SSGCNet: A Sparse Spectra Graph Convolutional Network for Epileptic EEG Signal Classification

Jialin Wang, Rui Gao, *Member, IEEE*, Haotian Zheng, Hao Zhu, and C.-J. Richard Shi, *Fellow, IEEE*

***Abstract*— In this article, we propose a sparse spectra graph convolutional network (SSGCNet) for epileptic electroencephalogram (EEG) signal classification. The goal is to develop a lightweighted deep learning model while retaining a high level of classification accuracy. To do so, we propose a weighted neighborhood field graph (WNFG) to represent EEG signals. The WNFG reduces redundant edges between graph nodes and has lower graph generation time and memory usage than the baseline solution. The sequential graph convolutional network is further developed from a WNFG by combining sparse weight pruning and the alternating direction method of multipliers (ADMM). Compared with the state-of-the-art method, our method has the same classification accuracy on the Bonn public dataset and the spikes and slow waves (SSW) clinical real dataset when the connection rate is ten times smaller.**

***Index Terms*— Alternating direction method of multipliers (ADMM), electroencephalogram (EEG) signal classification, graph neural network (GNN), nonconvextiy, weight pruning.**

NOMENCLATURE

$\mathbf{X}$	Matrix.
$\mathbf{x}$	Vector.
x_i	ith element of vector $\mathbf{x}$.
$(\cdot)^\top$	Transposition.
$\mathbf{x}^{(k)}$	Value of $\mathbf{x}$ at the k:th iteration.
$\text{vec}(\cdot)$	Vectorization operator.
$\mathcal{G}$	Graph representation.
$\text{card}(\cdot)$	Return the number of nonzero elements.
$\|\cdot\|_2$	Standard ℓ_2-norm.
const	Constant.

Manuscript received 24 March 2022; revised 25 September 2022, 7 December 2022, 28 January 2023, and 15 February 2023; accepted 21 February 2023. This work was supported in part by the Science and Technology Commission of Shanghai Municipality under Grant 2018SHZDZX01 and Grant 2021SHZDZX, in part by the Natural Science Foundation of Shanghai Municipality under Grant 22ZR1406900, in part by the Young Scientist Project of MOE Innovation Platform, and in part by the Shanghai Pujiang Program under Grant 22PJ1405400. *(Corresponding author: Rui Gao.)*

Jialin Wang, Haotian Zheng, and Hao Zhu are with the State Key Laboratory of ASIC and Systems, the Institute of Brain-Inspired Circuits and Systems, and the Zhangjiang Fudan International Innovation Center, Fudan University, Shanghai 201203, China.

Rui Gao is with the Department of Naval Architecture and Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: rui.gao@ieee.org).

C.-J. Richard Shi is with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3252569.

Digital Object Identifier 10.1109/TNNLS.2023.3252569

I. INTRODUCTION

EPILEPSY is one of the most common neurological diseases, which is accompanied by super-synchronous abnormal discharge of electroencephalogram (EEG) signals [1], [2]. In general, doctors need to detect epileptic seizures from dozens of hours of EEG signals. However, the duration of epileptic seizures is pretty short (only a few seconds) [3]. Therefore, it is crucial to detect epileptic seizures from massive EEG signals automatically. In this article, the objective is to develop an autonomous effective method to classify epileptic EEG signals.

The effective data representation of EEG signals is an essential prerequisite for EEG signals classification [4]. The representation methods of EEG signals are generally based on the spatial–temporal structure [5] and the transition network [6]. Due to the complexity of EEG waveform, the methods usually have the limitation on representing the relationship between EEG signal sampling points, which makes these methods challenging to extract relationship features between sampling points [7], [8], [9]. In recent years, converting signals into graphs has received extensive attention [6]. Although the graph representation-based methods can effectively extract the relationship between signal sampling points, these methods have limitations in building effective connections between sample points, especially when there exist a lot of redundant edges in the graph representation. This redundancy leads to an enormous generation time and memory usage, that may limit the promotion of graph representation methods on portable hardware devices.

Recently, traditional machine learning methods have been used for EEG signal classification tasks, for example, empirical mode decomposition method, hybrid-type machine learning method, and logistic model tree-based method [10], [11], [12]. However, most of the traditional methods require manual feature selection on EEG signals, which heavily relies on researcher's experience and domain knowledge of EEG signals. These methods can be biased when the human expert is subjective [13]. Therefore, it is of great value to propose an autonomous method for epileptic EEG signal classification.

Deep learning methods have received extensive attention in autonomous classification [14], [15]. Input-to-output deep learning models can independently extract useful features from data [16], [17]. However, it is difficult to directly obtain the hidden features from the original data [18], [19]. For extracting a large number of features from datasets, large-scale

deep learning models have been derived by introducing extra prior [20], [21]. The training network problem is formulated as a sparse optimization problem with L_1-penalized terms, and solve such the problem by the sparse regularization approach [22]. With increasing model scale, they have to take a huge of computing resources and memory storage, which leads that signal classification scenarios cannot be deployed on practical low-power device. Hence, it becomes promising to optimize lightweight deep learning models arising in epileptic EEG signal classification.

In this article, we propose a sparse spectra graph convolutional network (SSGCNet) for epileptic EEG signal classification. We first represent EEG signals as a frequency domain graph representation, and then use the sequential convolutional module to extract features between graph nodes. Under sparsity constraints, we introduce the alternating direction method of multipliers (ADMM)-type splitting and weight pruning strategy, which can compress the model while retaining the classification accuracy. Experimental results demonstrate the promising performance of the proposed SSGCNet in various real-world datasets. The main contributions of the article are summarized as follows.

1) We present a weighted neighborhood field graph (WNFG) representation method to represent epileptic EEG signals, which effectively extracts the node relationship features and sequential features.
2) We develop a sparse spectra graph convolutional neural network model, which achieves ten times compression rate with high classification accuracy.
3) We formulate the deep learning training problem as a constrained nonconvex problem, and then analyze the convergence results under mild assumptions.
4) We apply our SSGCNet method to several clinical-real applications.

The main advantage of our model is that the computational cost and space occupancy rate are much less than in other traditional methods. The average redundant edge of our graph representation is reduced by ten times on public datasets and ten times on clinical-real datasets, respectively. Our model compression exceeds other deep learning models in the epileptic EEG signal classification around ten times.

The rest of this article is structured as follows. We introduce related work in Section II. In Section III, we introduce the graph representation of EEG signals, where the model can effectively extract the node relationship features in the sparse graph representation. The proposed sparse spectra graph convolutional neural network method is presented in Section IV. In Section V, by public and clinical-real datasets, various experimental results demonstrate the effectiveness and accuracy. The discussions of the proposed method are illustrated in Section VI. Section VII draws the concluding remarks.

The notation is listed in Nomenclature.

II. Related Works

In this section, we introduce the related work of EEG signal data represent methods in Section II-A and EEG signal classification methods in Section II-B.

A. EEG Signal Data Representation Methods

The most common data representation-based method is to represent EEG signals in the time domain, the frequency domain, or their combination [5], [13]. However, the relationship between the sampling points of the EEG signal is ignored. With the recent increase in available computational capacities, the graph representation-based method has recently attracted much attention [7], [8]. The earliest signal graph representation method is the visibility graph (VG), which uses a principle called "Connection Criterion" (see Table I) to build edges [23]. The horizontal VG (HVG) [24] is derived from VG, which uses a simpler connection criterion than VG. The variants of VG and HVG are limited penetration VG (LPVG) proposed by Zhou et al. [25] and limited penetration HVG (LPHVG) proposed by Gao et al. [26], focusing on processing data with different granularities. Wang et al. [27] conducted a more detailed study on the LPHVG topology.

Recently, numerous research efforts have been dedicated to develop the utilization of graph neural networks (GNNs) [28]. In [1], a weighted HVG constructing algorithm is proposed to identify seizure from EEG signals. Depending on seizure patterns, EEG signal is converted into graphs, including basic VG and HVG, and is represented by these graphs [4]. Multiscale LPHVG [26] has also been applied to EEG signal analysis. However, most of the existing graph representation methods are limited by the above connection criterion to some extent. For epilepsy detection, this restriction is usually too strict, resulting in too few edges to be created, which makes it difficult to distinguish between epileptic seizures and nonepileptic seizures. Therefore, the weighted overlook graph (WOG) method [29] enhances the ability of the graph representation method to distinguish EEG limit numbers by improving the connection criterion. Since these graph structures have many redundant edges, such graph representation occupies large computation space to store redundancy of weights. For this reason, we design a WNFG based on the EEG signal graph representation, which can significantly reduce the redundant edges between sampling points.

B. EEG Signal Classification Methods

There has been a growing body of literature addressing EEG signal classification problems based on various classifications models [30], [31], [32], [33], [34]. The methods usually take the model as a whole module and train the model with a large number of parameters. For example, the directed transfer function-based convolutional neural network is proposed to address EEG signal classification problems [5], [33]. Despite good performance, the method is time-consuming. Dropout techniques are then introduced to randomly drop units from the neural network during training [35]. The methods are optimized on multilayer perceptrons (MLPs) and convolutional neural networks [36], [37]. For example, the magnitude-based pruning method is proposed to compress the model in [38]. However, these methods have ignored the sparsity property, especially for large-scale networks. With the development of optimization methods, extensive research work has been explored on sparse weight pruning of neural network. Based

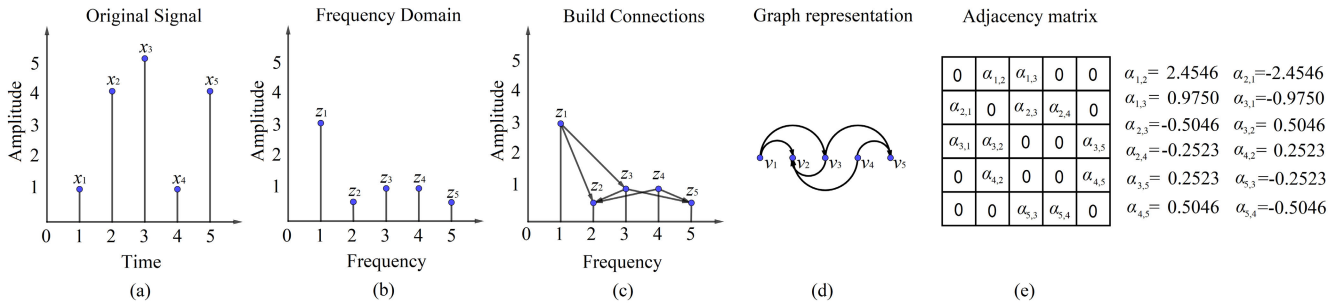

Fig. 1. (a) Original EEG signal $\mathbf{x}$. (b) Frequency domain signal $\mathbf{z}$. (c) Process of building connections. (d) Graph representation of the signal $\mathcal{G}$. (e) Adjacency matrix $\mathbf{A}$ of the graph after computing each $\alpha_{i,j}$.

on sparse regularization, clustering-based multitask feature learning algorithm is proposed to select informative features of EEG signals [39]. The pruning methods introduce L_1 and L_2 regularizers to optimize redundant weights. In [31], the L_1-regularized stage-wise pruning is employed and then set as a single pruning rate for all layers. While sparse weight pruning offers several benefits, they lack theoretical guarantees on pruning performance.

