EEG Deep Learning

GCNs-Net: A Graph Convolutional Neural Network Approach for Decoding Time-Resolved EEG Motor Imagery Signals

Yimin Hou [1], Shuyue Jia [1, 2], Xiangmin Lun [1], Ziqian Hao [3], Yan Shi [1], Yang Li [4], Rui Zeng [5], and Jinglei Lv [5]

[1] School of Automation Engineering, Northeast Electric Power University

[2] Department of Computer Science, City University of Hong Kong

[3] Jinan Vocational College

[4] School of Electrical Engineering, Northeast Electric Power University

[5] School of Biomedical Engineering and Brain and Mind Center, University of Sydney

EEG Deep Learning Library: https://github.com/SuperBruceJia/EEG-DL

Research Background

- **BCI**: establish connections between the brain and machines

  (1) **Acquire and analyze brain signals** while conducting actual or imagery tasks

  (2) **Control machines**

- **Significance**: help the disabled (*e.g.*, suffered from strokes) and understand our brain
- **Types of BCI**:
  - Electroencephalography (EEG)
  - Magnetoencephalography (MEG)
  - Functional Magnetic Resonance Imaging (fMRI)
  - Invasive BCI Technologies (*e.g.*, Neuralink)

- **Reasons for using EEG for this project**:
  - Non-invasiveness
  - High Temporal Resolution
  - Portability
  - Inexpensive Equipment

  A potential market

- **Specific Task**: EEG Motor Imagery Task (*e.g.*, control a wheelchair via brain signals)
- **Our Research**: develop EEG-based BCI applications to improve the current stroke rehabilitation strategies

Image Credit: in the public domain.

Key Points in dealing with EEG time series

- **Individual Variability → Lower Classification Accuracy**
  - Low SNR (Signal-to-noise Ratio)
  - Different brain electrical conductivity ← different anatomical structure of brain
  - Electrodes' position error
- **Slow Real-time Responding → Hard to develop Real-life applications**
  - [**most literature**] Trial-level prediction (*e.g.*, 4 s)
  - Window/Slide-level prediction (*e.g.*, 0.4 s)
  - Time-resolved prediction (*e.g.*, 6.25 ms) (Our Work)
- **Low Group-level Accuracy → Hard to develop Applications for a Group of People**
  - [**most literature**] Subject-level prediction (Our Work)
  - Group-level prediction (Our Work)

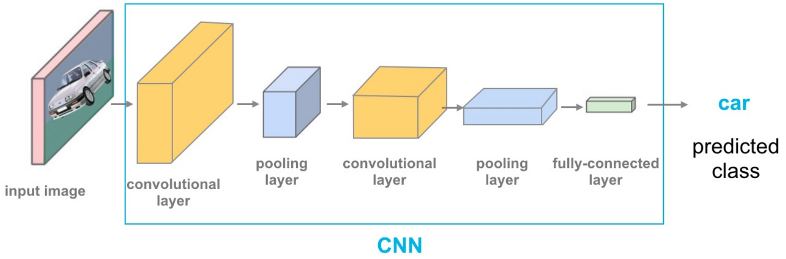
EEG
Feature Extraction

EEG Electrodes'
Structure Modeling

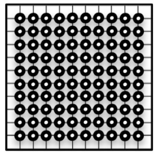
Time-resolved or Window-based
Signal Sampling

Motivation

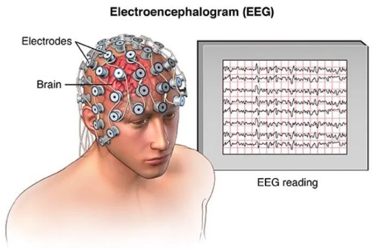
Convolutional Neural Network (CNN):

