging.

We adopt cross-validation to evaluate the performance of various models. Subjects in SleepEDF-20 are divided into 20 groups, which corresponds to a leave-one-subject-out (LOSO) cross-validation. Subjects in SleepEDF-78 are divided into 10 groups. The SHHS dataset is split into training and test sets with 70% for training and 30% for testing.

The proposed model is built on the PyTorch version 3.7 and the experiments are conducted on an RTX2080ti. The batch size is set to 64 and the length of sequence is set to 10. We train 200 epochs by the Adam optimizer, the learning rate starts with 1e-4, and the weight decay is set to 1e-3. We apply dropout with a rate of 0.2 to prevent overfitting.

D. Comparison With State-of-the-Art Approaches

We evaluate LKSleepNet against several state-of-the-art models in terms of overall accuracy, macro F1-score, Cohen's kappa and parameter count across three datasets. As shown in Table I, LKSleepNet achieves higher overall accuracy and macro F1-score on SleepEDF-20, SleepEDF-78 and SHHS, while requiring only 0.47 M parameters. The model performs better across all stages except for a slightly lower F1 score on N1. Notably, it also delivers superior performance on the REM stage compared with other baselines.

Superior efficiency in both training and inference is another advantage of the proposed method. DeepSleepNet [20], ResnetLSTM [23] and TinySleepNet [11] rely on LSTMs for temporal modeling, which introduce large parameter overhead and slow recurrent computation. In contrast, LKSleepNet extracts temporal dependencies using convolutional operations, enabling parallel processing and more efficient optimization.

Table II. illustrates the experiment evaluates the efficiency of the proposed convolutional feature extraction against state-of-the-art multi-scale convolution heads. Validations are conducted on the SleepEDF-20 dataset using 20-fold cross-validation. Architectures such as SleepPyCo [19] requires extensive parameters to achieve better performance. DeepSleepNet and MSDAN with over 0.5 million parameters without delivering commensurate accuracy. Optimized models such as AttnSleepNet balance performance and size. The proposed LKConv outperforms these baselines with a accuracy of 85.8%. It accomplishes this with the lowest parameter count of 0.26 million. This result confirms the efficacy of the large-kernel design in capturing discriminative features efficiently.

TABLE II. COMPARISON WITH STATE-OF-THE-ART METHODS

Dataset	Sleep-edf-20	Class-wise F1 Score					Overall metrix			Parameters ($\times 10^6$)
		W	N1	N2	N3	REM	Acc	MF1	k	
Sleep-edf-20	FC-STGCN [32]	87.9	33.5	87.5	85.8	80.3	80.8	75.0	0.72	0.49
	DeepSleepNet [20]	86.7	45.5	85.1	83.3	82.6	81.9	76.6	0.76	24.03
	ResnetLSTM [23]	86.5	28.4	87.7	89.8	76.2	82.5	73.7	0.76	1.71
	SsleepNet [33]	89.6	**50.0**	87.6	87.3	**82.8**	84.5	79.1	0.75	0.64
	TinySleepNet [11]	85.7	47.8	87.2	83.9	81.8	84.8	77.3	0.76	1.52
	AttnSleepNet [10]	89.5	42.6	89.0	90.1	79.5	85.1	78.3	0.80	0.55
	LKSleepNet	**89.7**	41.3	**89.1**	**90.3**	82.8	**85.8**	**79.4**	**0.82**	**0.47**
Sleep-edf-78	FC-STGCN [32]	90.9	39.7	83.2	76.6	73.5	77.8	71.8	0.70	0.49
	DeepSleepNet [20]	91.1	45.3	79.4	73.0	71.4	78.1	72.0	0.71	24.03
	ResnetLSTM [23]	91.0	34.9	83.9	81.2	69.1	79.2	71.7	0.72	1.71
	SsleepNet [33]	90.8	44.3	83.7	82.3	77.3	82.3	73.1	0.72	0.64
	TinySleepNet [11]	91.9	**47.7**	84.2	76.9	78.8	82.8	75.3	0.75	1.52
	AttnSleepNet [10]	**92.0**	42.4	85.7	82.4	74.5	81.8	75.3	0.76	0.55
	LKSleepNet	**92.4**	41.7	**85.5**	82.3	78.6	**83.2**	**76.4**	**0.79**	**0.47**
SHHS	ResnetLSTM [23]	86.2	29.3	87.4	87.0	82.1	84.4	70.2	0.74	1.71
	DeepSleepNet [20]	86.4	42.7	83.4	80.3	82.4	81.9	75.2	0.74	24.03
	TinySleepNet [11]	88.4	**48.9**	86.7	86.3	85.7	85.2	77.3	0.76	1.52
	AttnSleepNet [10]	87.9	39.3	88.1	**87.8**	84.2	85.6	78.8	0.80	0.55
	SsleepNet [33]	88.3	46.7	88.5	86.2	83.5	86.0	79.0	0.81	0.64
	LKSleepNet	**89.1**	45.8	**88.5**	85.9	**88.3**	**86.3**	**79.6**	**0.82**	**0.47**

TABLE I. COMPARISON WI OTHER CONVOLUTIONAL BASED MODELS.

