# GraphSleepNet: Adaptive Spatial-Temporal Graph Convolutional Networks for Sleep Stage Classification

**Ziyu Jia**[1,2] , **Youfang Lin**[1,2,3] , **Jing Wang**[1,2,3*] , **Ronghao Zhou**[1] , **Xiaojun Ning**[1] , **Yuanlai He**[1] and **Yaoshuai Zhao**[3]

[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
[3]CAAC Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing, China
{ziyujia, yflin, wj, rhzhou, ningxj, yuanlaihe}@bjtu.edu.cn, yszhao@travelsky.com

## Abstract

Sleep stage classification is essential for sleep assessment and disease diagnosis. However, how to effectively utilize brain spatial features and transition information among sleep stages continues to be challenging. In particular, owing to the limited knowledge of the human brain, predefining a suitable spatial brain connection structure for sleep stage classification remains an open question. In this paper, we propose a novel deep graph neural network, named GraphSleepNet, for automatic sleep stage classification. The main advantage of the GraphSleepNet is to adaptively learn the intrinsic connection among different electroencephalogram (EEG) channels, represented by an adjacency matrix, thereby best serving the spatial-temporal graph convolution network (ST-GCN) for sleep stage classification. Meanwhile, the ST-GCN consists of graph convolutions for extracting spatial features and temporal convolutions for capturing the transition rules among sleep stages. Experiments on the Montreal Archive of Sleep Studies (MASS) dataset demonstrate that the GraphSleepNet outperforms the state-of-the-art baselines.

## 1 Introduction

Sleep stage classification is important for the assessment of sleep quality and the diagnosis of sleep disorders. Sleep experts identify sleep states based on sleep standard and observations recorded in polysomnography (PSG), which contains electroencephalography (EEG) at different positions on the head, electromyography (EMG), and electrooculography (EOG), etc. Currently, the two commonly used sleep standards are the Rechtschaffen and Kales (R&K) standard [Wolpert, 1969] and American Academy of Sleep Medicine (AASM) standard [Berry *et al.*, 2012]. The transition rules between different sleep stages, which can assist sleep experts in identifying the sleep stages, are also recorded in the AASM standard. Although these rules provide very valuable information, classifying the sleep stages by human sleep experts is still a tedious and time-consuming task. Moreover, the classification results are affected by the variability and subjectivity of sleep experts.

Automatic sleep stage classification can greatly improve the efficiency of traditional sleep stage classification and has important clinical value. A large number of researchers have made great contributions to automate this classification task. At first, traditional machine learning methods based on time domain, frequency domain, and time-frequency domain features are adopted. However, the classification accuracy of these methods depends heavily on feature engineering and feature selection, which require a lot of expert knowledge. Recently, deep learning methods have been widely applied to automatically classify sleep stage thanks to its powerful ability of representation learning. For example, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are often utilized to learn appropriate feature representations from transformed data or directly from raw data.

Although CNNs and RNNs can achieve high accuracy for sleep stage classification, their limitation is that the model's input must be grid data (themselves or after conversion to image-like representations) ignoring the connection among brain regions. Due to brain regions are in non-Euclidean space, graph is the most appropriate data structure to indicate brain connection. Motivated by the success of the graph convolution network (GCN) model in graph data, the sleep stage classification is studied by the graph representation method, where each EEG channel corresponds to a node of the graph, and the connection between the channels correspond to the edge of the graph. But an important input of GCN is the graph connection representation, which is generally a fixed graph structure which is set in advance. Although existing methods, such as the phase-locking value (PLV) method, can obtain a fixed graph structure, due to the limited understanding of the brain, it is still challenging to determine a suitable graph structure in advance for sleep stage classification.

Besides, how to exploit the sleep transition rules between neighboring stages is another challenge. Actually, sleep experts usually identify the class label of one sleep state according to both the characteristic EEG waves of the current state and the class labels of its neighbors. Hence, it is reasonable to take transition rules into account to improve classification accuracy.

[*]corresponding author: wj@bjtu.edu.cn

In order to address the above challenges, we propose GraphSleepNet, a novel deep graph neural network, for automatic sleep stage classification. The GraphSleepNet is used to learn an adaptive graph structure representation that best serves spatial-temporal-GCN (ST-GCN) for sleep stage classification. The code has been released at: https://github.com/jingwang2020/GraphSleepNet.

Overall, the main contributions of the proposed GraphSleepNet for automatic sleep stage classification are summarized as follows:

- To the best of our knowledge, it is the first attempt to apply ST-GCN for automatic sleep stage classification. Moreover, we propose a novel adaptive sleep graph learning mechanism, which is integrated with ST-GCN simultaneously in a unified network architecture.
- We design a spatial-temporal convolution, which consists of graph convolutions for capturing spatial features and temporal convolutions for capturing the transition among different sleep stages.
- Experimental results demonstrate that the GraphSleepNet achieves state-of-the-art performance in sleep stage classification.