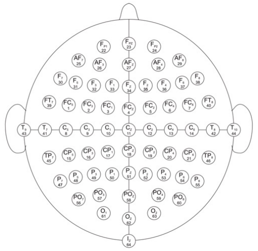


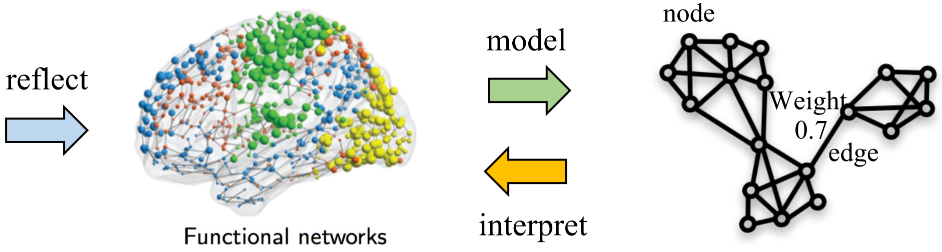
- **Module: Convolution → Pooling → Fully-connected**
- **Pros: Translation Equivalence, Translation Invariance, Weights Sharing**
- **Modeling: Euclidean-structured data** (e.g. Image, voice, natural languages)



- Neuroscience research has increasingly emphasized **Brain Network Dynamics**
  - Model **Functional Topological Connectivity** of EEG electrodes → **Graph** instead of Euclidean structure







Our Question

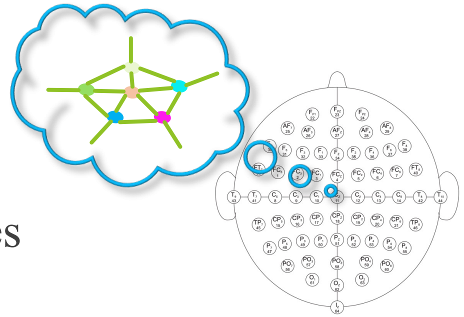
Can we model the EEG System as a **Graph**?

Can we process EEG systems via **Graph Representation Learning**?

Image Credit: The PhysioNet Dataset.

Q: Can we directly apply convolutions on graphs?

- Traditional CNN **cannot** directly process graph signals:
  - **Graphs are irregular** (*i.e.*, unordered and vary in size)
  - Convolutions **cannot keep translation invariance on the non-Euclidean signals**
- Graph Convolutional Neural Networks (GCN)
  - Can directly process non-Euclidean graph-structured signals
  - Consider the relationship properties (*e.g.*, correlations) between nodes

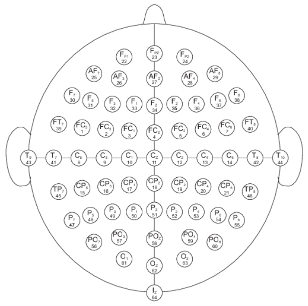


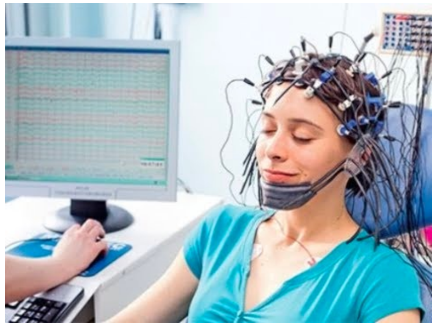
→ Model **Functional Topological Relationships** among EEG electrodes

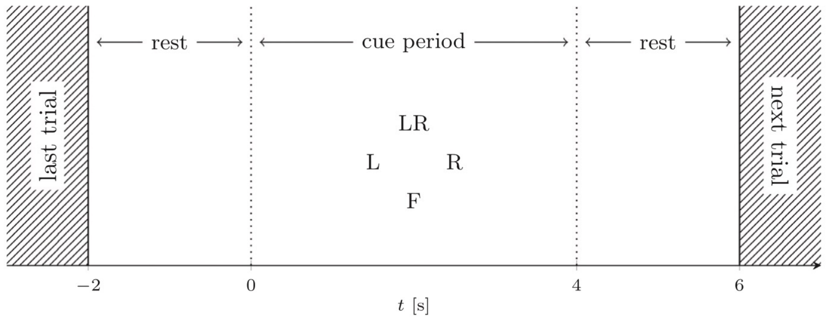
→ Model, Analyze and Interpret **Brain Network Dynamics**

Benchmark Dataset Description

- The PhysioNet Dataset (EEG Motor Movement/Imagery Dataset)
- International 10-10 EEG System, 64 electrodes (excluding electrodes Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, and P10)







- 109 subjects (**the largest number of participants** in the field of EEG Motor Imagery)
- Task: 4-class EEG Motor Imagery Classification
  - Imagining (1) left fist, (2) right fist, (3) both fists (4) both feet
- For each subject, 3 runs, 7 trials, 4 classes → 84 trials in total
- For each trial, 4 seconds experimental duration, 160 Hz Sampling Rate → 640 Time Points
- We applied Time-resolved Method for real-time Applications:
  - Total samples per subject: 3 runs × 7 trials × 4 classes × 4 seconds × 160 Hz = 53,760 samples
  - Experimental Settings: 90% as the training set and the left 10% as the test set

Image Credit: The PhysioNet Dataset.