Variable splitting methods have recently been introduced in the arena of sparsity-type problems, where it has become increasingly important due to splitting property [40], [41]. The methods, for example ADMM [42], [43], are efficient methods that can tackle this kind of sparsity problem. Weight pruning-based neural network can be solved using ADMM [44], [45]. Instead of minimizing the given constrained minimization problem, ADMM can transform it into an unconstrained one. Its main advantage is the splitting ability that splits a complex problem into several simpler subproblems [46]. The weight pruning problem has been formulated as a nonconvex constrained problem with L_1-norm, but it penalizes individual elements of each weight vector instead of groups of elements in them [44]. In [45], the methods incorporate both ADMM and masked retraining to reduce the weights. While the empirical experiments show that ADMM converges to a stationary point, it lacks theoretical guarantees of convergence. In this article, we propose the ADMM weight pruning method that outperforms the existing methods in terms of pruning rate.

III. Sparse Spectra Graph Representation

In this section, we present a WNFG to represent EEG signals.

A. Preliminary

Before introducing our graph representation, we define the concept of graph representation in this article.

Definition 1 (EEG Signal): The EEG signal represents the electrical activity of brain cortex. The single-channel EEG signal (1-D time series) $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, where x_i is the ith sampling point of the EEG signal, and n is the total number of sampling points contained in the signal segment.

Definition 2 (Frequency Domain of EEG Signal): A frequency domain of EEG signal $\mathbf{z} = \{z_1, z_2, \ldots, z_n\}$ as an EEG signal $\mathbf{x}$ converted through Fourier transform

$$z_m = \sum_{i=1}^{n} x_i e^{-\bar{j}\frac{2\pi}{n}im}, \quad m = 1, 2, \ldots, n \tag{1}$$

where z_m is the frequency domain of the ith element of $\mathbf{x}$ [see Fig. 1(b)], and $\bar{j}$ stands for the imaginary unit.

TABLE I
Efficiency of the Different Graph Representations

Methods	Time	Connection Criterion
VG [23]	$O(n^3)$	$\frac{x_i - x_j}{i-j} < \frac{x_i - x_l}{i-l}$, $j < l < i$
LPVG [25]	$O(n^3)$	$\frac{x_i - x_j}{i-j} < \frac{x_i - x_{l_r}}{i-x_{l_r}}$, $j < l_r < i$, $l_r \in (0, m]$
HVG [24]	$O(n \log n)$	$x_l < x_i, x_j$, $j < l < i$
LPHVG [26]	$O(n \log n)$	$x_{l_r} < x_i, x_j$, $j < l_r < i$, $l_r \in (0, m]$
WOG [29]	$O(n^2)$	$x_j < x_i$
WNFG	$O(K \cdot n)$	$x_j < x_i$ with $(\lvert i - j \rvert < K)$.

Definition 3 (EEG Signal Graph Representation): $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is the graph representation of EEG signal. $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes represented by the graph, and the node v_i corresponds to the sampling point x_i. $\mathcal{E} = \{E_1, E_2, \ldots, E_n\}$ is the sets of edges for each node. Each edge set E_i of the ith node has the weight set $\{\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n}\}$, where represents the edge connecting two nodes v_i and v_j. The element $\alpha_{i,j}$ in the adjacency matrix A is

$$\alpha_{i,j} = x_i - x_j, \ i, \ j = 1, 2, \ldots, n. \tag{2}$$

Note that we only use the real part of $|z_m|$ of the frequency domain. The graph representation mentioned in this article is a directed graph. When $\alpha_{i,j} = x_i - x_j$, $\alpha_{j,i} = x_j - x_i$ is used to distinguish the direction.

B. Weighted Neighbor Field Graph

The method we proposed is based on the classical graph representation [23], [25], [29]. Considering two arbitrary data points x_i and x_j $(i < j)$ in an EEG signal $\mathbf{x}$, an weighted, directed edge $\alpha_{i,j}$ is created. We then have $x_i > x_j$. The time complexity of WOG is $O(n^2)$. Since this WOG method generates large number of edges, the main computational demand is in computing edges. Our main goal is to derive an efficient method for graph structure.

Here, we present the WNFG method. Given points x_i and x_j $(i < j)$ in the EEG signal $\mathbf{x}$, an weighted, directed edge

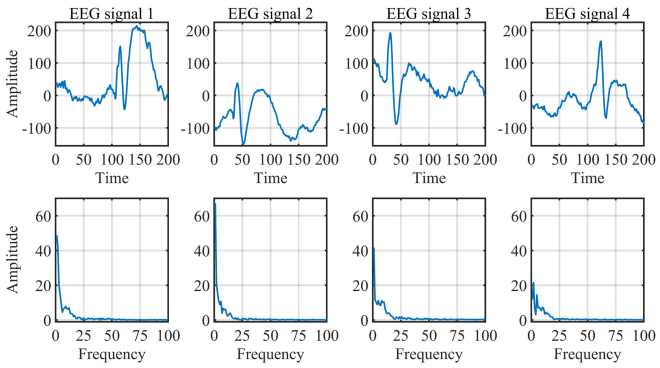

Fig. 2. Epileptic waveforms in the time domain (first row) and frequency domain (last row) EEG signal.

$\alpha_{i,j}$ is created. We have the relation

$$x_i > x_j \text{ with } (|i-j| < K) \tag{3}$$

where K is the neighbor field coefficient. Note that when $K = n$, WOG is a special case of WNFG.

As shown in Fig. 1(c), we set the neighbor field coefficient K with $K \ll n$. The time complexity of the graph structure then becomes $O(Kn)$, which is smaller than that of WOG. We now obtain a sparse graph structure with lower complexity. In particular, when $K = 1$, the graph structure of each signal segment is a chain structure. When K is small, the deep learning model is difficult to converge, which causes the model nonconvergent. In our article, the value of K is set in the range [20, 25], which simplifies the balanced graph structure and improves the stability of the classification accuracy.

Due to the interpretability of frequency domain, we then construct the *spectra* graph structure. As shown in Fig. 2, the epileptic waveforms in the time domain EEG signal are random in duration, amplitude, and phase. But, in the frequency domain, the spectra present the similar distribution (i.e., the low-frequency values are higher than high-frequency values).

In the WNFG, for any two data points z_i, $z_j \in \{z_m\}_{m=1}^n$, the distance range between i and j is less than K. As shown in Fig. 1(d), the connection rule between different points can be expressed as follows:

$$z_i > z_j \text{ with } (|i-j| < K). \tag{4}$$

The value of the edge between the data points is $\alpha_{i,j}$, which has the following equation:

$$\alpha_{i,j} = \frac{z_i - z_j}{i-j}. \tag{5}$$

After computing all $\alpha_{i,j}$, we obtain the whole adjacency matrix $\mathbf{A}$ of the graph [see Fig. 1(e)]. We use positive and negative signs as the connection direction for the direct selection.

As a result, our graph structure has an effective edge connection and low computation complexity. This is also consistent with the relationship between EEG signal rhythms, which is adaptable to real-world applications. In the following, we will propose the graph classification method for EEG signals.

IV. Graph Classification Method

In this section, we propose the sparse spectra graph convolutional neural network for epileptic EEG signal classification.

A. Problem Formulation

Let $\mathbf{w}_l \in \mathbb{R}^{n_w}$ be a weight parameter of the lth layer and $\mathbf{y} \in \mathbb{R}^n$ be the input data. The problem of training an N-layer deep learning model can be formulated as follows:

$$\{\hat{\mathbf{w}}, \hat{\mathbf{y}}\} = \arg\min_{\mathbf{w}_{1:N}} f(\mathbf{w}_{1:N}, \mathbf{y}) \quad \text{s.t.} \quad \mathbf{card}(\Omega \mathbf{w}_l) \le \text{const}, \quad l = 1, \ldots, N \tag{6}$$

where **card** returns the number of nonzero elements, $\hat{\mathbf{w}} \in \mathbb{R}^{n_w}$ is the optimal weight parameter sequence, $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$ is the output in deep learning model corresponding to the probability of seizure or nonseizure, $\Omega \in \mathbb{R}^{n_o \times n_w}$ is an operator, f is the loss function, and $\mathbf{w}_{1:N} = \text{vec}(\mathbf{w}_1, \ldots, \mathbf{w}_N)$. Our objective is to obtain the output $\hat{\mathbf{y}}$ and the parameters $\hat{\mathbf{w}}$.

Formulation (6) is amendable for loss functions, such as root-mean-square error, structure similarity index measure, min–max function, or cross-entropy-type function [47]. The settings of the operator Ω can be used to represent three kinds of pruning methods.

1) When Ω is an identity matrix, the constraint set $\{\mathbf{card}(\Omega \mathbf{w}_l) \le \text{const}\}$ represents the number of nonzero elements of the parameter $\mathbf{w}_l$.
2) When the matrix Ω is a learned matrix, an analysis sparse representation can be obtained [48]. The constraint set represents the number of grouped nonzero elements of the parameter $\mathbf{w}_l$.
3) When the matrix Ω is structured, the weights can be pruned in parallel.

It should be noted that, if all the products $\Omega \mathbf{w}_l$ are out of the constraint sets, the objective becomes training a deep learning model without any weight pruning. Due to the nonconvexity, it is challenging to solve such the problem, particularly when the parameters and the model are in large-scale size. To address the issue, we introduce the SSGCNet method, which combines the ADMM-type splitting and weight pruning strategy.

B. Framework of SSGCNet

In this section, we introduce the framework of SSGCNet. The SSGCNet includes one hop aggregation operations, four sequential convolutional layers, two fully connected layers, and weight pruning process. The aggregation operation is performed by multiplying the vector by the adjacency matrix. The sequential convolutional layers can accurately extract sequential features in the sequence. The fully connected structure can be equivalent to the readout structure in the GNN [28]. The framework of our SSGCNet is shown in Fig. 3. Although the node aggregation of the GNN belongs to a multihop operation for all nodes, our graph structure can completely retain the information of graph node aggregation in this process. We construct the aggregated part of the graph nodes as a single module for operations on the convolutional layer and fully connected layer. Then, we use the ADMM

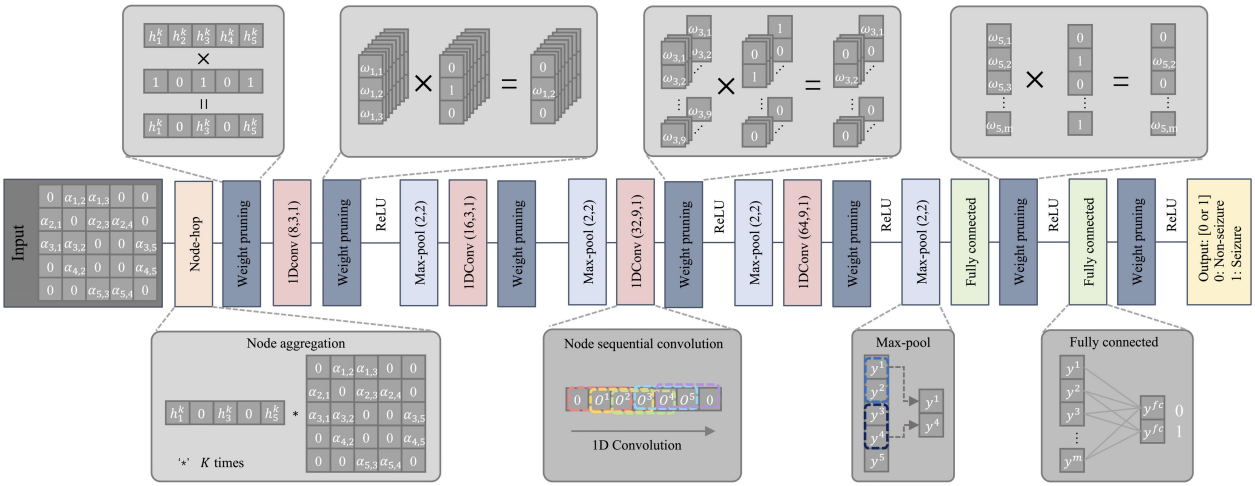

Fig. 3. Framework of SSGCNet consists of the following main parts: node aggregation, node sequential convolution, fully connected layer, and weight pruning. Note that we only perform weight pruning operations on the node sequential convolution layer and the fully connected layer.

weight pruning method to optimize the structure of the model, which will be discussed in Section IV-C.