## 2 Related Work

### 2.1 Sleep Stage Classification

Identifying sleep stages plays an important role in diagnosing and treating sleep disorders. In earlier research, machine learning methods such as Support Vector Machine (SVM) and Random Forests (RF) are utilized to classify sleep stages. However, these methods need to extract hand-crafted features, which requires a lot of prior knowledge. Therefore, some researchers have turned to use deep learning methods for sleep stage classification.

Currently, two popular deep learning models, CNN and RNN, are widely used in sleep stage classification. A fast discriminative complex-valued CNN (FDCCNN) [Zhang and Wu, 2017] is proposed to capture the sleep information hidden inside EEG signals. A CNN model based on multivariate and multimodal physiological signals [Chambon *et al.*, 2018] takes into account the transitional rules of sleep stages to assist classification. The extracted functional connectivity metrics are employed as multi-dimensional images to input into the CNN model for classification [Chriskos *et al.*, 2019]. A hierarchical RNN named SeqSleepNet [Phan *et al.*, 2019] tackles the task as a sequence-to-sequence classification task. At the same time, hybrid models are also used by some researchers. DeepSleepNet [Supratak *et al.*, 2017] utilizes CNN to extract time-invariant features, and Bi-directional Long Short-Term Memory (BiLSTM) to learn the transition rules among sleep stages. A hierarchical neural network [Sun *et al.*, 2019] implements comprehensive feature learning stage and sequence learning stage, respectively. Additionally, with the development of attention mechanisms, a deep Bi-directional RNN with attention mechanism is used for single-channel sleep stage classification [Phan *et al.*, 2018].

Although the models above can extract spatial or temporal features from EEG signals, there is one obvious limitation: their input must be grid data. As different brain regions are not in the Euclidean space, grid data may not be the optimal data representation. Hence, the graph is the most appropriate data structure.

### 2.2 Graph Convolution Network

Recently, GCNs have shown advanced performance in addressing graph structure data. Graph convolution methods are classified into spatial convolution and spectral convolution methods. Spatial convolution methods usually perform graph convolution operations directly on the nodes and their neighbors of the graph. For example, an end-to-end CNN that operates on a graph is presented in [Duvenaud *et al.*, 2015]. The GraphSAGE model [Hamilton *et al.*, 2017] generates embeddings by sampling and aggregating features from a node's local neighborhood. Graph Attention Networks (GAT) [Veličković *et al.*, 2018] uses masked self-attentional layers to assign different weights to different nodes in a neighborhood. Spectral convolution methods usually define graph convolution operation on the spectral representation of the graph. For example, a general graph convolution framework is proposed based on graph Laplacian [Bruna *et al.*, 2014]. Chebyshev expansion of graph Laplacian is utilized to reduce computational complexity [Defferrard *et al.*, 2016]. A simpler GCN for semi-supervised learning is proposed in [Kipf and Welling, 2016]. Although, the existing GCNs usually utilize a fixed graph, which is not optimal for sleep stage classification, because the graph structure of different sleep states may be different. Moreover, the GCNs are applied directly to classify sleep stages without the capture of sleep transition rules in the temporal dimension.

## 3 Preliminaries

In our study, a sleep stage network is defined as an undirected graph $G = (V, E, \boldsymbol{A})$, where $V$ denotes the set of vertices and each vertex in the network represents an electrode on brain; $|V| = N$ is the number of vertices in sleep stage network; $E$ denotes the set of edges and indicates the connection between vertices; $\boldsymbol{A}$ denotes the adjacency matrix of sleep stage network $G$. As shown in Figure 1, $G_t$ is constructed from a 30s EEG signal sequence $S_t$. It is worth noting that the adjacency matrix $\boldsymbol{A}$ used in the proposed model is obtained through learning, while traditional GCNs usually use a fixed adjacency matrix.

The sleep feature matrix is the input of the GraphSleepNet. We define the raw signals sequence as $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_L) \in \mathbb{R}^{L \times N \times T_s}$, where $L$ denotes the number of samples, $T_s$ denotes the time series length of each sample $\boldsymbol{s}_i \in \boldsymbol{S}(i \in \{1, 2, \cdots, L\})$. For each sample $\boldsymbol{s_i}$, we extract the differential entropy (DE) features [Hyvärinen, 1998] on various frequency bands and define each sample $\boldsymbol{s_i}$'s feature matrix $\boldsymbol{X}_i = \left(\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_N^i\right)^T \in \mathbb{R}^{N \times F_{de}}$, where $\boldsymbol{x}_n^i \in \mathbb{R}^{F_{de}} (n \in \{1, 2, \cdots, N\})$ denotes $F_{de}$ features of node $n$ at sample $i$. Figure 1 shows the sleep stage network and sleep feature matrix, which are constructed from the raw signals.