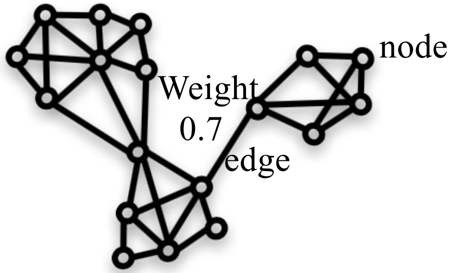
Preliminary: Graph Representation

Definition: An Undirected and Weighted Graph with N nodes: $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{A}\}$

– **V**: nodes (vertices), $|\mathbf{V}| = N$

– **E**: edges (links) that connect nodes

– **A**: weights / correlations between nodes



Nodes Correlations: Pearson Matrix $\mathbf{P} \in \mathrm{R}^{N\mathrm{x}N}$ (denote as PCC matrix)

– Measure the linear correlations between nodes **x** and **y**

– μ is the mean, σ is the standard deviation, and $P_{x,y}$ is the Pearson Correlation Coefficient between two nodes

$$P_{x,y} = \frac{\mathrm{E}((\mathbf{x} - \mu_x)(\mathbf{y} - \mu_y))}{\sigma_x \sigma_y}$$

– Absolute Pearson Matrix: $|\mathbf{P}| \in \mathrm{R}^{N\mathrm{x}N}$ and $|P_{ij}| \in [0, 1] \rightarrow$ Note: in this work, we only consider **scale**

Graph Weights: Adjacency Matrix $\mathbf{A} = |\mathbf{P}| - \mathbf{I} \in \mathrm{R}^{N\mathrm{x}N}$, where **I** is an Identity Matrix

Graph Degrees: Degree Matrix $\mathbf{D} \in \mathrm{R}^{N\mathrm{x}N}$

$$D_{ii} = \sum_{j=1}^{N} A_{ij}$$

Graph Representation: Combinatorial Laplacian $\mathbf{L} \in \mathrm{R}^{N\mathrm{x}N}$

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

Normalized:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}$$

Preliminary:

Spectral Theorem for Graph Laplacian **L**

$$\mathbf{L} = \mathbf{U\Lambda U}^{\mathrm{T}}$$

$$\mathbf{LU} = \mathbf{\Lambda U}$$

– **U**: Fourier basis. → **real** and **orthonormal** eigenvectors of **L**

– **Λ**: Fourier modes (frequencies) → the diagonal is the **ordered** and **real nonnegative** eigenvalues of **L**

Graph Fourier Transform

can be seen as the e^{-jwt}
in Fourier Transformation

$$\mathrm{F}[f(\boldsymbol{\lambda})] = \hat{f}\,(\boldsymbol{\lambda}) = \sum_{i=1}^{\mathrm{n}} f(i) * U(i)$$

$$\hat{f}\,(\boldsymbol{\lambda}) = \mathbf{U}^{\mathrm{T}} f \Leftrightarrow f = \mathbf{U}\hat{f}(\boldsymbol{\lambda})$$