The graph $\mathcal{G}$ with n nodes is equivalent to the input signal containing n data points. We first obtain the primary aggregation vector of each node and its neighbor nodes through the product of the vector $\mathbf{h^k} = \{h^k_{v_1}, h^k_{v_2}, h^k_{v_3}, \ldots, h^k_{v_n}\}$. In particular, when $k = 0$, the vector $\mathbf{h}$ is an all-ones vector. With performing the k-hops, we then have

$$h^k_{v_i} = h^{k-1}_{v_i} + \sum_{\{u_i\}} h^{k-1}_{u_i} \tag{7}$$

where v_i denotes the ith node of the graph $\mathcal{G}$, and u_i denotes the set of all the neighborhood nodes of v_i.

1) Node Aggregation: To aggregate the node information of directional graph, we use the vector $\mathbf{h}$ and the adjacency matrix $\mathbf{A}$ to multiply k times

$$\mathbf{h}^k = \mathbf{h}^{k-1} \cdot \mathbf{A} \tag{8}$$

where $\mathbf{h}^k$ denotes the value of $\mathbf{h}$ at the kth iteration. At the kth time, the relative position of each node in the vector remains unchanged.

2) Node Sequential Convolution: After obtaining the node aggregation vector, $h^k_{v_i}$ is multiplied by each element of the learnable parameter $\boldsymbol{\theta} = \{\theta^1, \theta^2, \ldots, \theta^n\}$, given by

$$o^i = \theta^i \cdot h^k_{v_i} \tag{9}$$

which effectively weighs the importance of different node information in backpropagation. Note that the operation does not change the node order of the aggregation vector. We can prune the learnable vector to reduce aggregated vectors.

We introduce the sequential convolution structure represented by sequential graphs in SGCN to extract sequential features in the graph structure the output of the aggregation. The size of the convolution kernel is the receptive field of the node range. The 1-D convolution uses the operation of zero padding on both sides. The size of the output vector is consistent with the size of the input vector after the convolution is completed by

$$y^i = \sigma\left(\sum_{i=1}^{n} w^u_l \cdot o^{i+u+1} + b_l\right) \tag{10}$$

where $\sigma(\cdot)$ is the rectified linear unit (ReLU) activation function [49], u is the kernel size, w^u_l is the convolution kernel parameter in lth layer, and b_l is the bias in the lth layer.

3) Fully Connected Layer: We perform a max-pooling operation on the two results of the convolutional layer, expressed as follows:

$$y^i = \max\left(y^{2i-1},\ y^{2i}\right) \tag{11}$$

and then, we feed the signal to the fully connected layer by

$$y^{\text{fc}}_i = \sigma\left(\sum_{i=1}^{n} w_l \cdot y^i + b_l\right). \tag{12}$$

Here, y^{fc} is the output of the SSGCNet. Note that we use Adam optimizer to optimize the model [50], and the function f is a cross-entropy function, defined by

$$f(\mathbf{w}_{1:N}, \mathbf{y}) = -\log\left(\frac{\exp^{y^{\text{fc}}_i}}{\sum_{i=1}^{c} \exp^{y^{\text{fc}}_i}}\right). \tag{13}$$

To date, we can obtain the output $\hat{\mathbf{y}}$ by computing (13). Our model is suitable for signals that can be well represented in graph nodes. However, as the scale of graph nodes and the computational complexity of the networks increase, the methods lack strong performance guarantee. In the following, we leverage the sparse redundancy in the number of weights of SSGCNet.

C. ADMM Weight Pruning Method

Using the ADMM-type splitting method [42], we introduce auxiliary variables $\mathbf{z}_1, \ldots, \mathbf{z}_N$ and an indicator function $g(\cdot)$. The constrained problem (6) can be rewritten as follows:

$$\begin{aligned} &\min_{\mathbf{w}_{1:N}} \left[f\left(\mathbf{w}_{1:N}, \mathbf{y}\right) + \sum_{l=1}^{N} g(\mathbf{z}_l) \right] \\ &\text{s.t. } \mathbf{z}_l = \Omega \mathbf{w}_l, \quad l = 1, \ldots, N \end{aligned} \tag{14}$$

where the indicator function is defined by

$$g(\mathbf{z}) = \begin{cases} \mathbf{0}, & \mathbf{card}(\mathbf{z}) \leq \text{const} \\ \infty, & \text{otherwise.} \end{cases} \tag{15}$$

The augmented Lagrangian function associated with (14) is formulated as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{w}_{1:N}, \mathbf{z}_{1:N}; \boldsymbol{\eta}_{1:N}) = & f\left(\mathbf{w}_{1:N}, \mathbf{y}\right) + \sum_{l=1}^{N} g(\mathbf{z}_l) \\ & + \sum_{l=1}^{N} \boldsymbol{\eta}_l^{\top} (\mathbf{z}_l - \Omega \mathbf{w}_l) \\ & + \sum_{l=1}^{N} \frac{\rho}{2} \|\mathbf{z}_l - \Omega \mathbf{w}_l\|_2^2 \end{aligned} \tag{16}$$

where $\boldsymbol{\eta}_l$ is the Lagrange multiplier, $\mathbf{z}_{1:N} = \text{vec}(\mathbf{z}_1, \ldots, \mathbf{z}_N)$, $\boldsymbol{\eta}_{1:N} = \text{vec}(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_N)$, and ρ is a penalty parameter. Starting at $\mathbf{w}_{1:N} = \mathbf{w}_{1:N}^{(k)}$, $\mathbf{z}_{1:N} = \mathbf{z}_{1:N}^{(k)}$, and $\boldsymbol{\eta}_{1:N} = \boldsymbol{\eta}_{1:N}^{(k)}$, the iteration steps of ADMM become

$$\mathbf{w}_{1:N}^{(k+1)} = \arg\min_{\mathbf{w}_{1:N}} \mathcal{L}\left(\mathbf{w}_{1:N}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) \tag{17a}$$

$$\mathbf{z}_l^{(k+1)} = \arg\min_{\mathbf{z}_l} \mathcal{L}\left(\mathbf{w}_l^{(k+1)}, \mathbf{z}_l; \boldsymbol{\eta}_l^{(k)}\right) \tag{17b}$$

$$\boldsymbol{\eta}_l^{(k+1)} = \boldsymbol{\eta}_l^{(k)} + \rho\left(\mathbf{z}_l^{(k+1)} - \Omega \mathbf{w}_l^{(k+1)}\right) \tag{17c}$$

which objective is to find a stationary point $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\eta}})$. The $\mathbf{z}_l$ and $\boldsymbol{\eta}_l$ subproblems compute the updates for each layer. The main benefit of combing with ADMM is that each auxiliary variable $\mathbf{z}_l$ can split the nonconvex sets arising in the weight pruning problems.

We write the subproblem in (17a) as follows:

$$\begin{aligned} \mathbf{w}_{1:N}^{(k+1)} = & \arg\min_{\mathbf{w}_{1:N}} f(\mathbf{w}_{1:N}, \mathbf{y}) \\ & + \sum_{l=1}^{N} \left(\boldsymbol{\eta}_l^{(k)}\right)^{\top} \left(\mathbf{z}_l^{(k)} - \Omega \mathbf{w}_l\right) \\ & + \sum_{l=1}^{N} \frac{\rho}{2} \left\|\mathbf{z}_l^{(k)} - \Omega \mathbf{w}_l\right\|_2^2. \end{aligned} \tag{18}$$

The function f is the cross-entropy function [see (13)], and the second term is a square of L_2 regularization. Thus, the sum of both functions is differentiable. We then solve (18) by computing the lth layer weight

$$\mathbf{w}_l^{(k+1)} = \begin{cases} \left(\dfrac{\exp^{y^i}}{\sum_{i=1}^{n} \exp^{y^i}} - \mathbf{I}\right)\mathbf{y} - \rho\Omega\left(\mathbf{z}_l^{(k)} - \Omega \mathbf{w}_l + \dfrac{\boldsymbol{\eta}_l^{(k)}}{\rho}\right), & i = fc \\ \dfrac{\exp^{y^i}}{\sum_{i=1}^{n} \exp^{y^i}}\mathbf{y} - \rho\Omega(\mathbf{z}_l^{(k)} - \Omega \mathbf{w}_l + \dfrac{\boldsymbol{\eta}_l^{(k)}}{\rho}), & i \neq fc. \end{cases} \tag{19}$$

For solving each $\mathbf{z}_l$-subproblem, we write

$$\begin{aligned} \mathbf{z}_l^{(k+1)} = & \arg\min_{\mathbf{z}_l} g(\mathbf{z}_l) + (\boldsymbol{\eta}_l^{(k)})^{\top}\left(\mathbf{z}_l - \Omega \mathbf{w}_l^{(k+1)}\right) \\ & + \frac{\rho}{2}\left\|\mathbf{z}_l - \Omega \mathbf{w}_l^{(k+1)}\right\|_2^2. \end{aligned} \tag{20}$$

Inspired by Boyd et al. [42], we can obtain

$$\mathbf{z}_l^{(k+1)} = \Pi\left(\Omega \mathbf{w}_l^{(k+1)} - \boldsymbol{\eta}_l^{(k+1)}/\rho\right) \tag{21}$$

where $\Pi(\cdot)$ is projection operator onto the constraint set $\{\mathbf{x} | \mathbf{card}(\mathbf{x}) \leq \text{const}\}$. Note that the parameter const controls the trade-off between least-squares errors and desired carnality.

After all the iteration, we can obtain the stationary point $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\eta}})$. Here, we can use other splitting methods, such as augmented Lagrangian splitting method and Peaceman–Rachford splitting [51], to compute the objective function (14). When the objective function (16) is convex, then the function globally converges to the global point $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\eta}})$ [42]. However, the function here is nonconvex. We need assumption conditions that ensure the convergence results, for example, proper choices of ρ and Ω [48]. In the following, we will prove the convergence results.

D. Convergence Analysis

Since pruning a deep neural network is nonconvex problem, even when both the loss function and constraints are convex, we will consider the nonconvexity in this article. We use the following assumptions for establishing the convergence.