CNN Head	Overall performance			Parameters ($\times 10^6$)
	Acc	MF1	k	
SleepPyCo [19]	85.7	**80.4**	**0.83**	1.24
DeepSleepNet [20]	80.8	75.0	0.77	0.76
SsleepNet [33]	84.5	79.1	0.75	0.64
TinySleepNet [11]	84.8	77.3	0.76	0.33
AttnSleepNet [12]	85.1	78.3	0.80	0.40
LKConv	**85.8**	79.4	0.82	**0.26**

E. Grad-CAM for 1D large kernel convolution

Grad-CAM is an enhanced version of Class Activation Mapping (CAM) that uses the gradient information flowing into target convolutional layer to understand the importance of each input in the decision-making process of the network [40]. It calculates the gradients of the target class y^c based on the chosen convolutional layer feature map. Then average these gradients to get the weights α, which capture the importance of each feature map k for the target class c:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}(\textstyle\sum_k \alpha^c_k A^k), \alpha^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \quad (14)$$

The calculated heatmap $L^c_{\text{Grad-CAM}}$ highlights the convolutional feature areas that the model focuses on. We normalize and resize it to facilitate a comprehensive joint visualization with the raw EEG signal.

Fig. 6 illustrates the identified EEG signal patterns based on LKConv layer features using the Grad-CAM method. The deep-colored areas in the heatmap indicate higher attention weights calculated from the convolution feature map. The heatmap analysis mainly yields two findings. First, the model shows a stronger response to the alpha rhythm during wake stage, which differs from the AASM manual emphasis on beta activity. Second, the heatmap clearly represents other critical waveforms. These waveforms include spindles, K-complexes within slow oscillations, and vertex waves from the N1 stage [6]. Furthermore, spindles following sharp EEG deflections serve as strong indicators of N2 staging, as highlighted in the class-discriminative regions. Lastly, the model can focus on short-duration, high-frequency EEG bursts related to scalp muscle activity, which provide evidence for the REM stage. The described performance aligns with the AASM manual, and the heatmap-highlighted EEG patterns strongly support staging results.

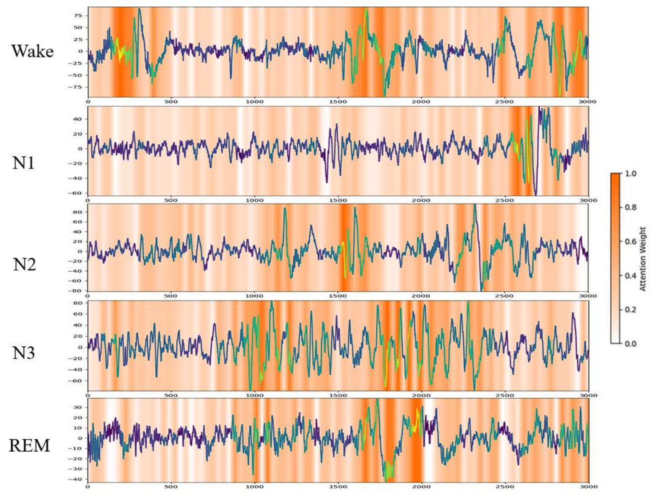


Fig 6. Use Grad-CAM to visualize the last LKConv layer features among five classes (SleepEDF-20, subject 16).

F. Visualize waveform features of convolutional modules

Fig. 7 illustrates the features extracted by the FTConv module across five channel groups. Each row in the figure corresponds to the features averaged from each frequency band channels in the FTConv module, while each column represents different stage samples. In the delta band channel, module responds to the large-scale waveforms as shown in W and N3 stage, the alpha band channel responds to the N2 stage sleep spindle, and the sigma band channel focuses on high-frequency information that relative with N1 and REM stages. The above results indicate that the proposed FTConv is capable of capture multi-scale EEG waveforms through its convolutional channels, supporting further feature extraction.