$\hat{f}\,(\boldsymbol{\lambda})$ is the projection value of Fourier basis **U**

Preliminary: *Graph Convolution via Graph Fourier Transform*

Convolution in the spatial domain equals
point-wise multiplication of two signals in the frequency domain, i.e, Fourier-transformed spatial signals

$$F\left((f * h)_{\mathbf{G}}\right) = \hat{f}(w)\times\hat{h}(w)$$

$$(f * h)_{\mathbf{G}} = F^{-1}(\hat{f}(w)\hat{h}(w))$$

$$\downarrow \quad \hat{f}\,(\boldsymbol{\lambda}) = \mathbf{U}^{\mathrm{T}} f$$

Hamada Product
Element-wise Multiplication

$$(f * h)_{\mathbf{G}} = F^{-1}\left(\left(\mathbf{U}^{\mathrm{T}} f\right) \odot (\mathbf{U}^{\mathrm{T}} h)\right)$$

$$\downarrow \quad f = \mathbf{U}\hat{f}(\boldsymbol{\lambda})$$

$$(f * h)_{\mathbf{G}} = \mathbf{U}\left(\left(\mathbf{U}^{\mathrm{T}} f\right) \odot (\mathbf{U}^{\mathrm{T}} h)\right)$$

$$\downarrow$$

$$(f * h)_{\mathbf{G}} = \underbrace{\mathbf{U}}_{[\mathrm{n\ x\ n}]}\ \underbrace{\mathrm{diag}[\hat{\mathrm{h}}(\lambda_1), \hat{\mathrm{h}}(\lambda_2), \ldots, \hat{\mathrm{h}}(\lambda_n)]}_{[\mathrm{n\ x\ n}]}\ \underbrace{\mathbf{U}^{\mathrm{T}}}_{[\mathrm{n\ x\ n}]}\ \underbrace{f}_{[\mathrm{n\ x\ d}]}$$

Source: https://en.wikipedia.org/wiki/Convolution_theorem.

Graph Convolution

$$(f * h)_{\mathbf{G}} = \mathbf{U}\,\boxed{\mathrm{diag}[\hat{\mathrm{h}}(\lambda_1), \hat{\mathrm{h}}(\lambda_2), \dots, \hat{\mathrm{h}}(\lambda_{\mathrm{n}})]}\,\mathbf{U}^{\mathrm{T}} f$$

Activation Function

$$\boldsymbol{\mathcal{Y}} = \sigma(\mathbf{U}\mathbf{g}_{\boldsymbol{\theta}}\mathbf{U}^{\mathrm{T}}\boldsymbol{\chi})$$

K$^{\text{th}}$ Polynomial Function

$$\mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}) = \sum_{k=0}^{\mathrm{K}} \theta_k \boldsymbol{\Lambda}^k$$

Approximate

$$\boldsymbol{\mathcal{Y}} = \sigma(\mathbf{U}\mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})\mathbf{U}^{\mathrm{T}}\boldsymbol{\chi})$$
$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_{\mathrm{n}})$$

$$\boldsymbol{\mathcal{Y}} = \sigma\left(\mathbf{U}\sum_{k=0}^{\mathrm{K}} \theta_k \boldsymbol{\Lambda}^k\,\mathbf{U}^{\mathrm{T}}\boldsymbol{\chi}\right) = \sigma\left(\sum_{k=0}^{\mathrm{K}} \theta_k\,(\mathbf{U}\boldsymbol{\Lambda}^k\mathbf{U}^{\mathrm{T}})\boldsymbol{\chi}\right) = \sigma\left(\sum_{k=0}^{\mathrm{K}} \theta_k\,(\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}})^{\mathrm{k}}\boldsymbol{\chi}\right) = \sigma\left(\sum_{k=0}^{\mathrm{K}} \theta_k\,\mathbf{L}^{\mathrm{k}}\boldsymbol{\chi}\right)$$

$$\boxed{\boldsymbol{\mathcal{Y}} = \sigma\left(\sum_{k=0}^{\mathrm{K}} \theta_k\,\mathbf{L}^{\mathrm{k}}\boldsymbol{\chi}\right)}$$

Reference: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, *NIPS*, 2016.