Assumption 1: The function $f(\mathbf{x}, \mathbf{y})$ is prox-regular [52], [53] at $\mathbf{x}$ with constants $M > 0$. That is, for any $\mathbf{x}_1$ and $\mathbf{x}_2$ in a neighborhood of $\mathbf{x}$, there exists M, such that

$$\begin{aligned} & f(\mathbf{x}_1, \mathbf{y}) - f(\mathbf{x}_2, \mathbf{y}) \\ & \quad \geq -\frac{M}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2 + \langle \partial_{\mathbf{x}} f(\mathbf{x}_2, \mathbf{y}), \mathbf{x}_1 - \mathbf{x}_2 \rangle. \end{aligned} \tag{22}$$

Assumption 2: Ω is full-column rank with

$$\Omega\Omega^{\top} \succeq \kappa^2 \mathbf{I}.$$

Assumption 1 can be used to bound the partial of the loss function $f(\mathbf{w}_{1:N}, \mathbf{y})$. According to Assumption 1, we have the following equations:

$$\begin{aligned} & f(\mathbf{w}_{1:N}^{(k)}, \mathbf{y}) - f(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}) \\ & \geq -\frac{M_w}{2}\left\|\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right\|^2 \\ & \quad + \left\langle \partial_{\mathbf{w}} f(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}), \mathbf{w}_{1:N}^{(k)} - \mathbf{w}_{1:N}^{(k+1)} \right\rangle \end{aligned} \tag{23}$$

where M_w is a positive parameter. Due to the optimality condition on $\mathcal{L}(\mathbf{w}_{1:N}, \mathbf{z}_{1:N}; \boldsymbol{\eta}_{1:N})$, we can obtain

$$\partial_{\mathbf{w}} \mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) = 0 \tag{24}$$

which implies that

$$\partial f_{\mathbf{w}}(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}) = \Omega_{1:N}^{\top} \boldsymbol{\eta}_{1:N}^{(k)} + \rho \Omega_{1:N}^{\top}\left(\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N} \mathbf{w}_{1:N}^{(k+1)}\right) \tag{25}$$

where $\Omega_{1:N} = \text{blkdiag}(\Omega, \Omega, \ldots, \Omega)$ with the block diagonal matrix operator $\text{blkdiag}(\cdot)$. Now, we are ready to prove that the sequence $\mathcal{L}(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)})$ is monotonically nonincreasing when updating the variables $\mathbf{w}_{1:N}$.

Lemma 1: Let Assumptions 1 and 2 be satisfied. Also, let $\{\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\}$ be the iterative sequence. If $\rho \geq {}^{M_w}/_{\kappa^2}$, then the sequence $\mathcal{L}(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)})$ is nonincreasing with the updates of $\mathbf{w}_{1:N}$.

Proof: See the Appendix. ∎

Lemma 2: Let assumptions of Lemma 1 be satisfied. The sequence $\mathcal{L}(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)})$ is monotonically nonincreasing.

Proof: For the $\boldsymbol{\eta}_{1:N}$-subproblem, we obtain

$$\begin{aligned}&\mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k+1)}; \boldsymbol{\eta}_{1:N}^{(k+1)}\right) - \mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k+1)}; \boldsymbol{\eta}_{1:N}^{(k)}\right)\\ &= \left\langle \boldsymbol{\eta}_{1:N}^{(k+1)} - \boldsymbol{\eta}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k+1)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)} \right\rangle\\ &= \frac{1}{\rho}\left\| \boldsymbol{\eta}_{1:N}^{(k+1)} - \boldsymbol{\eta}_{1:N}^{(k)} \right\|^2. \end{aligned} \tag{26}$$

Using Lemma 1, we have

$$\begin{aligned}&\mathcal{L}\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) - \mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k+1)}; \boldsymbol{\eta}_{1:N}^{(k+1)}\right)\\ &\geq \frac{\rho\kappa^2 - M_w}{2}\left\| \mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)} \right\|^2 + \frac{1}{\rho}\left\| \boldsymbol{\eta}_{1:N}^{(k+1)} - \boldsymbol{\eta}_{1:N}^{(k)} \right\|^2. \end{aligned} \tag{27}$$

Since the condition $\rho \geq {}^{M_w}/_{\kappa^2}$ is satisfied, we write

$$\mathcal{L}\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) - \mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k+1)}; \boldsymbol{\eta}_{1:N}^{(k+1)}\right) \geq 0. \tag{28}$$

∎

Based on Lemma 2, we can now establish the convergence results in the following theorem.

Theorem 1: Let assumptions of Lemmas 2 be satisfied. Then, the sequence $\{\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}, \boldsymbol{\eta}_{1:N}^{(k)}\}$ generated by the ADMM weight pruning algorithm converges to a local minimum $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\eta}})$.

Proof: By Lemmas 1 and 2, the sequence $\mathcal{L}(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)})$ is monotonically nonincreasing, since $\mathcal{L}(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)})$ is upper bounded by $\mathcal{L}(\mathbf{w}_{1:N}^{(0)}, \mathbf{z}_{1:N}^{(0)}; \boldsymbol{\eta}_{1:N}^{(0)})$ and lower bounded by $f(\mathbf{w}_{1:N}^{(k)}, \mathbf{y})$. We assume there exists a local minimum $\hat{\mathbf{w}}$, such that the sequence $\{\mathbf{w}_{1:N}^{(k)}\}$ converges to $\hat{\mathbf{w}}$, which is a local minimum of $\mathbf{w}_{1:N}$-subproblem. We deduce the iterative sequence $\{\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}, \boldsymbol{\eta}_{1:N}^{(k)}\}$ generated by the algorithm that is locally convergent to $(\hat{\mathbf{w}}, \hat{\mathbf{z}}, \hat{\boldsymbol{\eta}})$. ∎

E. Summary

To explicitly utilize the graph structure of SSGCNet, we transform the input signals into the frequency domain and propose the WNFG method to represent the signals. Note that the EEG signal segment should be short enough to make that the signal is stationary within the segment. The duration of the epileptiform wave is the cause of the instability of the signal, so the epileptiform wave cannot change significantly during a segment. The length of a segment should be less than the duration of an epileptic seizure waveform, which will be discussed in Section V-B.

Based on the WNFG method, we propose the SSGCNet model, including node aggregation, node sequential convolution, and fully connected layer. We also develop a lightweighted version of the model by using the ADMM weight pruning method. At each epoch k_e, we execute the training and pruning operations. Our SSGCNet model offers competitive accuracy with faster speed and less memory allocation. The main steps of our model are summarized in Algorithm 1.

Algorithm 1 SSGCNet

Input: weights $\mathbf{w}_{1:N}$, signal $\mathbf{x}$, auxiliary variables $\mathbf{z}_{1:N}$ and $\boldsymbol{\eta}_{1:N}$, parameter ρ, parameters $\boldsymbol{\theta}_{1:n}$

Output: $\hat{\mathbf{w}}$, $\hat{\mathbf{y}}$.

```
compute graph representation y by (1) and (5);
for epoch k_e ≤ K_{e_max} do
    compute the node aggregation {o^i} by (9);
    compute the node sequential convolution layer {y^i} by (10);
    while not convergent do
        compute w_{1:N} by (19);
        compute z_{1:N} by (21);
        compute η_{1:N} by (17c);
    end
    compute the fully connected layer {y^fc} by (12);
    while not convergent do
        compute w_{1:N} by (19);
        compute z_{1:N} by (21);
        compute η_{1:N} by (17c);
    end
    return ŷ = [y_1^fc, ..., y_{n_y}^fc] and ŵ.
end
```

V. Experiments

In this section, we evaluate our method on the Bonn dataset [54] and the spikes and slow waves (SSW) dataset [29]. We conduct experiments on graph representation, model classification performance, and pruning methods to estimate the performance of our proposed method in three parts. All our source code can be found at https://github.com/anonymous2020-source-code/WNFG-SSGCNet-ADMM.

A. Datasets

In this article, we evaluate our method on two epileptic EEG signal datasets, namely, the Bonn dataset and the SSW dataset.

1) Bonn Dataset: The Bonn dataset is a single-channel epileptic EEG signal dataset from Bonn University, Germany, which contains five subsets A–E. Subsets A and B are from five healthy subjects. Subsets A and B are collected from the normal EEG signal of the subject with eyes open and closed, respectively. Subsets C–E are from five patients with confirmed epilepsy, including reverse area, epilepsy lesion area, and epileptic seizures. Only subset E is the data of epileptic seizures. Each category is composed of 100 EEG signals containing 4097 data points. The sampling frequency is 173.6 Hz, and the total duration of each signal is 23.6 s. The dataset has been preprocessed, such as myoelectricity and power frequency interference. See [54] for details. We split the data into nonoverlapping samples of length 256; that is, each piece of data is divided into 16 segments, and each subset contains 1600 segments. The experiments we designed are difficult due to the reduced information contained in each segment. Since only E is epilepsy, according to the experimental settings of previous work, we set up a total of four experiments, namely, A versus E, B versus E, C versus E, and D versus E. Each experiment contained 3200 EEG

segments, of which 1600 are nonepileptic and 1600 are epileptic.

2) SSW Dataset: This dataset is a single-channel absence epileptic EEG signal dataset containing two categories (seizure data and nonseizure data). Absence epileptic seizures have no obvious symptoms of convulsions. Absence epilepsy is clinically diagnosed by identifying SSW, which are epileptic waveforms consisting of a spike followed by a slow wave. The sampling frequency is 200 Hz. Usually, the duration of an SSW was less than 1 s. See [29] for details. Therefore, we divide the signal into 1 s (containing 200 data points) and mark the signal by clinical experts from famous local hospitals. Data are collected from ten patients diagnosed with absence epilepsy. Each seizure EEG signal contains at least one complete SSW data. The total number of the SSW dataset is 20 946 EEG signal segments, of which 10 473 are non-SSW and 10 473 are SSW.

B. Experimental Settings

We implemented the experiment in PyTorch 1.10.1 with python3, which was tested on four Nvidia Titan XP GPUs. For comparison, all the methods in the following have the same experimental settings.

For the EEG signal segment, the duration of epileptiform waves is around 3 s to several hours [55]. Meanwhile, the EEG signal segment includes enough epileptiform vibration cycles, and the frequency can only be resolved by repeating enough vibration cycles. The fundamental frequency of epilepsy ranges from 2.5 to 4 Hz [56], which is 0.25 and 0.4 s when converted into cycles. Since a segment contains multiple cycles, it is generally at least greater than 0.4 s. In this article, we segment the single EEG signal from Bonn dataset into small segments of equal length with around 1.5 s (containing 256 data points without overlap). For the SSW dataset, we set 1 s as each segment. We test the following deep learning models used in EEG signal classification tasks.