The research goal is to learn the mapping relationship between the encoded signals and sleep stage class based on the proposed model. The problem of sleep stage classification is defined as: given $\boldsymbol{\mathcal{X}} = (\boldsymbol{X}_{i-d}, \dots, \boldsymbol{X}_i, \dots, \boldsymbol{X}_{i+d}) \in \mathbb{R}^{N \times F_{de} \times T_n}$ identify the current sleep stage $y$, where $\boldsymbol{\mathcal{X}}$ denotes the temporal context of $\boldsymbol{X}_i$, $y$ denotes the $\boldsymbol{X}_i$'s sleep stage class label, and $T_n = 2d + 1$ denotes the length of sleep stage networks, where $d \in \mathbb{N}^+$ denotes temporal context coefficient.

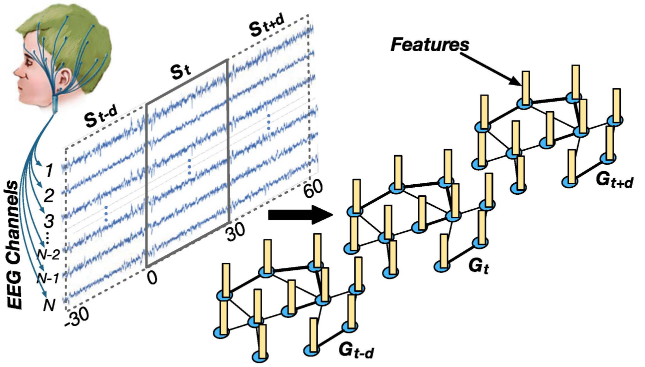


Figure 1: Sleep stage network and sleep feature matrix.

## 4 Adaptive Spatial-Temporal GCN

The overall architecture of the GraphSleepNet is shown in Figure 2. We summarize three key ideas of the GraphSleepNet: 1) Represent the pairwise relationship between nodes to dynamically construct adjacency matrix $\boldsymbol{A}$. 2) Combine spatial graph convolution and temporal convolution to extract both spatial and temporal features. 3) Employ a spatial-temporal attention mechanism to automatically pay relatively more attention to valuable spatial-temporal information. The overall architecture is elaborately designed to accurately identify sleep stages.

### 4.1 Adaptive Sleep Graph Learning

Adaptive sleep graph learning is the first key part of the GraphSleepNet as shown in Figure 3. It dynamically learns the graph structure, rather than the graph constructed by prior knowledge or artificially (e.g., $k$-nearest neighbor graph [Jiang *et al.*, 2013]). Therefore, we define a non-negative function $A_{mn} = g(\boldsymbol{x}_m, \boldsymbol{x}_n)\, (m, n \in \{1, 2, \cdots, N\})$ to represent the connection relationship between nodes $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ based on the input feature matrix $\boldsymbol{X}_i = \left(\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \dots, \boldsymbol{x}_N^i\right)^T \in \mathbb{R}^{N \times F_{de}}$. $g(\boldsymbol{x}_m, \boldsymbol{x}_n)$ is implemented through a layer neural network, which has the learnable weight vector $\boldsymbol{w} = (w_1, w_2, \dots, w_{F_{de}})^T \in \mathbb{R}^{F_{de} \times 1}$. The learned graph structure (adjacency matrix) $\boldsymbol{A}$ is defined as:

$$A_{mn} = g(\boldsymbol{x}_m, \boldsymbol{x}_n) = \frac{\exp(\mathrm{ReLU}(\boldsymbol{w}^T |\boldsymbol{x}_m - \boldsymbol{x}_n|))}{\sum_{n=1}^N \exp\left(\mathrm{ReLU}\left(\boldsymbol{w}^T \left|\boldsymbol{x}_m - \boldsymbol{x}_n\right|\right)\right)} \tag{1}$$

where rectified linear unit (ReLU) is an activation function to guarantee that $A_{mn}$ is non-negative. The softmax operation normalizes each row of $\boldsymbol{A}$. The weight vector $\boldsymbol{w}$ is updated by minimizing the following loss function,

$$\mathcal{L}_{\text{graph\_learning}} = \sum_{m,n=1}^{N} \|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2^2 A_{mn} + \lambda \|\boldsymbol{A}\|_F^2 \tag{2}$$

That is, the larger distance $\|\boldsymbol{x}_m - \boldsymbol{x}_n\|_2$ between $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, the smaller $A_{mn}$ is. Due to the brain connection structure is not a fully connected graph, we utilize the second term in the loss function to control the sparsity of graph $\boldsymbol{A}$, where $\lambda \geq 0$ is a regularization parameter.

