Supplementary File for A Lightweight Multi-Scale Convolutional Neural Networks Based on Spatial-Spectral Feature Fusion for Interpretable EEG Decoding

This supplementary file consists of the following parts:
1) **The definitions of mathematical symbols**
2) **All acronyms and their full names**
3) **The interpretable algorithm for analysis of spatial-spectral patterns**
4) **The pseudo code of proposed LM-CNN model**

A. The definitions of mathematical symbols

For ease of reference, Table I provides a list of essential mathematical symbols used throughout the paper.

B. All acronyms and their full names

Table II provides all acronyms and their full names used in the paper.

C. The interpretable algorithm for analysis of spatial-spectral patterns

Based on the Eigen-CAM, we propose a spatial feature interpretability algorithm. For any branch of MSM, $\mathbf{d}_{j,i}$ is the signal of the i-th $(i = 1, 2, ..., I)$ trial at class j after spatial convolutional layer. The singular value decomposition can be expressed as follows.

$$\mathbf{d}_{j,i} = \mathbf{U}_{j,i}\mathbf{\Sigma}_{j,i}\mathbf{V}_{j,i}^{\mathrm{T}} \tag{1}$$

where $\mathbf{d}_{j,i} \in \mathbb{R}^{T\times D}$, T is the number of data samples and D is the number of spatial convolutional filters. $\mathbf{U} \in \mathbb{R}^{T\times T}$ is an orthogonal matrix of dimension T, where the column vectors of $\mathbf{U}$ are the left singular vectors. $\mathbf{\Sigma} \in \mathbb{R}^{T\times D}$ is a diagonal matrix. $\mathbf{V} \in \mathbb{R}^{D\times D}$ is an orthogonal matrix of dimension D, where the column vectors of $\mathbf{V}$ are the right singular vectors.

Then, the spatial weight matrix corresponding to the features of class j can be obtained

$$\mathbf{M}_j = \frac{1}{I}\frac{1}{D}\sum_{i=1}^{I}\sum_{d=1}^{D}\mathbf{S}\mathbf{v}_{d_{j,i}} \tag{2}$$

In (2), $\mathbf{v}_{d_{j,i}}$ is the data of $\mathbf{V}_{j,n}$ at column d. $\mathbf{S} \in \mathbb{R}^{K\times D}$ is the weight of the spatial convolutional kernel, K represents the number of signal electrodes. The matrix $\mathbf{M}_j \in \mathbb{R}^{K\times 1}$ represents the spatial weight matrix corresponding to class j. Mapping $\mathbf{M}_j$ onto the brain topography yields the visualization results.

D. The pseudo code of proposed LM-CNN model

The detailed training and evaluation procedures of proposed LM-CNN model are summarized in **Algorithms** 1 and 2.