1) *MLP [57]:* MLP is a multilayer perceptron that includes five fully connected layers. The numbers of the neuron are $[n, 128]$, $[128, 64]$, $[64, 32]$, $[32, 16]$, and $[16, 2]$, respectively. Here, n is the node number of the input graph. Compared with SSGCNet, the MLP model has no node aggregation module and no sequential convolution module. The fully connected layer is exactly the same as SSGCNet. The purpose is to evaluate the performance of a module designed for EEG graph representation in SSGCNet.
2) *GNN [20]:* GNN is a graph neural network that includes node aggregation and fully connected layers. The model includes node aggregation and five fully connected layers, and other parameter settings of the model are the same as those of MLP. Compared with SSGCNet, the GNN model has no sequential convolution module. In addition to the node aggregation module, the fully connected layer is exactly the same. The purpose is to evaluate the performance of sequential convolution modules in SSGCNet.
3) *1-D CNN [58]:* The 1-D CNN is a deep learning model that includes the 1-D convolutional layer, the max-pooling layer, and the fully connected layer. The convolutional layer has four layers, the max-pooling layer has four layers, and the fully connected layer has two layers. The convolution kernel size is [1, 3], its step size is 1, and the number of channels in each layer is 8, 16, 32, and 64, respectively. The size of the largest pooling layer is [1, 2], and the step size is 2. The dimensions of the fully connected layer are [64, 16] and [16, 2]. Compared with SSGCNet, the 1-D-CNN model has no node aggregation module. In addition to the sequential convolution module, the fully connected layer is exactly the same. The purpose is to evaluate the performance of sequential convolution modules in SSGCNet.
4) *Our SSGCNet:* SSGCNet includes node aggregation, 1-D convolution, and fully connected layers (see Fig. 3 for details). In the 1-D convolutional layer, the size of the convolution kernel is [1, 3], its step size is 1, and the number of channels in each layer is 8, 16, 32, and 64. The size of the largest pooling layer is [1, 2], and the step size is 2. Other settings are the same as those of GNN.

In MLP, GNN, 1-D-CNN models, other parameters, learning rate, experimental conditions, and so on are the same as parameters of SSGCNet. To transform the adjacency matrix into a 1-D vector, all models implement a node aggregation as in 8. The matrix Ω is an identity matrix, and the parameter ρ of the iteration is 0.1.

C. Generation Time and Space of WNFG

In this section, we evaluate the generation time, generation space, and the accuracy of the graph representation in the time and frequency domains, respectively. We conducted ten experiments with different connection rates of the connection rule (connection rates $= 1, 0.2, \ldots, 0.1$). The connection rate represents the proportion of the connection rule for each node. For instance, the connection rate is 0.1, and the node is only connected to other near field nodes with a maximum distance of 10% in the graph.

The connection rate is from 1 to 0.1, the time and frequency domains have the same decreasing trend in generation time. It can be seen from Fig. 4 that the generation time and space occupied by different datasets are reduced as follows.

1) The generation time reduces from 2.07 to 0.61 s on the bonn dataset and from 5.76 to 1.58 s on the SSW dataset (time domain). The generation time reduces from 2.06 to 0.63 s on the bonn dataset and from 6.17 to 2.10 s on the SSW dataset (frequency domain).
2) The generation space reduces from 870 261 to 136 511 kb on the bonn dataset and from 4 589 179 to 757 256 kb on the SSW dataset (time domain). The generation space reduces from 1 441 290 to 247 069 kb on the bonn dataset and from 5 375 560 to 918 378 kb on the SSW dataset (frequency domain).

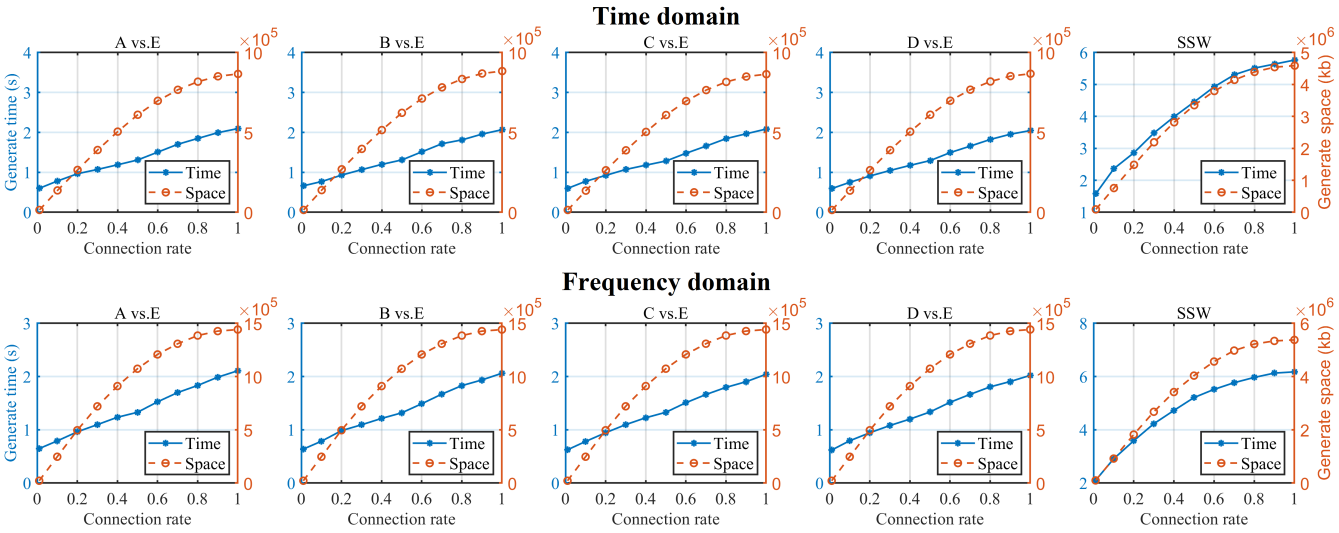

Fig. 4. Time and space overhead of WNFG in different connection rates.

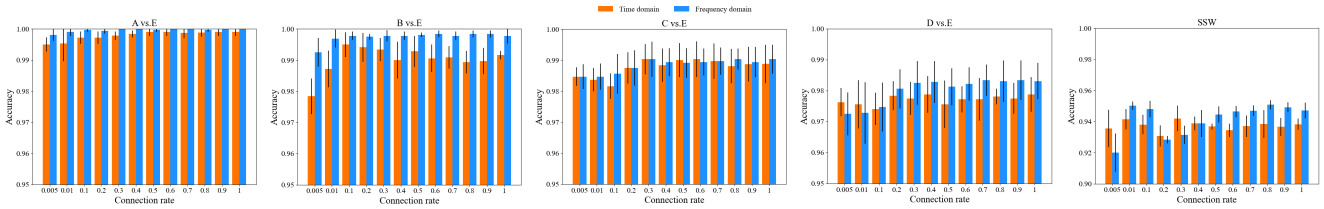

Fig. 5. Accuracy of different connection rates of WNFG in the Bonn and SSW datasets. The orange bar represents the time domain, and the blue bar represents the frequency domain.

TABLE II
PERFORMANCE OF DEEP LEARNING MODELS IN DIFFERENT GRAPH REPRESENTATIONS

Models	Domain	Bonn												SSW		
		A vs.E			B vs.E			C vs.E			D vs.E					
		Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen
MLP	Time	**1.000**	**1.000**	**1.000**	0.989	0.989	0.989	0.983	0.985	0.981	0.963	0.969	0.957	0.913	0.919	0.907
	Frequency	0.993	**1.000**	0.986	0.980	0.996	0.964	0.982	0.993	0.971	0.963	0.965	0.961	0.915	0.918	0.911
GNN	Time	0.999	0.999	0.999	0.990	0.985	0.995	0.979	0.978	0.980	0.960	0.954	0.967	0.891	0.868	0.920
	Frequency	0.995	**1.000**	0.991	0.984	0.992	0.976	0.984	0.983	0.985	0.967	0.966	0.969	0.903	0.896	0.917
1D-CNN	Time	**1.000**	**1.000**	0.999	0.988	0.988	0.988	0.976	0.971	0.981	0.955	0.962	0.947	0.906	0.914	0.897
	Frequency	0.997	0.999	0.996	0.995	0.997	0.992	0.982	0.994	0.979	0.964	0.976	0.962	0.911	**0.929**	0.893
SSGCNet	Time	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.997	**1.000**	0.994	0.978	0.978	0.979	0.916	0.920	0.912
	Frequency	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.999**	0.999	**0.998**	**0.983**	**0.987**	**0.979**	**0.924**	0.926	**0.923**

We also evaluate the classification performance of different connection rate graph representations. As shown in Fig. 5, when the connection rate is higher than 0.1, the classification performance of the graph representation does not decrease significantly. For example, the classification performance is even higher when the decrease in the connection rate on SSW is lower. The experiments show that reducing the number of connections in the graph representation does not lead to a significant drop in model performance. When the neighborhood rate is lower than 0.1, the classification accuracy tends to decrease, especially in the experiments of C versus E and D versus E. The experimental results show that when the graph representation neighborhood ratio is lower than 0.1, the information contained in the edges in the graph structure is limited, and it cannot effectively distinguish samples of different categories. In the classification accuracy of time and frequency domain graph representation, we can find that the classification accuracy of frequency domain graph representation is significantly higher than that of time domain graph representation in most cases. In our experiments, we choose the connection rate of graph representation to be 0.1.

D. Ablation Study of EEG Signal Classification Model

To evaluate the performance of the classification model proposed in this article, we implement ablation experiments to evaluate the effect of each module of the deep learning model. We test the MLP, GNN, 1-D-CNN, and SSGCNet deep learning models. As shown in Table II, we conduct experiments in the time and frequency domain WNFG with near field rate equal to 0.1. We set the maximum epoch to 30, which is

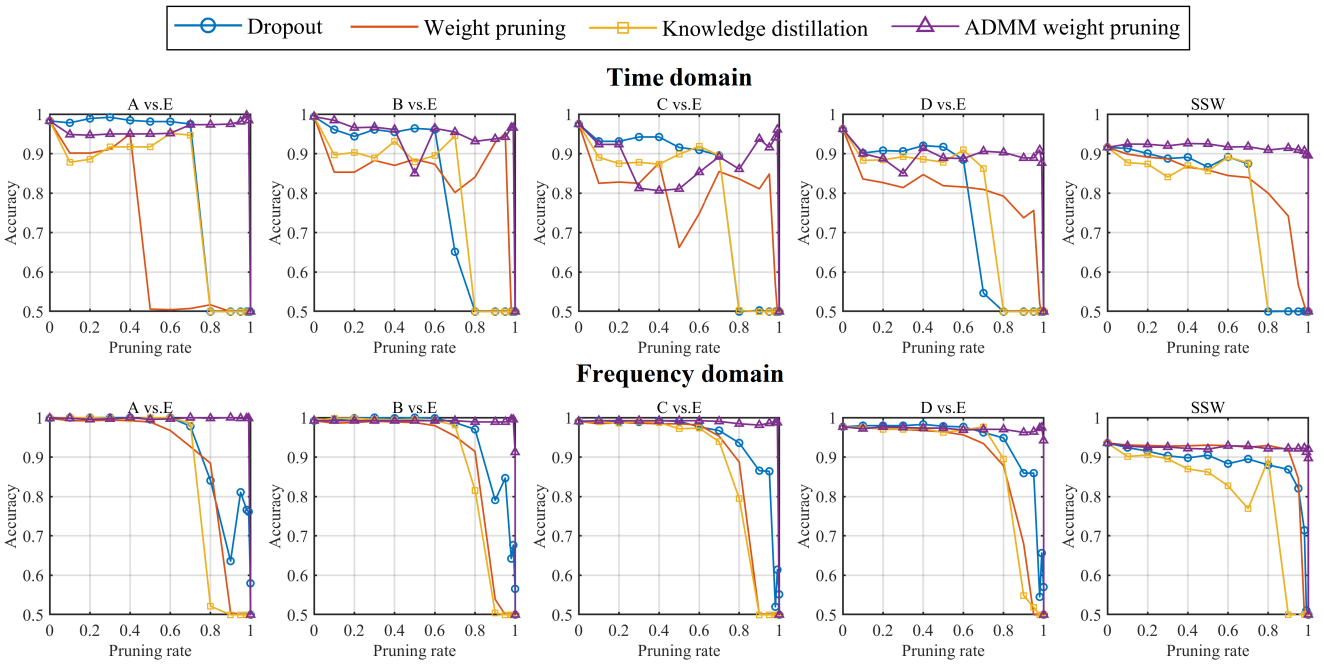

Fig. 6. Accuracy of the four pruning methods in the Bonn and SSW datasets. We use MLP as the baseline model.

because the loss of different baseline methods converged after 30 iterations. We use a fivefold cross validation to test the dataset. Each result comes from the best of 30 epochs, and we choose the five best results to average. In all models, the results of the frequency-domain WNFG are better than the results of the time-domain WNFG, indicating that the frequency-domain WNFG has better classification performance. The model with node aggregation module (GNN) or the model with sequential convolution module (1-D CNN) outperformed the model without node aggregation and sequential convolution module (MLP) in classification results. This shows that both the node aggregation module and the sequential convolution module can provide gains for the classification task of epileptic EEG graph representation. The model (SSGCNet) that includes both the node aggregation module and the sequential convolution module has the highest classification accuracy, which indicates that combining the above modules can further improve the classification performance. Hence, according to the experimental results, it is found that the frequency domain WNFG-SSGCNet provides the best results for epileptic EEG signal graph representation classification.