The proposed graph learning mechanism automatically constructs the neighborhood connection of the nodes. To avoid the trivial solution (i.e., $\boldsymbol{w} = (0, 0, \cdots, 0)$), which is due to minimizing the above loss function $\mathcal{L}_{\text{graph\_learning}}$ independently, we utilize it as a regularized term to form the final loss function.

$$\mathcal{L}_{\text{loss}} = \mathcal{L}_{\text{cross\_entropy}} + \mathcal{L}_{\text{graph\_learning}} \tag{3}$$

$$\mathcal{L}_{\text{cross\_entropy}} = -\frac{1}{L} \sum_{i=1}^{L} \sum_{r=1}^{R} y_{i,r} \log \hat{y}_{i,r} \tag{4}$$

where $\mathcal{L}_{\text{cross\_entropy}}$ denotes the original loss function of the multi-classification task, $L$ denotes the number of samples, $R$ denotes the number of classes. $y$ is the true label and $\hat{y}$ is the value predicted by the model.

### 4.2 Spatial-Temporal Graph Convolution

Spatial-temporal graph convolution is a combination of spatial graph convolution and temporal standard convolution, which is used to extract both spatial and temporal features. As shown in Figure 2, the spatial features are extracted by aggregating information from neighbor nodes for each sleep stage network and the temporal features are captured by exploiting temporal dependencies from neighbor sleep stages.

#### Spatial Graph Convolution

We use graph convolution based on spectral graph theory to extract spatial features in the spatial dimension. For each sleep stage to be identified, the adaptive sleep graph learning module provides an adjacency matrix $\boldsymbol{A}$ for graph convolution. In addition, we employ the Chebyshev expansion of graph Laplacian to reduce computational complexity. Chebyshev graph convolution [Defferrard *et al.*, 2016] using the $K - 1$ order polynomials is defined as:

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x \tag{5}$$

where $g_\theta$ denotes the convolution kernel, $*_G$ denotes the graph convolution operation, $\theta \in \mathbb{R}^K$ denotes a vector of Chebyshev coefficients and $x$ is the input data. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ denotes Laplacian matrix, where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is degree matrix. $\tilde{\boldsymbol{L}} = \frac{2}{\lambda_{\max}} \boldsymbol{L} - \boldsymbol{I}_N$, where $\lambda_{\max}$ is Laplacian matrix's maximum eigenvalue and $\boldsymbol{I}_N$ is a identity matrix. $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ denotes the Chebyshev polynomials recursively, where $T_0(x) = 1$ and $T_1(x) = x$.