Graph Convolution

Convolution:
Weighted Sum

Weights Sharing → Translation Invariance

$$\mathcal{Y} = \sigma\left(\sum_{k=0}^{K} \theta_k \mathbf{L}^k \chi\right)$$

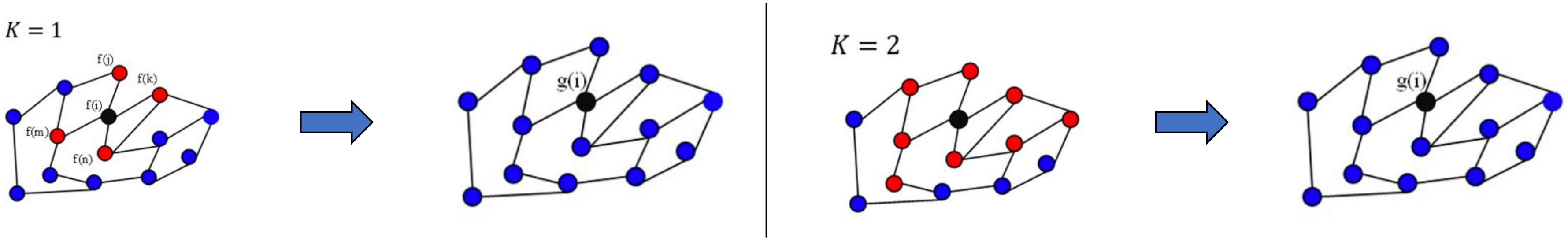
No need for Fourier Transform

“Node Aggregation”
K is Filter Size

GCN Key Idea: Use "**edge information**" to **aggregate** "**node information**" to generate a new "**node representation**"

“Laplace Operator”
Local connectivity

$$\mathbf{x}_{\mathbf{new}} \leftarrow \mathbf{L}\mathbf{x}_{\mathbf{i}} = \sum_{\mathbf{j}} \mathbf{A}_{\mathbf{ij}}(\mathbf{x}_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}})$$



Pros:

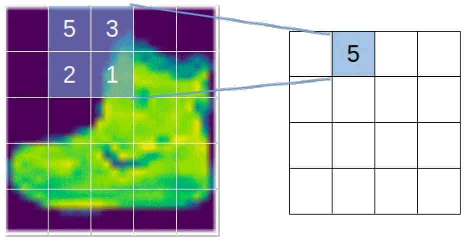
1. No need for Spectral Decomposition
2. Less number of parameters (decrease model complexity) → $K \ll n$

Cons: Need to compute L^k

Image Credit: in the public domain.

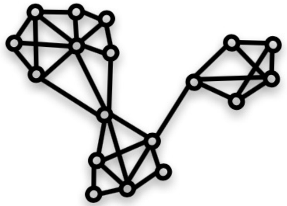
Pooling on Graphs (*Graph Coarsening*)

- Traditional CNN doesn't need to consider neighbors after convolutions
  - [Euclidean Structure] The output feature maps are regular
  - The neighbors are "meaningful"



- GCNs need to consider neighbors after convolutions
  - [Non-Euclidean Structure] The output graphs' nodes are not arranged in any meaningful way