E. Performance of Different Weight Pruning Methods

To verify the performance of pruning methods in the EEG signal classification task, we compare the commonly used pruning strategies. To ensure that the effect of pruning is not disturbed by factors, such as model structure, we use an MLP model that only contains fully connected layers to experiment with all different pruning strategies. The MLP model consists of one input layer, two hidden layers, and one output layer. The maximum epoch $K_{e_{\max}}$ is 50, which is to ensure that the loss of all pruning methods converges. We use the fivefold cross validation to divide the dataset, and each result comes from the best of 50 epochs. We choose the average of the five best results as the final value. The input size is $[n, 256]$, where n is the length of the signal, and the number of hidden layer neurons is 256. The size of the hidden layer remains unchanged as [256, 256], and the size of the output layer is $[256, c]$ with the number of categories c.

TABLE III
NUMBER OF PARAMETERS ON DIFFERENT DEEP LEARNING MODELS WITH THE SAME ACCURACY

Layers of model	Deep learning models					
	GNN		2DCNN		SSGCNet	
	-	ADMM	-	ADMM	-	ADMM
Layer 1	32768	6554	72	1	24	3
Layer 2	8192	1639	1152	12	384	39
Layer 3	2048	410	41472	415	4608	461
Layer 4	512	103	82944	830	9216	922
Layer 5	32	7	524288	5243	32768	3277
Layer 6	-	-	128	2	128	13
Non-train	754	754	154	154	154	154
Total	44306	9467	650210	6657	47282	4869

We compare four pruning strategies, namely, dropout [30], weight pruning [44], knowledge distillation [32], and our ADMM weight pruning. The dropout method randomly resets the weights to zero, thereby reducing the number of model connections. The weight pruning method adopts the removal of smaller weights, thereby reducing the overfitting of the model. The knowledge distillation method improves the classification performance of the small model by using the small-scaled student model to learn the features extracted by the teacher model.

For each pruning strategy, we set the pruning rate to 0 (model compression 0 times), 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.98, 0.99 (model compression 100 times), and 0.999 (model compression 1000 times). As shown in Fig. 6, the classification accuracy represented by the frequency

TABLE IV
PERFORMANCE OF DIFFERENT METHODS

Methods	Complexity of Graph	Number of Parameters	Operation of multiplication	Bonn												SSW		
				A vs.E			B vs.E			C vs.E			D vs.E					
				Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen	Acc	Spe	Sen
CT-LS-SVM [59]	-	-	-	**1.000**	**1.000**	**1.000**	0.995	0.992	0.998	0.964	0.948	0.980	0.940	0.820	0.940	-	-	-
LSTM [60]	-	-	-	0.970	0.980	0.960	0.925	0.940	0.910	0.920	0.950	0.890	0.910	0.870	0.950	-	-	-
VG-SVMKNN [61]	$O(n^3)$	-	-	**1.000**	**1.000**	**1.000**	0.973	0.952	0.995	0.983	**0.980**	0.985	0.933	0.963	0.906	-	-	-
VG-GNN [23]	$O(n^3)$	44306	43552	0.997	0.999	0.994	0.975	0.987	0.962	0.963	0.969	0.956	0.916	0.900	0.931	0.837	0.724	**0.949**
LPVG-GNN [25]	$O(n^3)$	44306	43552	0.978	0.962	0.994	0.969	0.956	0.981	0.963	0.975	0.950	0.906	0.856	0.956	0.864	0.818	0.910
HVG-GNN [24]	$O(n \log n)$	44306	43552	0.988	0.987	0.988	0.916	0.912	0.919	0.972	0.950	0.994	0.925	**0.969**	0.881	0.873	0.847	0.898
LPHVG-GNN [26]	$O(n \log n)$	44306	43552	0.994	0.994	0.994	0.959	0.950	0.969	0.966	0.969	0.962	0.922	0.940	0.922	0.862	0.837	0.887
WOG-GNN [29]	$O(n^2)$	44306	43552	0.509	**1.000**	0.000	0.513	0.025	**1.000**	0.503	**0.991**	0.016	0.500	0.000	**1.000**	0.500	0.000	**1.000**
WNFG-GNN-ADMM	$O(n)$	9467	8713	**1.000**	**1.000**	**1.000**	**0.997**	**0.997**	0.997	**0.984**	0.969	**1.000**	**0.972**	0.963	0.981	**0.914**	**0.975**	0.853
WOG-2DCNN [29]	$O(n^2)$	650210	278429824	**1.000**	**1.000**	**1.000**	0.995	0.997	0.994	0.995	0.994	0.997	0.978	0.981	0.975	0.965	0.974	**0.960**
WOG-2DCNN-ADMM	$O(n^2)$	6657	2817149	**1.000**	**1.000**	**1.000**	0.997	0.997	0.997	0.992	0.988	0.997	0.967	0.963	0.972	0.909	0.975	0.842
WNFG-2DCNN-ADMM	$O(n)$	6657	2817149	0.998	0.997	**1.000**	0.997	0.997	0.997	0.992	0.991	0.994	0.967	0.963	0.972	0.910	0.978	0.842
WOG-SSGCNet-ADMM	$O(n^2)$	4869	68058	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.988	0.975	**1.000**	0.969	0.984	0.953	0.911	**0.989**	0.832
LPVG-SGCN [62]	$O(n^3)$	47282	678016	0.992	**1.000**	0.984	0.988	0.997	0.978	0.966	0.969	0.963	0.939	0.947	0.931	0.802	0.759	0.843
LPVG-SSGCNet-ADMM	$O(n^3)$	4869	68058	0.991	**1.000**	0.981	0.986	0.984	0.988	0.964	0.975	0.953	0.956	0.963	0.950	0.768	0.693	0.844
WNFG-SSGCNet	$O(n)$	47282	678016	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.999**	**0.999**	0.998	**0.983**	0.987	0.979	**0.970**	0.984	0.956
WNFG-SSGCNet-ADMM-Dropout	$O(n)$	4869	68058	0.998	**1.000**	0.996	**1.000**	**1.000**	**1.000**	0.986	0.988	0.984	0.967	**0.997**	0.938	0.948	0.977	0.920
WNFG-SSGCNet-ADMM	$O(n)$	4869	68058	0.998	0.997	**1.000**	**1.000**	**1.000**	**1.000**	0.988	0.984	0.991	0.977	0.972	**0.981**	0.951	0.984	0.918

domain graph is higher in all pruning strategy. As the pruning rate increases, the frequency domain graph representation classification is more stable than the time domain methods. For example, in the time domain graph representation, when the pruning rate reaches 0.8, the classification accuracy of most methods becomes 50% (classification failure). However, most methods in the frequency-domain graph representation do not fail until the pruning rate reaches 0.9. This shows that the frequency domain graph representation can effectively provide classification information in the case of a higher pruning rate for model optimization pruning. All other pruning methods fail at a pruning rate of 0.95 (model compression 20 times), while the ADMM method can maintain effective classification performance at a pruning rate of 0.99 (model compression 100 times). Even in the experiments of A versus E, C versus E, D versus E, and SSW, the ADMM method still maintains effective classification performance at 0.999 (model compression 1000 times). This shows that the ADMM weight pruning method has excellent model compression performance in the classification task of epileptic EEG signal frequency domain graph representation.

F. Comparison With the Existing Methods

To further verify the performance of the proposed method, we compare our method with existing epilepsy classification methods. Specifically, it includes comparison with other graph representation methods (see Table I) and other EEG graph representation classification models (see Table III).

In Table I, compared with the existing graph representation methods, the time complexity of our graph representation is lowest. In Table IV, our model has higher classification accuracy, which compared with other existing graph representations. This demonstrate that our graph representation not only has efficient EEG representation capability (with small generation time and space) but also has better classification performance.

From Table III, we can find that the ADMM method of GNN and SSGCNet can compress the model by ten times without degrading the classification accuracy, especially the 2-DCNN model can be compressed by 97 times.

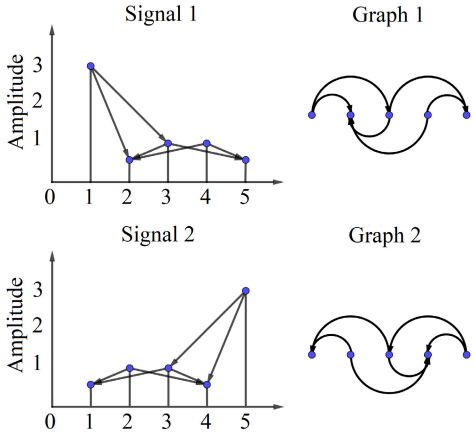

Fig. 7. Permutation variance of the graph is a main difference from traditional graph classification task.

The WNFG-SSGCNet-ADMM framework is the most lightweighted model while retaining a high level of classification accuracy.

VI. DISCUSSION

A. WNFG Representation

In this article, our EEG graph representation has two distinct differences from the traditional graph structure.