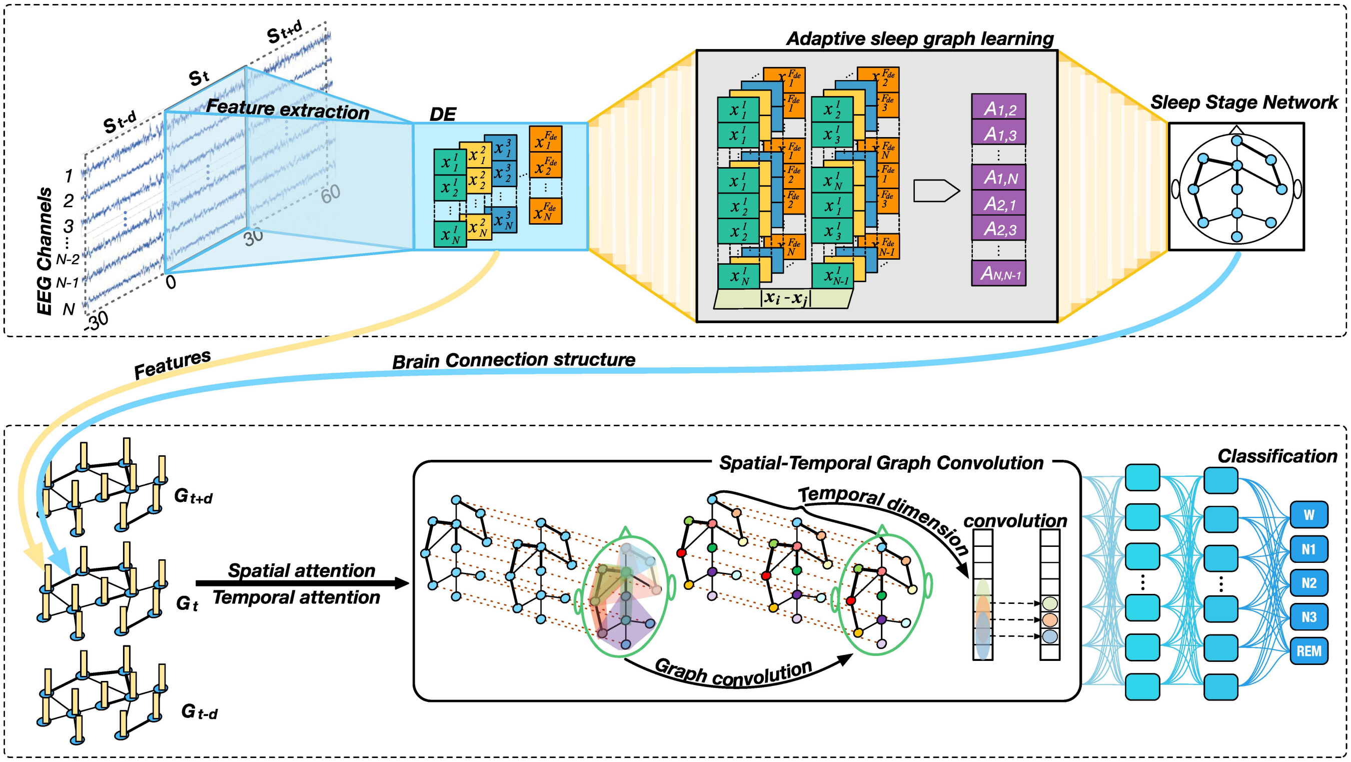


Figure 2: The overall architecture of the proposed GraphSleepNet for sleep stage classification. The GraphSleepNet consists of an adaptive sleep graph learning mechanism and the spatial-temporal graph convolution with spatial-temporal attention.

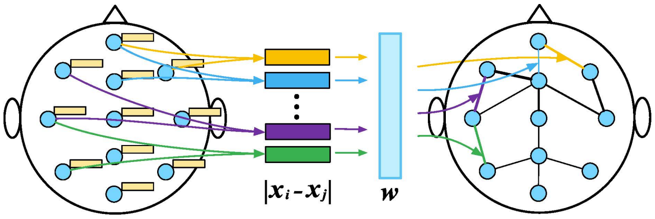


Figure 3: The structure of the proposed adaptive sleep graph learning in GraphSleepNet.

By using the approximate expansion of Chebyshev polynomial, the information of neighboring 0 to $K-1$ order neighbors centered at each node is extracted.

In this work, we generalize the above definition to the nodes with multiple channels. The $l$-th layer's input is $\hat{\mathcal{X}}^{(l-1)} = (\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_{T_{l-1}}) \in \mathbb{R}^{N \times C_{l-1} \times T_{l-1}}$, where $C_{l-1}$ denotes channel's number of each node, i.e., $l = 1, \quad C_0 = F_{de}$. $T_{l-1}$ denotes the $l$-th layer's temporal dimension. For each $\hat{\boldsymbol{X}}_i$, we obtain $g_\theta *_G \hat{\boldsymbol{X}}_i$ by using $C_l$ filters on $\hat{\boldsymbol{X}}_i$, where $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_{C_l}) \in \mathbb{R}^{K \times C_{l-1} \times C_l}$ is the convolution kernel parameter [Defferrard *et al.*, 2016]. Hence, the information of the $0 \sim K-1$ order neighbors is aggregated to each node.

### Temporal Convolution

To capture the sleep transition rules, which are used by sleep experts to classify the current sleep stage in combination with neighboring sleep stages, we employ CNN to perform convolution operation in the temporal dimension. Specifically, after graph convolution operation has sufficiently extracted the spatial features from each sleep stage network, we implement a standard 2D convolution layer to extract the temporal context information of the current sleep stage. The temporal convolution operation on the $l$-th layer is defined as:

$$\mathcal{X}^{(l)} = \mathrm{ReLU}(\Phi * (\mathrm{ReLU}(g_\theta *_G \hat{\mathcal{X}}^{(l-1)}))) \in \mathbb{R}^{N \times C_l \times T_l} \tag{6}$$

where ReLU is the activation function, $\Phi$ denotes the convolution kernel's parameters, $*$ denotes the standard convolution operation.

## 4.3 Spatial-Temporal Attention

The attention mechanism is often used to automatically extract valuable information. In this study, we employ a spatial-temporal attention mechanism [Guo *et al.*, 2019] to capture valuable spatial-temporal information on the sleep stage network. The spatial-temporal attention mechanism contains spatial attention and temporal attention.

### Spatial Attention

In the spatial dimension, different regions have different effects on the sleep stage which is dynamically changing during sleep. To automatically extract the attentive spatial dynamics, we utilize a spatial attention mechanism, which is defined as follows:

$$\boldsymbol{P} = \boldsymbol{V}_p \cdot \sigma((\mathcal{X}^{(l-1)} \boldsymbol{Z}_1) \boldsymbol{Z}_2 (\boldsymbol{Z}_3 \mathcal{X}^{(l-1)})^T + \boldsymbol{b}_p) \tag{7}$$