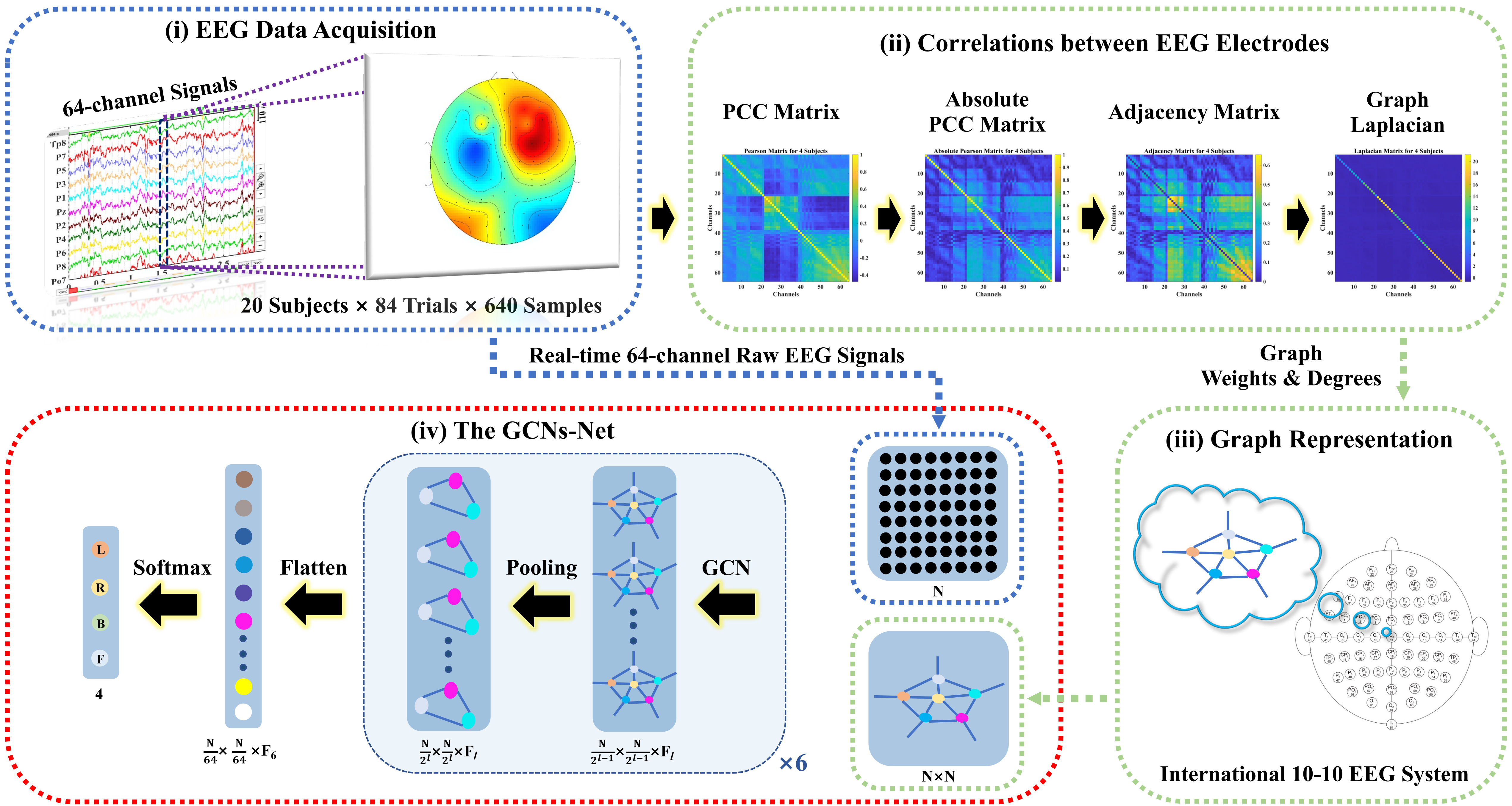
    → Need to find "meaningful" neighbors of graph nodes after convolution
  - Use Graclus Multilevel Clustering Algorithm, a clustering algorithm to find "meaningful" neighbors
  - Minimize the *Local Normalized Cut* (a cluster grouping method)