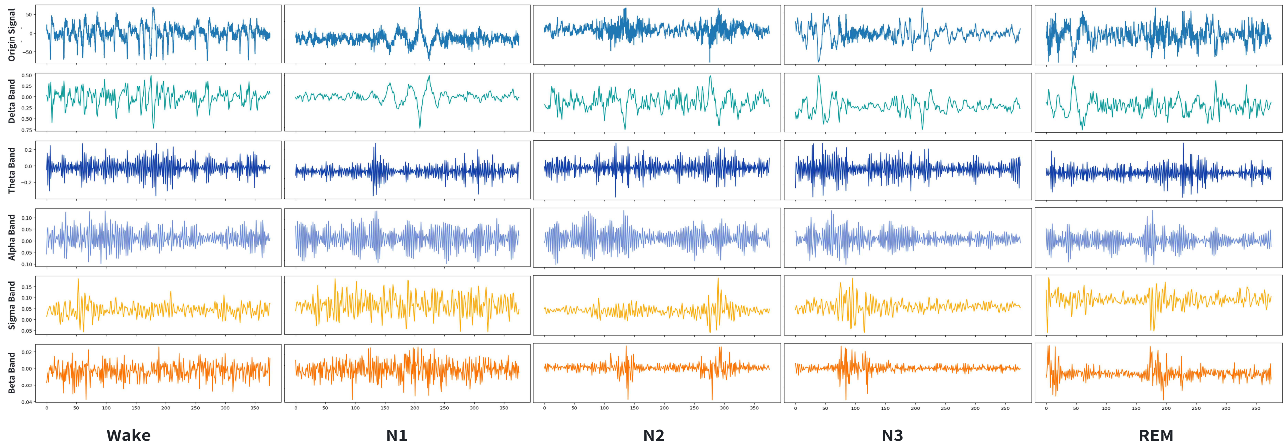


Fig. 7. Visualize the five band channel features of FTConv among classes.

Fig. 8 illustrates a feature snapshot extracted by the LKConv module on the SleepEDF-20 dataset. W stage waveform shows higher amplitudes. N1 stage waveforms resemble W stage waveforms but have lower amplitudes. N2 and N3 stage waveforms display closely matched temporal scales and amplitudes. REM stage features are dominated by high frequency sharp peaks. The described feature patterns align with the staging rules in the AASM manual.

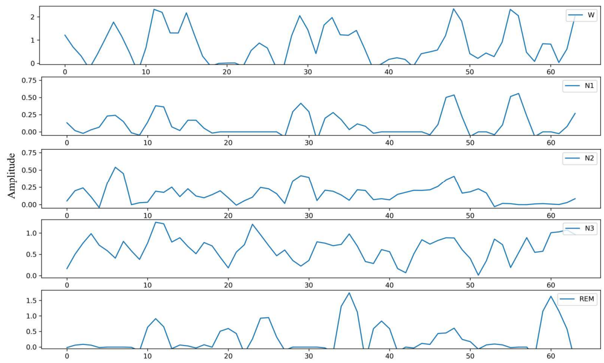


Fig. 8. Visualize the five band channel features of LKConv on the SleepEDF-20 dataset.

G. CRF analysis

We employ CRF post-processing to evaluate the ability of the model to capture sleep-stage transition patterns. As illustrated in Fig. 9, we present the transition probabilities of the classification results obtained from the CRF. The explanations of the figure are detailed below.

Firstly, each stage exhibits higher probability of self-transition. The most frequently corrected sleep transition rule is the correction from N1 to REM, followed by the correction from N1 to W. The facts align with the AASM manual, which is the N1 stage serves as the initial phase for transitioning to other stages. The probability of transitioning from the N3 stage to the N2 stage is not significantly different from the probability of remaining in the same stage, indicating that after entering a deep sleep state, these two stages tend to alternate back and forth. Furthermore, the probabilities of transitioning between N3 and N1, as well as between N3 and REM, are nearly zero. This can be explained by the different nature of these stages. N3 is a stable deep sleep stage, while N1 and REM are much lighter stages. Therefore, direct transitions from N3 to either N1 or REM are rare. The results indicate that the model effectively captures the transition patterns among sleep stages.

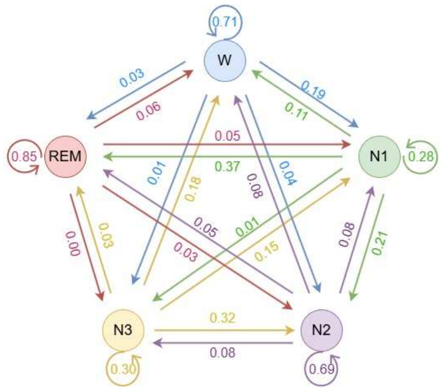


Fig. 9. Stage transition probabilities learned by the CRF.