$$P'_{m,n} = \mathrm{softmax}(P_{m,n}) \tag{8}$$

where $\boldsymbol{V}_p, \boldsymbol{b}_p \in \mathbb{R}^{N \times N}$, $\boldsymbol{Z}_1 \in \mathbb{R}^{T_{l-1}}$, $\boldsymbol{Z}_2 \in \mathbb{R}^{C_{l-1} \times T_{l-1}}$, $\boldsymbol{Z}_3 \in \mathbb{R}^{C_{l-1}}$ denote learnable parameters, $\sigma$ denotes the sigmoid activation function. $\boldsymbol{\mathcal{X}}^{(l-1)} = \left(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{T_{l-1}}\right) \in R^{N \times C_{l-1} \times T_{l-1}}$ denotes the $l$-th layer's input. $\boldsymbol{P}$ denotes spatial attention matrix, which is dynamically computed by current layer's input. $P_{m,n}$ denotes the correlation between node $m$ and $n$. The softmax operation is utilized to normalize the attention matrix $\boldsymbol{P}$. In the proposed model, when the graph convolution is performed, the learned adjacency matrix $\boldsymbol{A}$ and spatial attention matrix $\boldsymbol{P}$ can dynamically adjust the update of nodes.

### Temporal Attention

In the temporal dimension, there are correlations among neighboring sleep stages, and the correlations change in different situations. Therefore, a temporal attention mechanism is utilized to capture dynamic temporal information among sleep stage networks.

The temporal attention mechanism is defined as follows:

$$\boldsymbol{Q} = \boldsymbol{V}_q \cdot \sigma(((\boldsymbol{\mathcal{X}}^{(l-1)})^T \boldsymbol{M}_1)\boldsymbol{M}_2(\boldsymbol{M}_3 \boldsymbol{\mathcal{X}}^{(l-1)}) + \boldsymbol{b}_q) \tag{9}$$

$$Q'_{m,n} = \mathrm{softmax}(Q_{m,n}) \tag{10}$$

where $\boldsymbol{V}_q, \boldsymbol{b}_q \in \mathbb{R}^{T_{l-1} \times T_{l-1}}$, $\boldsymbol{M}_1 \in \mathbb{R}^{N}$, $\boldsymbol{M}_2 \in \mathbb{R}^{C_{l-1} \times N}$, $\boldsymbol{M}_3 \in \mathbb{R}^{C_{l-1}}$ denotes learnable parameters. $Q_{m,n}$ denotes the strength of correlation between sleep stage network $m$ and $n$. Finally, the softmax operation is utilized to normalize the attention matrix $\boldsymbol{Q}$. The input of the ST-GCN is tuned by the temporal attention: $\hat{\boldsymbol{\mathcal{X}}}^{(l-1)} = (\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_{T_{l-1}}) = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{T_{l-1}})\boldsymbol{Q}' \in \mathbb{R}^{N \times C_{l-1} \times T_{l-1}}$ to pay more attention to valuable temporal information.

## 5 Experiments

To verify the effectiveness of the GraphSleepNet, we evaluate it on the Montreal Archive of Sleep Studies (MASS)-SS3 dataset [O'Reilly *et al.*, 2014].

### 5.1 Dataset

We use the MASS-SS3 dataset, which contains the PSG recordings from 62 healthy subjects (28 male and 34 female). Each recording contains 20 EEG channels, 2 EOG channels, 3 EMG channels, and 1 ECG channel. In addition, experts classify these PSG recordings into five sleep stages (W, N1, N2, N3, and REM) according to AASM standard [Berry *et al.*, 2012].

### 5.2 Preprocessing

The PSG recordings are pre-processed with the bandpass filters of 0.30-100 Hz (EEG), 0.10-100 Hz (EOG, ECG) and 10-100 Hz (EMG). We extract DE features for each channel from 9 crossed frequency bands: 0.5-4 Hz, 2-6 Hz, 4-8 Hz, 6-11 Hz, 8-14 Hz, 11-22 Hz, 14-31 Hz, 22-40 Hz, 31-50 Hz. To make a fair comparison of existing sleep classification methods, we employ all signals in our experiments.

### 5.3 Baseline Methods

We compare the GraphSleepNet with the following baselines:

- **[Dong *et al.*, 2017]**: A mixed neural network, which combines multilayer perceptron (MLP) and LSTM, and also compare its performance with RF and SVM.
- **[Supratak *et al.*, 2017]**: A model combines CNN and BiLSTM to capture both time-invariant features and transition rules among sleep stages.
- **[Chambon *et al.*, 2018]**: A temporal sleep stage classification use multivariate and multimodal time series.
- **[Phan *et al.*, 2019]**: SeqSleepNet changes the single sleep stage classification problem into a sequence-to-sequence classification problem by using attention-based bidirectional RNN (ARNN) and RNN.
- **[Sun *et al.*, 2019]**: A hierarchical neural network, which learns comprehensive features and sequence respectively.
- **[Jiang *et al.*, 2019]**: Robust sleep stage classification which uses multimodal decomposition and Hidden Markov Model (HMM) -based refinement.