$$-W_{ij}\left(\frac{1}{d_i}+\frac{1}{d_j}\right)$$

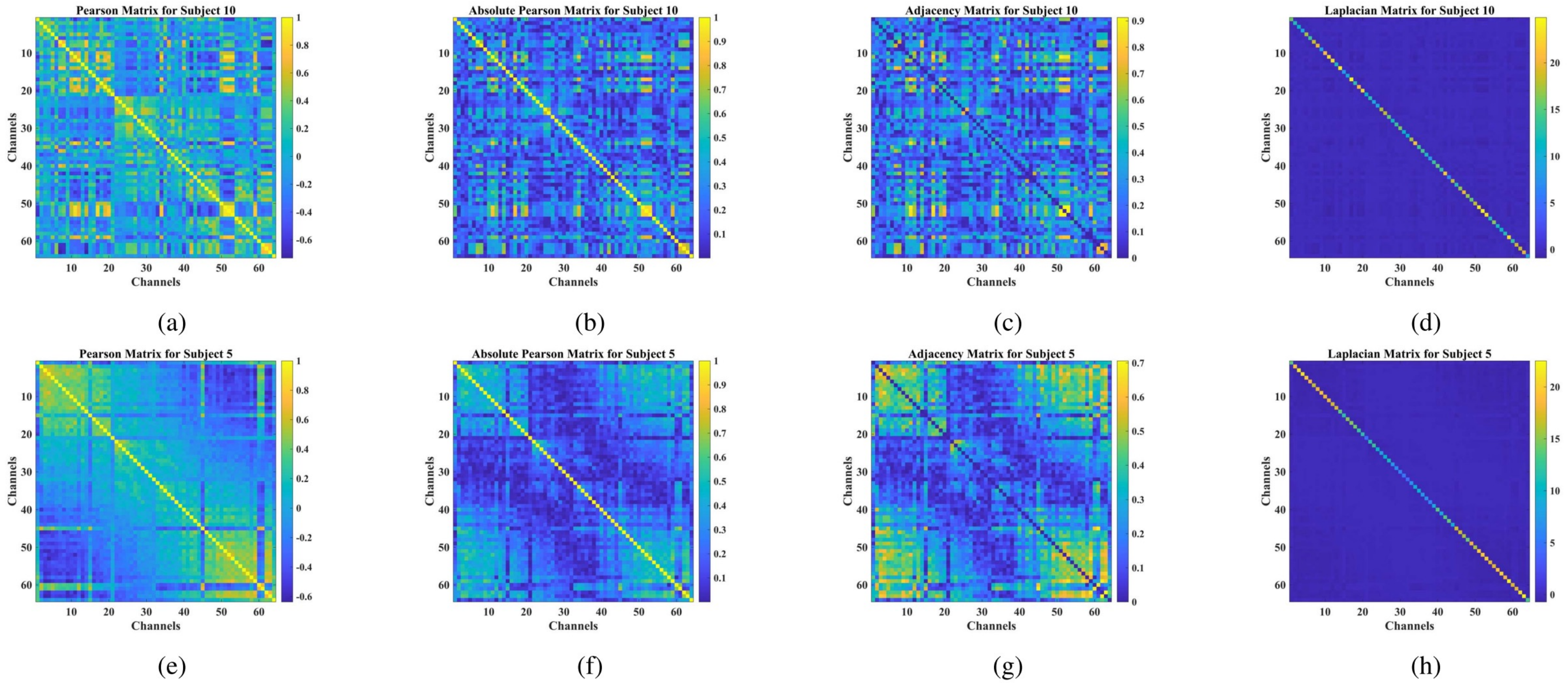
  - i and j denote two nodes
  - W_{ij} is the learned weight between node i and node j

Image Credit: in the public domain.


(i) EEG Data Acquisition
64-channel Signals
20 Subjects × 84 Trials × 640 Samples
(ii) Correlations between EEG Electrodes
PCC Matrix
Absolute PCC Matrix
Adjacency Matrix
Graph Laplacian
Pearson Matrix for 4 Subjects
Absolute Pearson Matrix for 4 Subjects
Adjacency Matrix for 4 Subjects
Laplacian Matrix for 4 Subjects
Channels
Real-time 64-channel Raw EEG Signals
Graph Weights & Degrees
(iii) Graph Representation
International 10-10 EEG System
(iv) The GCNs-Net
Softmax
Flatten
Pooling
GCN
L
R
B
F
4
N
N×N
×6

Correlation among EEG electrodes
Two Subjects: Subject 10 and 5

Problem: Individual Variability



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for Subjects 10 and 5 from the PhysioNet dataset. (a) PCC matrix for Subject 10. (b) Absolute PCC matrix for Subject 10. (c) Adjacency matrix for Subject 10. (d) Graph Laplacian for Subject 10. (e) PCC matrix for Subject 5. (f) Absolute PCC matrix for Subject 5. (g) Adjacency matrix for Subject 5. (h) Graph Laplacian for Subject 5.