H. Ablation study

The proposed model consists of four components: the FTConv module, the LKConv module, the SE-embed module and the LKTimes module. We conduct six sets of experiments on the SleepEDF-20 dataset. Table III shows the ablation results, each row corresponds to one ablation setting, referred to as Experiment 1-6.

TABLE III. THE IMPLEMENTATION OF THE ABLATION STUDY.

Epoch-level module		Temporal module		Spec Loss	Acc	Params ($\times 10^5$)
FTConv	LKConv	SE-Embed	LKTimes			
√					77.1	1.8
√	√				78.3	2.6
√	√	√			82.7	2.8
√	√		√		83.3	4.5
√	√	√	√		85.7	4.7
√	√	√	√	√	85.8	4.7

We can draw the following conclusions from ablation experiments. Experiments 1 and 2 demonstrate that FTConv can effectively extract epoch-level features from EEG signals. A comparison of the results from Experiment 2 and Experiment 3 indicates the SE-Embed layer is essential for addressing the temporal and spatial correlations among EEG features. Furthermore, the results from Experiments 3 and 4 suggest that using LKTimes captures more discriminative temporal features than the SE-embed layer. Additionally, the results of Experiments 4 and 5 reveal that deploying both modules significantly enhances the overall classification performance. Experiment 6 reaches the highest accuracy, indicating that proposed spectral constraint loss improves model generalization and performance.

IV. Conclusion And Future Work

We propose LKSleepNet, an explainable and lightweight sleep staging network for single channel EEG. The model comprises four components that jointly capture epoch level features and temporal dependencies. The FTConv module extracts stage related features from raw EEG, the LKConv backbone models intra epoch waveform relationships, the SE embed module emphasizes discriminative channels, and the LKTimes module learns stage transition dynamics.

Experimental results show that LKSleepNet effectively captures sleep stage transition rules. The Grad-CAM visualizations further reveal that the model focuses on regions consistent with AASM stage indicators, supporting the interpretability of the predictions. Overall, LKSleepNet achieves superior performance compared with other advanced methods.

References

[1] Institute of Medicine, Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem, Washington DC: The National Academies Press, 2006.

[2] Han H, Wang Y, Li T, et al. "Sleep duration and risks of incident cardiovascular disease and mortality among people with type 2 diabetes," *Diabetes Care.*, 2023, 46(1): 101-110.

[3] J. Hobson and E. Pace-Schott, "The cognitive neuroscience of sleep: Neuronal systems, consciousness and learning," *Nature Rev. Neurosci.*, vol. 3, no. 9, pp. 679–693, 2002.

[4] Keenan S A. An overview of polysomnography[J]. Handbook of clinical neurophysiology, 2005, 6: 33-50.

[5] Schulz H. Rethinking sleep analysis: Comment on the AASM manual for the scoring of sleep and associated events[J]. Journal of Clinical Sleep Medicine, 2008, 4(2): 99-103.

[6] Chen H, Chu X, Ren Y, et al. Pelk: Parameter-efficient large kernel convnets with peripheral convolution[C]. *Conference on computer vision and pattern recognition.*, pp. 5557-5567, 2024.

[7] Luo D, Wang X. Moderntcn: A modern pure convolution structure for general time series analysis[C], *The twelfth international conference on learning representations.*, pp. 1-43, 2024.

[8] Chen H, Chu X, Ren Y, et al. Pelk: Parameter-efficient large kernel convnets with peripheral convolution[C], *Conference on computer vision and pattern recognition.*, pp. 5557-5567, 2024.

[9] Supratak A, Guo Y. TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel EEG[C], *Annual International Conference of the IEEE Engineering in Medicine & Biology Society,*. pp. 641-644, 2020.

[10] Eldele E, Chen Z, Liu C, et al. An attention-based deep learning approach for sleep stage classification with single-channel EEG[J], *IEEE Transactions on Neural Systems and Rehabilitation Engineering.*, vol. 29, pp. 809-818, 2021.

[11] Zhou W, Zhu H, Shen N, et al. A lightweight segmented attention network for sleep staging by fusing local characteristics and adjacent information[J], *IEEE Transactions on Neural Systems and Rehabilitation Engineering.*, vol. 31, pp. 238-247, 2022.