### 5.4 Experiment Settings

We employ 31-fold cross-validation to evaluate the performance of the GraphSleepNet. In addition, we use the subject-independent schemes to randomly divide the training set and test set. The hyperparameters of the experiment are shown in Table 1. We implement the GraphSleepNet using TensorFlow. Please refer to the code for more detailed.

| Hyperparameter description | Value |
|---|---|
| Sleep stage networks length $T_n$ | 5 |
| Layer number of ST-GCN | 1 |
| Standard convolution kernels | 10 |
| Graph convolution kernels | 10 |
| Chebyshev polynomial K | 3 |
| Regularization parameter | 0.001 |
| Dropout probability | 0.5 |
| Batchsize | 64 |
| Number of training epochs | 50 |
| Learning rate | 0.001 |
| Optimizer | Adam |

Table 1: Experiment hyperparameter setting

### 5.5 Comparison with State-of-the-Art Methods

We compare our GraphSleepNet with the other eight baseline methods for sleep stage classification on the MASS-SS3 dataset, as shown in Table 2. The results show that our GraphSleepNet consistently achieves the best performance on all evaluation indicators compared to other baseline methods.

The traditional machine learning methods (SVM and RF) cannot learn the complex spatial or temporal features well. However, existing deep learning models such as CNN and RNN [Dong *et al.*, 2017; Supratak *et al.*, 2017; Chambon *et al.*, 2018; Phan *et al.*, 2019; Sun *et al.*, 2019] can directly

| | Method | Overall results | | | F1-score for each class | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy | F1-score | Kappa | Wake | N1 | N2 | N3 | REM |
| [Dong *et al.*, 2017] | SVM | 0.797 | 0.750 | - | 0.786 | 0.487 | 0.861 | 0.825 | 0.792 |
| [Dong *et al.*, 2017] | RF | 0.817 | 0.724 | - | 0.782 | 0.351 | 0.880 | 0.815 | 0.794 |
| [Dong *et al.*, 2017] | MLP+LSTM | 0.859 | 0.805 | - | 0.846 | 0.563 | 0.907 | 0.848 | 0.861 |
| [Supratak *et al.*, 2017] | CNN+BiLSTM | 0.862 | 0.817 | 0.800 | 0.873 | 0.598 | 0.903 | 0.815 | 0.893 |
| [Chambon *et al.*, 2018] | CNN | 0.739 | 0.673 | 0.640 | 0.730 | 0.294 | 0.812 | 0.765 | 0.764 |
| [Jiang *et al.*, 2019] | RF+HMM | 0.808 | 0.793 | 0.710 | - | - | - | - | - |
| [Phan *et al.*, 2019] | ARNN+RNN | 0.871 | 0.833 | 0.815 | - | - | - | - | - |
| [Sun *et al.*, 2019] | CNN+BiLSTM | 0.881 | 0.824 | 0.819 | 0.912 | 0.551 | 0.916 | 0.826 | 0.914 |
| **GraphSleepNet** | **Adaptive ST-GCN** | **0.889** | **0.841** | **0.834** | **0.913** | **0.603** | **0.921** | **0.851** | **0.919** |

Table 2: The performance comparison of the state-of-the-art approaches on the MASS dataset

extract the spatial or temporal features. Therefore, their performance is better than the methods based on traditional machine learning.

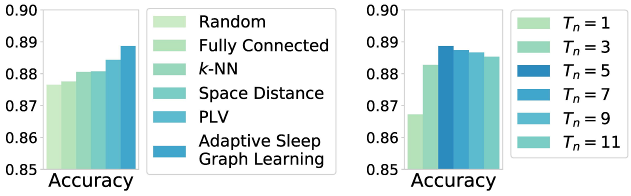
Although CNN and RNN achieve high accuracy, their limitation is that the model's input must be grid data ignoring the connection among regions. Due to brain regions are in non-Euclidean space, graph is the most appropriate data structure to indicate the connection. The proposed GraphSleepNet extracts both spatial and temporal features based on an adaptive sleep graph structure. Therefore, the proposed GraphSleepNet is superior to other baseline methods.

### 5.6 Experimental Analysis

To further investigate the effectiveness of the adaptive sleep graph learning, we design five fixed adjacency matrices to compare with it. We compare these different adjacency matrices with an adaptive adjacency matrix. The different adjacency matrices are defined as below:

- Fully Connected Adjacency Matrix: A matrix whose elements are all 1. It represents that there are all connections among all nodes and each node also has self-connection in the graph.
- Random Adjacency Matrix: A randomly generated matrix with each element's value between 0 and 1 indicating that the connections between nodes are random.
- Space Distance Adjacency Matrix: A matrix constructed based on the actual spatial distance between electrodes.
- $K$-Nearest Neighbor Adjacency Matrix [Jiang *et al.*, 2013]: A matrix, which represents a $k$-nearest neighbor graph. That is, each node has $k$ neighbor nodes.
- PLV Adjacency Matrix [Aydore *et al.*, 2013]: A matrix generated by PLV method between each pair of nodes.