1) The WNFG graph is a directed graph, and its adjacency matrix is nondiagonally symmetric. It is difficult to aggregate nodes by constructing a Laplacian matrix on a diagonally symmetric matrix. As a result, the traditional node aggregation method will be incorrect for the graph representation studied in this article.
2) The sampling points of the EEG signals corresponding to the nodes represented by the graph have sequential characteristics in time order or frequency order. The sequence of the EEG signal sampling points changes, and then, the semantics of the EEG signals are completely different. As shown in Fig. 7, the topological structure of the two graphs is isomorphic. As a result, traditional graph classification models are not able to

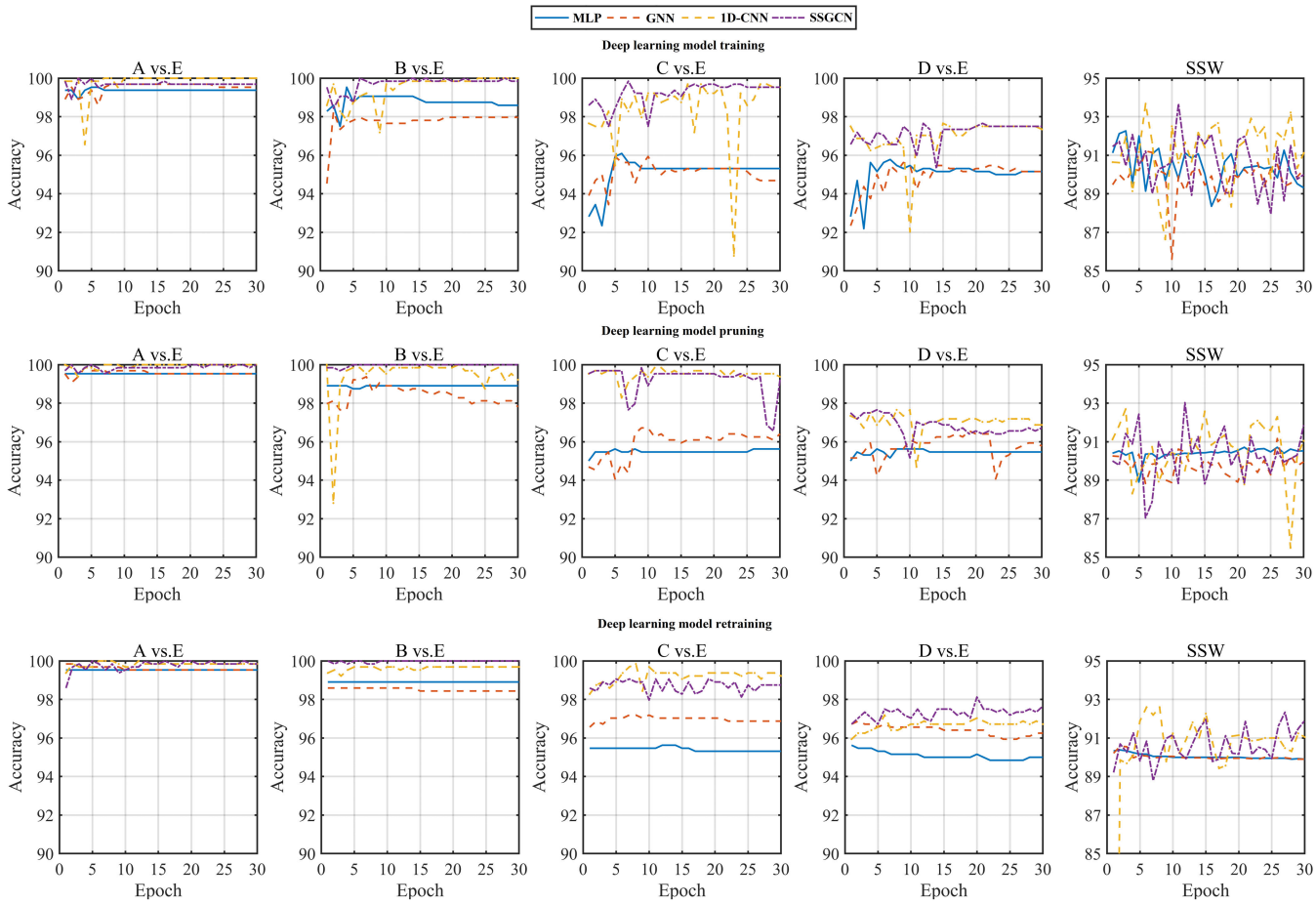

Fig. 8. Training curves of four deep learning models with ADMM weight pruning for four subsets—A versus E, B versus E, C versus E, and D versus E.

distinguish the graph representations with the same topology [63], [64], [65].

Hence, in the part of node aggregation, we design a learnable vector to perform matrix multiplication operation on the adjacency matrix. For the sequential features represented by the EEG graph, we introduce the sequential convolution structure in SGCN to extract the sequential features in the graph structure.

B. Optimization of ADMM Pruning Strategy

Here, we verify the universal applicability of ADMM-type splitting and weight pruning strategy in deep learning models. For the Bonn dataset and the SSW dataset, we select four different deep learning models, including MLP, GNN, 1-D CNN, and SSGCNet. We train the original model on both datasets, then use ADMM weight pruning to optimize the model, and retrain the pruned model. In this experiment, we uniformly set the model connection rate of different models to 0.1. The model structure is ten times less than the original model parameters.

As shown in Fig. 8, the training process of the original model has fluctuated. After the pruning is completed, the training process becomes stable. This shows that after subtracting part of the redundant structure, the training process of the model tends to be more stable than the original process. When the model connection rate is 0.1, we find

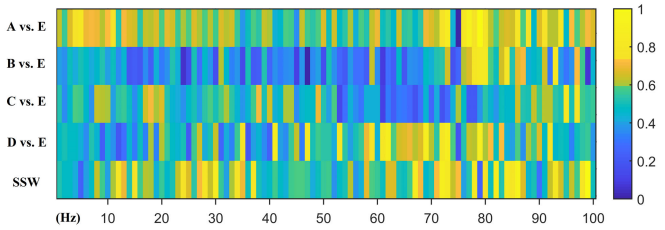

Fig. 9. Frequency domain learnable weight vectors of SSGCNet on all datasets, which can be interpreted as the importance of each frequency to epilepsy detection.

that the training process remains consistent with the unpruned situation. Even in some cases, the pruned model converges to a higher accuracy. Fig. 8 also shows the comparison results of the MLP, GNN, CNN, and SSGCNet models for Bonn and SSW datasets. Our weight pruning method achieves ten-time weight pruning on the deep learning models. This experiment demonstrates that the ADMM weight pruning method has universal applicability in different deep learning models, and it can maintain accuracy without reducing the number of model parameters.

C. Interpretability Analysis

Our interpretability comes from the learnable weight vector in Fig. 3. We convert the EEG signal as a graph representation, which is based on the relationship features among extracted vertices. It is helpful to understand the characteristics extracted

from EEG signals by our SSGCNet model. This also indirectly provides the explanation for the expected classification results. In Fig. 9, we visualize the weight vectors on all the datasets. The weights are generally higher on the frequency bands of 60–100 Hz, which correspond to gamma rhythm in EEG data stream. This is generally consistent with the medical domain studies [66], [67].

VII. Conclusion

In this article, we have introduced a SSGCNet method for epileptic EEG signal classification, which is based on ADMM-type splitting and weight pruning methods. We proposed an EEG signal graph representation method, a WNFG, which reduce the data generation time and memory usage. We then have introduced an ADMM weight pruning method for compressing redundancy in SSGCNet. Our method has achieved a model connection rate of up to ten times in both the Bonn and SSW datasets. Compared with other methods, our method has a lower computational cost and a smaller loss of classification accuracy.

Appendix
Proof of Lemma 1

For the $\mathbf{w}_{1:N}$-subproblem, we get

$$\begin{aligned}
&\mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) - \mathcal{L}\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) \\
&= f\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}\right) + g\left(\mathbf{z}_{1:N}^{(k)}\right) + \left(\boldsymbol{\eta}_{1:N}^{(k)}\right)^{\top}\left(\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\right) \\
&\quad + \frac{\rho}{2}\|\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\|_2^2 \\
&\quad - \left(f\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{y}\right) + g\left(\mathbf{z}_{1:N}^{(k)}\right) + \left(\boldsymbol{\eta}_{1:N}^{(k)}\right)^{\top}\left(\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\right)\right. \\
&\quad \left. + \frac{\rho}{2}\|\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\|_2^2\right) \\
&= f\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}\right) - f\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{y}\right) \\
&\quad + \left(\boldsymbol{\eta}_{1:N}^{(k)}\right)^{\top}\left(\Omega_{1:N}\mathbf{w}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\right) \\
&\quad + \frac{\rho}{2}\left\|\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\right\|_2^2 - \frac{\rho}{2}\left\|\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\right\|_2^2 \\
&= f\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}\right) - f\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{y}\right) + \frac{\rho}{2}\left\|\Omega_{1:N}\mathbf{w}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\right\|_2^2 \\
&\quad - \left\langle \boldsymbol{\eta}_{1:N}^{(k)} + \rho\left(\mathbf{z}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\right), \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\right\rangle \\
&= f\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}\right) - f\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{y}\right) + \frac{\rho}{2}\left\|\Omega_{1:N}\mathbf{w}_{1:N}^{(k)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)}\right\|_2^2 \\
&\quad - \left\langle \partial_{\mathbf{w}} f\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{y}\right), \mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right\rangle. \qquad (29)
\end{aligned}$$

We assume that Ω is full-column rank with $\Omega\Omega^{\top} \succeq \kappa^2\mathbf{I}$, and we have

$$\begin{aligned}
&\left\|\Omega_{1:N}\mathbf{w}_{1:N}^{(k+1)} - \Omega_{1:N}\mathbf{w}_{1:N}^{(k)}\right\|^2 \\
&\quad = \left\langle \Omega_{1:N}\left(\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right), \Omega_{1:N}\left(\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right)\right\rangle \\
&\quad = \left\langle \Omega_{1:N}\Omega_{1:N}^{\top}\left(\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right), \left(\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right)\right\rangle \\
&\quad \geq \kappa^2\left\|\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right\|^2. \qquad (30)
\end{aligned}$$

Combining (23), (29), and (30), we can obtain

$$\begin{aligned}
&\mathcal{L}\left(\mathbf{w}_{1:N}^{(k)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) - \mathcal{L}\left(\mathbf{w}_{1:N}^{(k+1)}, \mathbf{z}_{1:N}^{(k)}; \boldsymbol{\eta}_{1:N}^{(k)}\right) \\
&\quad \geq \frac{\rho\kappa^2 - M_w}{2}\left\|\mathbf{w}_{1:N}^{(k+1)} - \mathbf{w}_{1:N}^{(k)}\right\|^2. \qquad (31)
\end{aligned}$$

References

[1] G. Zhu, Y. Li, and P. Wen, "Epileptic seizure detection in EEGs signals using a fast weighted horizontal visibility algorithm," *Comput. Methods Programs Biomed.*, vol. 115, no. 2, pp. 64–75, 2014.

[2] Z. Mohammadpoory, M. Nasrolahzadeh, and J. Haddadnia, "Epileptic seizure detection in EEGs signals based on the weighted visibility graph entropy," *Seizure*, vol. 50, pp. 202–208, Aug. 2017.

[3] A. R. Hassan, S. Siuly, and Y. Zhang, "Epileptic seizure detection in EEG signals using tunable-Q factor wavelet transform and bootstrap aggregating," *Comput. Methods Programs Biomed.*, vol. 137, pp. 247–259, Dec. 2016.

[4] L. Wang, X. Long, J. B. A. M. Arends, and R. M. Aarts, "EEG analysis of seizure patterns using visibility graphs for detection of generalized seizures," *J. Neurosci. Methods*, vol. 290, pp. 85–94, Oct. 2017.

[5] Z. Gao et al., "EEG-based spatio–temporal convolutional neural network for driver fatigue evaluation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2755–2763, Sep. 2019.