[12] Yang C, Li B, Li Y, et al. LWSleepNet: A lightweight attention-based deep learning model for sleep staging with singlechannel EEG[J], *Digital Health.*, vol. 9, 2023.

[13] Phan H, Lorenzen K P, Heremans E, et al. L-SeqSleepNet: Whole-cycle long sequence modeling for automatic sleep staging[J], *IEEE Journal of Biomedical and Health Informatics.*, vol. 27, no. 10, pp. 4748-4757, 2023.

[14] Ali M H, Uddin M B. Detection of sleep arousal from STFT-based instantaneous features of single channel EEG signal[J], *Physiological Measurement*., vol. 45, no. 10, pp. 105005, 2024.

[15] Li C, Qi Y, Ding X, et al. A deep learning method approach for sleep stage classification with EEG spectrogram[J], *International journal of environmental research and public health.*, vol. 19, no. 10, pp. 6322, 2022.

[16] J. Pradeepkumar *et al*., Toward Interpretable Sleep Stage Classification Using Cross-Modal Transformers[J], *IEEE Transactions on Neural Systems and Rehabilitation Engineering.*, vol. 32, pp. 2893-2904, 2024.

[17] Chen Z, Yang Z, Zhu L, et al. Automated sleep staging via parallel frequency-cut attention[J], *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1974-1985, 2023.

[18] W. Zhang, S. Zhang, Y. Wang, C. Li, H. Peng and X. Chen, A CNN-Transformer-ConvLSTM-CRF Hybrid Network for Sleep Stage Classification[J], *IEEE Sensors Journal*, 2024, vol. 24, no. 18, pp. 29018-29029.

[19] Lee S, Yu Y, Back S, et al. Sleepyco: Automatic sleep scoring with feature pyramid and contrastive learning[J]. Expert Systems with Applications, 2024, 240: 122551.

[20] Supratak A, Dong H, Wu C, et al. DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG[J]. IEEE transactions on neural systems and rehabilitation engineering, 2017, 25(11): 1998-2008.

[21] Cao Y, Xiang W, Wei J, et al. CrossFusionSleepNet: A multimodal deep learning model for automatic sleep stage classification[J]. Biomedical Signal Processing and Control, 2026, 112: 108538.

[22] Koonce B. MobileNetV3[M] Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization[J], *Berkeley, CA: Apress.*, pp. 125-144, 2021.

[23] Y. Sun, B. Wang, J. Jin, and X. Wang. Deep convolutional network method for automatic sleep stage classification based on neurophysiological signals[C], *Proc. 11th Int. Congr. Image Signal Process.*, 2018, pp. 1–5.

[24] Wang F, Zheng Z, Hu B S, et al. EfficientSleepNet: a novel lightweight end-to-end model for automated sleep staging on single-channel EEG[C], ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025: 1-5.

[25] Fei K, Wang J, Pan L, et al. A sleep staging model on wavelet-based adaptive spectrogram reconstruction and light weight CNN[J], *Computers in Biology and Medicine*, vol. 173, pp. 108300, 2024.

[26] Wang Z, Yao J, Xu M, et al. Transformer-based network with temporal depthwise convolutions for sEMG recognition[J], *Pattern Recognition.*, vol. 145, pp. 109967, 2024.

[27] Vishwanath M, Cao S, Dutt N, et al. Reducing interdataset covariate shift in sleep EEG of traumatic brain injured humans and mice using Transfer Euclidean Alignment[J]. Biomedical Signal Processing and Control, 2025, 108: 107948.

[28] Lubana E S, Dick R, Tanaka H. Beyond batchnorm: Towards a unified understanding of normalization in deep learning[J], *Advances in Neural Information Processing Systems.*, vol. 34, pp. 4778-4791, 2021.

[29] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C], *Proceedings of the IEEE conference on computer vision and pattern recognition*., pp. 7132-7141, 2018.

[30] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, TimesNet: Temporal 2Dvariation modeling for general time series analysis[C], International Conference on Learning Representations, 2023.

[31] Goldberger A L, Amaral L A N, Glass L, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals[J], *circulation*., vol. 101, no. 23, pp. e215-e220, 2000.

[32] Y. Wang, Y. Xu, J. Yang, M. Wu, X. li, L. Xie, and Z. Chen, Fullyconnected spatial-temporal graph for multivariate time-series data[C], *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 15715–15724, 2024.

[33] Lv X, Ma J, Li J, et al. Ssleepnet: a structured sleep network for sleep staging based on sleep apnea severity[J]. Complex & Intelligent Systems, 2024, 10(2): 2689-2701.