TABLE I
LIST OF SYMBOLS IN THIS PAPER

Name	**Description**
T	sampling points of a single trial
K	number of electrodes
k	index value of $\{1, 2, ..., K\}$
$\mathbf{X} \in \mathbb{R}^{K\times T}$	data matrix of single-trial EEG data
$\mathbf{x}_k \in \mathbb{R}^{1\times T}$	kth row data of $\mathbf{X}$
σ	additive noise
$E \in \{2, 4\}$	class number
$y \in \{1, 2\}$ or $\{1, 2, 3, 4\}$	class label of $\mathbf{X}$
D	number of source components for $\mathbf{x}_k$
$\mathbf{w}_k \in \mathbb{R}^{1\times D}$	transfer vector of $\mathbf{x}_k$
$\mathbf{H'}_k \in \mathbb{R}^{T\times D}$	source components of $\boldsymbol{x}_k$
$\mathbf{X'}_k \in \mathbb{R}^{T\times(D+1)}$	data matrix by cascading $\mathbf{H'}_k$ with $\boldsymbol{x}_k$
$\mathcal{X} \in \mathbb{R}^{K\times T\times(D+1)}$	tensor data by concatenating $\mathbf{X}'_1, \mathbf{X}'_2, ..., \mathbf{X}'_K$
d	index value of $\in \{1, 2, ..., D+1\}$
$\mathbf{X}_d \in \mathbb{R}^{K\times T}$	d-th channel data of $\mathcal{X}$
$\mathbf{W}_{1,d} \in \mathbb{R}^{B_1\times M_1}$	convolution kernel used in MSM, with height B_1 and width M_1
$\mathbf{Z}_d \in \mathbb{R}^{K\times T}$	reuslt of depthwise convolution between $\mathbf{X}_d$ and $\mathbf{W}_{1,d}$
$\mathcal{Z} \in \mathbb{R}^{K\times T\times(D+1)}$	tensor data by concatenating $\mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_{D+1}$
N	number of Vanilla convolution kernel for $\mathbf{Z}_d$
n	index value of $\{1, 2, ..., N\}$
$\mathbf{W}_{2,n,d} \in \mathbb{R}^{B_2\times M_2}$	weights of the n-th Vanilla convolution kernel for $\mathbf{Z}_d$, with height B_2 and width M_2
$\mathbf{Y}_n \in \mathbb{R}^{K\times T}$	reuslt of Vanilla convolution between $\mathbf{Z}_d$ and $\mathbf{W}_{2,n,d}$
$\mathcal{Y} \in \mathbb{R}^{K\times T\times N}$	tensor data by concatenating $\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_N$
$\mathbf{F}' \in \mathbb{R}^{T\times N}$	result of spatial filtering for $\mathcal{Y}$ in each branch
$\mathbf{F}_{MSM} \in \mathbb{R}^{3N\times T}$	spatial-spectral feature representation of a single trial, by concatenating each $\mathbf{F}'$ from three branches
$\mathbf{F}_{in} \in \mathbb{R}^{3N\times T}$	ouput of a single trial data processed by MSM
$\mathbf{w}_{CA} \in \mathbb{R}^{3N\times 1}$	channel attention weights
$\mathbf{F}_{CA} \in \mathbb{R}^{3N\times T}$	output of WCAM
$\mathbf{f}_{ap} \in \mathbb{R}^{3N\times 1}$	average-aggregation feature of $\mathbf{F}_{in}$
$\mathbf{f}_{ap} \in \mathbb{R}^{3N\times 1}$	max-aggregation feature of $\mathbf{F}_{in}$
$\mathbf{w}_{CA}$	feature matrix concatenating $\mathbf{f}_{ap}$ and $\mathbf{f}_{mp}$
$\mathbf{f}_{CA} \in \mathbb{R}^{3NT}$	result of $\mathbf{F}_{CA}$ by average pooling and flattening
$\mathbf{W} \in \mathbb{R}^{E\times(3NT)}$	weight matrix for the softmax function computing probability distribution in classification module
$\mathbf{y}_b$	one-hot encoded true label for training data
$\{\mathbf{w}_k\}$	set of $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K\}$
$\{\mathbf{W}_{1,d}\}$	set of $\{\mathbf{W}_{1,1}, \mathbf{W}_{1,2}, ..., \mathbf{W}_{1,D}\}$
$\{\mathbf{W}_{2,n,d}\}$	set of $\{\mathbf{W}_{2,n,d}, n = 1, 2, ..., N, d = 1, 2, ..., D+1\}$
$\{\mathbf{W}_{3,d}\}$	all weight matrices used in depthwise convolution for spatial filtering
$\mathbf{w}_{am} \in \mathbb{R}^{2\times 1}$	weight vector for aggregating and average-aggregation feature and max-aggregation feature
$\{\mathbf{W}_{4,j}\}$	weight matrix set of vanilla convolution in WCAM

TABLE II
LIST OF ACRONYMS IN THIS PAPER

Acronym	Full Name
DL	deep learning
BCIs	brain-computer interfaces
MI	motor imagery
EEG	electroencephalography
CSAM	cascade spatial attention module
MSM	multi-scale spatial-spectral module
WCAM	weighted channel attention module
ERD	event-related desynchronization
ERS	event-related synchronization
CSP	common spatial pattern
FBCSP	filter bank common spatial pattern
CNN	convolutional neural network
C2CM	channel-wise convolution with channel mixing network
ShallowNet	shallow network
EEGNet	electroencephalography network
MI-EEGNet	motor imagery-based electroencephalography network
MTFB-CNN	multi-scale time-frequency block convolutional neural network
EEG-ITNet	explainable inception temporal convolutional network
LMDA-Net	lightweight multi-dimensional attention network
ELU	exponential linear units
BN	batch normalization
ReLU	rectified linear unit

Algorithm 1 LM-CNN Training

Input: Training set $\{(\mathbf{X}_m, y_m)\}_{m=1}^{M}$, batch size B, number of epochs E_p, learning rate α.
Output: Optimized model parameters $\mathbf{W}$, $[\mathbf{w}_k]$, $\{\mathbf{W}_{1,d}\}$, $\{\mathbf{W}_{2,n,d}\}$, $\{\mathbf{W}_{3,d}\}$, and $\{\mathbf{W}_{4,j}\}$.

```
Initialize all model parameters.
for epoch = 1 to E_p do
    Shuffle the training set.
    for each batch {(X_b, y_b)}_{b=1}^{B} do
        Compute augmented signals using CSAM.
        Extract spatial-spectral features via MSM.
        Aggregate features using WCAM.
        Flatten features and compute predictions via the classification module.
        Compute the cross-entropy loss according to (??).
        Update model parameters using gradient descent.
    end for
end for
```

Algorithm 2 LM-CNN Evaluation

Input: An unlabeled trial $\mathbf{X}_{\text{test}}$, trained model parameters.
Output: Predicted class label y_{test}.

```
Compute augmented signal using CSAM.
Extract spatial-spectral features via MSM.
Aggregate features using WCAM.
Flatten features and compute the probability distribution via the classification module.
Predict the label as y_test = arg max_i (p_test).
```