[6] Y. Zou, R. V. Donner, N. Marwan, J. F. Donges, and J. Kurths, "Complex network approaches to nonlinear time series analysis," *Phys. Rep.*, vol. 787, pp. 1–97, Jan. 2019.

[7] S. Supriya, S. Siuly, H. Wang, and Y. Zhang, "EEG sleep stages analysis and classification based on weighed complex network features," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 2, pp. 236–246, Apr. 2021.

[8] Y. Yang, Z. Gao, Y. Li, Q. Cai, N. Marwan, and J. Kurths, "A complex network-based broad learning system for detecting driver fatigue from EEG signals," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 9, pp. 5800–5808, Sep. 2021.

[9] J. Iacovacci and L. Lacasa, "Visibility graphs for image processing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 974–987, Apr. 2020.

[10] V. Bajaj and R. B. Pachori, "Classification of seizure and nonseizure EEG signals using empirical mode decomposition," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1135–1142, Nov. 2012.

[11] A. Subasi, J. Kevric, and M. A. Canbaz, "Epileptic seizure detection using hybrid machine learning methods," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 317–325, 2017.

[12] E. Kabir and Y. Zhang, "Epileptic seizure detection from EEG signals using logistic model trees," *Brain Informat.*, vol. 3, no. 2, pp. 93–100, Jun. 2016.

[13] L. Cai et al., "Epileptic seizure detection using DWT based weighted visibility graph," in *Proc. 13th World Congr. Intell. Control Automat. (WCICA)*, Jul. 2018, pp. 739–743.

[14] M. Zhou et al., "Epileptic seizure detection based on EEG signals and CNN," *Frontiers Neuroinform.*, vol. 12, p. 95, Dec. 2018.

[15] R. Hussein, H. Palangi, R. Ward, and Z. Jane Wang, "Epileptic seizure detection: A deep learning approach," 2018, *arXiv:1803.09848*.

[16] U. Asif, S. Roy, J. Tang, and S. Harrer, "SeizureNet: Multi-spectral deep feature learning for seizure type classification," in *Machine Learning in Clinical Neuroimaging and Radiogenomics in Neuro-Oncology*. Cham, Switzerland: Springer, 2020, pp. 77–87.

[17] X. Wei, L. Zhou, Z. Chen, L. Zhang, and Y. Zhou, "Automatic seizure detection using three-dimensional CNN based on multi-channel EEG," *BMC Med. Informat. Decis. Making*, vol. 18, no. S5, pp. 71–80, Dec. 2018.

[18] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Comput. Biol. Med.*, vol. 100, pp. 270–278, Sep. 2017.

[19] C. Li and C. J. Shi, "Constrained optimization based low-rank approximation of deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 732–747.

[20] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–17.

[21] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–16.

[22] H. Wang, R. Feng, Z.-F. Han, and C.-S. Leung, "ADMM-based algorithm for training fault tolerant RBF networks and selecting centers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3870–3878, Aug. 2018.

[23] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuño, "From time series to complex networks: The visibility graph," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 13, pp. 4972–4975, 2008.

[24] B. Luque, L. Lacasa, F. Ballesteros, and J. Luque, "Horizontal visibility graphs: Exact results for random time series," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 4, Oct. 2009, Art. no. 046103.

[25] T. Zhou, N. Jin, Z. Gao, and Y. Luo, "Limited penetrable visibility graph for establishing complex network from time series," *Acta Phys. Sinica*, vol. 61, no. 3, p. 30506, 2012.

[26] Z.-K. Gao, Q. Cai, Y.-X. Yang, W.-D. Dang, and S.-S. Zhang, "Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series," *Sci. Rep.*, vol. 6, no. 1, p. 35622, Oct. 2016.

[27] M. Wang et al., "Exact results of the limited penetrable horizontal visibility graph associated to random time series and its application," *Sci. Rep.*, vol. 8, no. 1, p. 5130, 2018.

[28] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Apr. 2021.

[29] J. Wang et al., "A weighted overlook graph representation of EEG data for absence epilepsy detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 581–590.

[30] P. Baldi and P. J. Sadowski, "Understanding dropout," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Harveys, Lake Tahoe, Dec. 2013, pp. 2814–2822.

[31] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Peter Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*.

[32] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4133–4141.

[33] G. Wang et al., "Seizure prediction using directed transfer function and convolution neural network on intracranial EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 2711–2720, Dec. 2020.

[34] S. Mao and E. Sejdic, "A review of recurrent neural network-based methods in computational physiology," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 7, 2022, doi: 10.1109/TNNLS.2022.3145365.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[36] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," 2018, *arXiv:1810.05270*.

[37] D. Blalock, J. Javier Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" 2020, *arXiv:2003.03033*.

[38] A. See, M.-T. Luong, and C. D. Manning, "Compression of neural machine translation models via pruning," 2016, *arXiv:1606.09274*.

[39] Y. Zhang et al., "Improving EEG decoding via clustering-based multitask feature learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3587–3597, Aug. 2022.

[40] R. Gao, F. Tronarp, and S. Särkkä, "Iterated extended Kalman smoother-based variable splitting for L_1-regularized state estimation," *IEEE Trans. Signal Process.*, vol. 97, no. 19, pp. 5078–5092, Oct. 2019.

[41] R. Gao, S. Särkkä, R. Claveria-Vega, and S. Godsill, "Autonomous tracking and state estimation with generalized group lasso," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12056–12070, Nov. 2022.

[42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[43] R. Gao, F. Tronarp, and S. Sarkka, "Variable splitting methods for constrained state estimation in partially observed Markov processes," *IEEE Signal Process. Lett.*, vol. 27, pp. 1305–1309, 2020.

[44] T. Zhang et al., "A systematic DNN weight pruning framework using alternating direction method of multipliers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 184–199.

[45] S. Ye et al., "Progressive weight pruning of deep neural networks using ADMM," 2018, *arXiv:1810.07378*.

[46] X. Lai, J. Cao, X. Huang, T. Wang, and Z. Lin, "A maximally split and relaxed ADMM for regularized extreme learning machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1899–1913, Jun. 2020.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Harveys, Lake Tahoe, Dec. 2012, pp. 1097–1105.

[48] R. Gao, S. A. Vorobyov, and H. Zhao, "Image fusion with cosparse analysis operator," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 943–947, Jul. 2017.

[49] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Ft. Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[51] R. Glowinski, S. J. Osher, and W. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering*. Cham, Switzerland: Springer, 2017.

[52] R. T. Rockafellar and R. Wets, *Variational analysis*. Berlin, Germany: Springer-Verlag, 1998.

[53] R. A. Poliquin and R. T. Rockafellar, "Prox-regular functions in variational analysis," *Trans. Amer. Math. Soc.*, vol. 348, no. 5, pp. 1805–1838, May 1996.

[54] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, 2001, Art. no. 061907.

[55] T. A. Glauser et al., "Ethosuximide, valproic acid, and lamotrigine in childhood absence epilepsy," *New England J. Med.*, vol. 362, no. 9, pp. 790–799, 2010.

[56] V. Crunelli and N. Leresche, "Childhood absence epilepsy: Genes, channels, neurons and networks," *Nature Rev. Neurosci.*, vol. 3, no. 5, pp. 371–382, May 2002.

[57] H. Daoud and M. A. Bayoumi, "Efficient epileptic seizure prediction based on deep learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 804–813, Oct. 2019.

[58] M. T. Avcu, Z. Zhang, and D. W. Shih Chan, "Seizure detection using least eeg channels by deep convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1120–1124.

[59] S. Siuly, Y. Li, and P. Wen, "Clustering technique-based least square support vector machine for EEG signal classification," *Comput. Methods Programs Biomed.*, vol. 104, no. 3, pp. 358–372, 2011.

[60] D. Ahmedtaristizabal, C. Fookes, K. Nguyen, and S. Sridharan, "Deep classification of epileptic signals," 2018, *arXiv:1801.03610*.

[61] S. Supriya, S. Siuly, H. Wang, J. Cao, and Y. Zhang, "Weighted visibility graph with complex network features in the detection of epilepsy," *IEEE Access*, vol. 4, pp. 6554–6566, 2016.

[62] J. Wang, S. Liang, D. He, Y. Wang, Y. Wu, and Y. Zhang, "A sequential graph convolutional network with frequency-domain complex network of EEG signals for epilepsy detection," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2020, pp. 785–792.

[63] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[64] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[65] J. B. Estrach, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[66] J. O. Willoughby, "Persistent abnormality detected in the non-ictal electroencephalogram in primary generalised epilepsy," *J. Neurol., Neurosurg. Psychiatry*, vol. 74, no. 1, pp. 51–55, Jan. 2003.

[67] K. Benedek, A. Berényi, P. Gombkötő, H. Piilgaard, and M. Lauritzen, "Neocortical gamma oscillations in idiopathic generalized epilepsy," *Epilepsia*, vol. 57, no. 5, pp. 796–804, May 2016.

Jialin Wang received the Ph.D. degree in software engineering from Fudan University, Shanghai, China, in 2021.

He currently holds a post-doctoral position with the State Key Laboratory of ASIC and Systems, the Institute of Brain-Inspired Circuits and Systems, and the Zhangjiang Fudan International Innovation Center, Fudan University. His current research interests include data mining, machine learning, complex network, and EEG signal analysis.

Rui Gao (Member, IEEE) received the M.S. degree in computer software and theory from Northeastern University, Shenyang, China, in 2012, and the D.Sc. (Tech.) degree in automation, systems and control engineering from Aalto University, Espoo, Finland, in 2020.

She is currently an Assistant Professor with the Department of Naval Architecture and Ocean Engineering, Shanghai Jiao Tong University, Shanghai, China. Her research interests include machine learning algorithms, state estimation, and autonomous systems.

Haotian Zheng is currently pursuing the M.Sc. degree with the Academy for Microelectronics, the Institute of Brain-Inspired Circuits and Systems, and the Zhangjiang Fudan International Innovation Center, Fudan University, Shanghai, China.

His research interests include machine and deep learning, hardware acceleration architectures for deep learning, and computing model.

Hao Zhu is currently pursuing the Ph.D. degree with the Academy for Microelectronics, the Institute of Brain-Inspired Circuits and Systems, and the Zhangjiang Fudan International Innovation Center, Fudan University, Shanghai, China.

His research interests include reinforcement learning, obstacle avoidance, and navigation algorithms for robot.

C.-J. Richard Shi (Fellow, IEEE) has been a Professor in electrical and computer engineering with the University of Washington, Seattle, WA, USA, since 2004, where he joined in 1998. Since 2005, he has directed several sponsored research projects in the area of ADC, PLL, SerDes, and LDPC design. His current research interests include energy-efficient circuit and system design for sensing, computing, learning, and communication.

Dr. Shi worked in the area of computer-aided design of mixed-signal integrated circuits, in which for his contribution, he was elevated to a fellow of the IEEE in 2005. He received a prestigious Doctoral Prize from the Natural Science and Engineering Research Council (NSERC) of Canada and a Governor-General's Silver Medal in 1995 for his Ph.D. Dissertation in computer science. He received several awards for his research, including the Donald O. Pederson Best IEEE TRANSACTIONS ON CAD Paper Award, the Best Paper Awards from the IEEE/ACM Design Automation Conference, the IEEE VLSI Test Symposium, and the SRC Technical Conference, and an NSF CAREER Award.