Figure 4(a) illustrates that the adaptive (learned) adjacency matrix achieves the highest accuracy for sleep stage classification. In addition, the adjacency matrix combined with prior knowledge also achieves a suboptimal effect, such as the PLV adjacency matrix. The fully connected adjacency matrix does not work well because the brain network is not a fully connected graph. In general, the adjacency matrix can significantly affect the classification effect. The proposed adaptive sleep graph learning for classification tasks is superior to all fixed graphs.



(a)

(b)

Figure 4: The experimental analysis. (a) Comparison of adjacency matrices. (b) Comparison of the sleep stage network length.

In addition, we study the impact of the number of input sleep stage networks (temporal context) on the classification performance. Figure 4(b) presents the performance of the classification improves as the number of input sleep stage networks $T_n$ increases, and the best accuracy is achieved when $T_n = 5$. It is worth noting that according to the AASM standard, sleep experts also judge the current sleep state based on 5 sleep stages, which shows the practicability of the proposed model. However, too long time context is input, resulting in degradation of classification performance, which may be caused by information redundancy.

## 6 Conclusion

In this paper, we propose a novel deep graph neural network for automatic sleep stage classification. The main advantage of the proposed model is to adaptively learn the sleep connection structure that best serves ST-GCN for the classification task. Furthermore, this model combines spatial-temporal convolution and spatial-temporal attention mechanisms to simultaneously capture attentive spatial-temporal features of the sleep data. Experimental results demonstrate our model achieves state-of-the-art performance in the existing models. In addition, our proposed model is a general-framework for multivariate physiological time series.

## Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61603029).

## References

[Aydore *et al.*, 2013] Sergul Aydore, Dimitrios Pantazis, and Richard M Leahy. A note on the phase locking value and its properties. *Neuroimage*, 74:231–244, 2013.

[Berry *et al.*, 2012] Richard B Berry, Rohit Budhiraja, Daniel J Gottlieb, David Gozal, Conrad Iber, Vishesh K Kapur, Carole L Marcus, Reena Mehra, Sairam Parthasarathy, Stuart F Quan, et al. Rules for scoring respiratory events in sleep: update of the 2007 aasm manual for the scoring of sleep and associated events. *Journal of clinical sleep medicine*, 8(05):597–619, 2012.

[Bruna *et al.*, 2014] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

[Chambon *et al.*, 2018] Stanislas Chambon, Mathieu N Galtier, Pierrick J Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018.

[Chriskos *et al.*, 2019] Panteleimon Chriskos, Christos A Frantzidis, Polyxeni T Gkivogkli, Panagiotis D Bamidis, and Chrysoula Kourtidou-Papadeli. Automatic sleep staging employing convolutional neural networks and cortical connectivity images. *IEEE transactions on neural networks and learning systems*, 2019.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[Dong *et al.*, 2017] Hao Dong, Akara Supratak, Wei Pan, Chao Wu, Paul M Matthews, and Yike Guo. Mixed neural network approach for temporal sleep stage classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(2):324–333, 2017.

[Duvenaud *et al.*, 2015] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[Hyvärinen, 1998] Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in neural information processing systems*, pages 273–279, 1998.

[Jiang *et al.*, 2013] Bo Jiang, Chris Ding, and Jin Tang. Graph-laplacian pca: Closed-form solution and robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3492–3498, 2013.

[Jiang *et al.*, 2019] Dihong Jiang, Ya-nan Lu, MA Yu, and WANG Yuanyuan. Robust sleep stage classification with single-channel eeg signals using multimodal decomposition and hmm-based refinement. *Expert Systems with Applications*, 121:188–203, 2019.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[O'Reilly *et al.*, 2014] Christian O'Reilly, Nadia Gosselin, Julie Carrier, and Tore Nielsen. Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of sleep research*, 23(6):628–635, 2014.

[Phan *et al.*, 2018] Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y Chén, and Maarten De Vos. Automatic sleep stage classification using single-channel eeg: Learning sequential features with attention-based recurrent neural networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1452–1455. IEEE, 2018.

[Phan *et al.*, 2019] Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y Chén, and Maarten De Vos. Seqsleepnet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410, 2019.

[Sun *et al.*, 2019] Chenglu Sun, Chen Chen, Wei Li, Jiahao Fan, and Wei Chen. A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning. *IEEE journal of biomedical and health informatics*, 2019.

[Supratak *et al.*, 2017] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wolpert, 1969] Edward A Wolpert. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *Archives of General Psychiatry*, 20(2):246–247, 1969.

[Zhang and Wu, 2017] Junming Zhang and Yan Wu. A new method for automatic sleep stage classification. *IEEE transactions on biomedical circuits and systems*, 11(5):1097–1110, 2017.