Correlation among EEG electrodes 20 Subjects and 100 Subjects

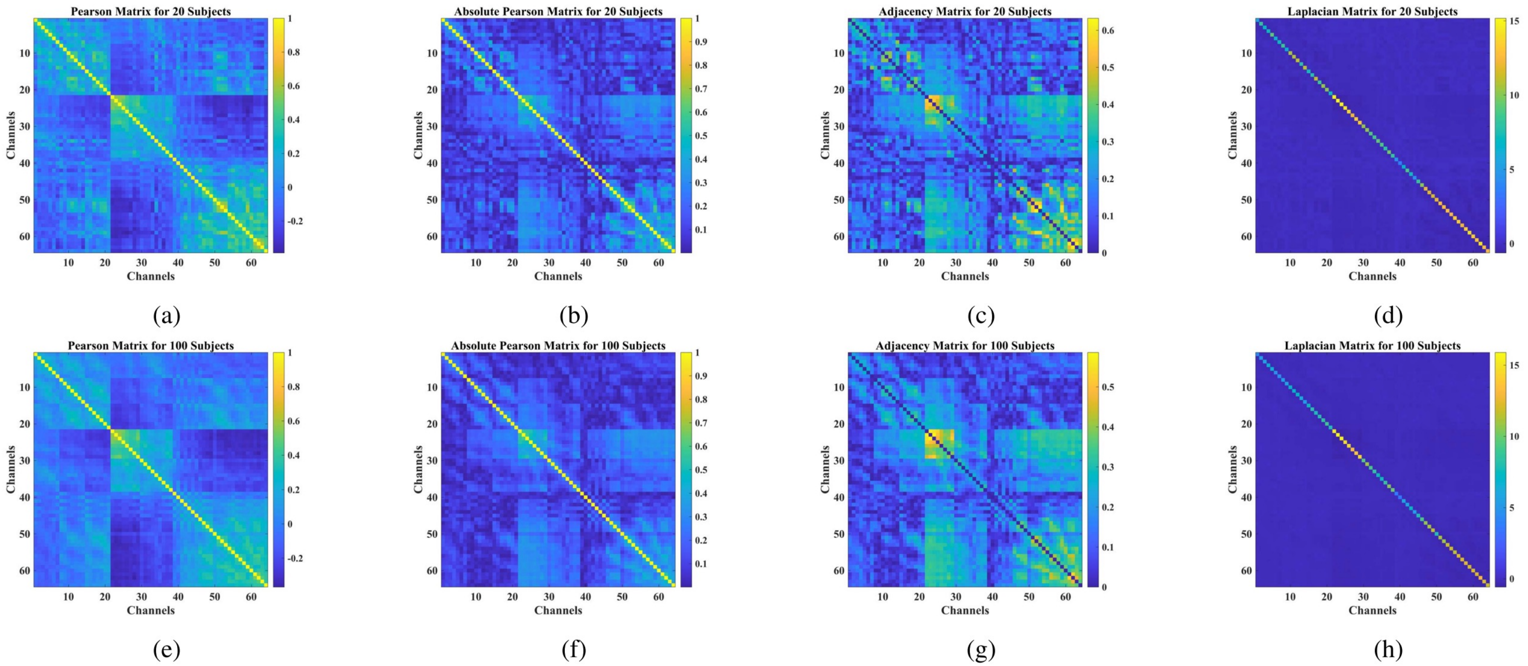


Fig. 2. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for 20 and 100 subjects, respectively, from the PhysioNet dataset. (a) PCC matrix for 20 subjects. (b) Absolute PCC matrix for 20 subjects. (c) Adjacency matrix for 20 subjects. (d) Graph Laplacian for 20 subjects. (e) PCC matrix for 100 subjects. (f) Absolute PCC matrix for 100 subjects. (g) Adjacency matrix for 100 subjects. (h) Graph Laplacian for 100 subjects.

Increasing *the num. of subjects* alleviates *individual variability*

Model Design for 64-electrode EEG System

TABLE I

IMPLEMENTATION DETAILS OF THE PROPOSED GCNS-NET ON THE PHYSIONET DATASET

Layer	Type	Maps	Size	Edges	Polynomial Order	Pooling Size	Activation	Weights	Bias
Softmax	Fully-connected	–	O	–	–	–	Softmax	$\frac{N}{64} \times \frac{N}{64} \times F_6 \times O$	O
Flatten	Flatten	–	$\frac{N}{64} \times \frac{N}{64} \times F_6$	–	–	–	–	–	–
P6	Max-pooling	F_6	$\frac{N}{32}$	$\sum_{i=1}^{\frac{N}{32}-1} i$	–	2	–	–	–
C6	Convolution	F_6	$\frac{N}{32}$	$\sum_{i=1}^{\frac{N}{32}-1} i$	K	–	Softplus	$F_5 \times F_6 \times K$	$\frac{N}{32} \times F_6$
P5	Max-pooling	F_5	$\frac{N}{16}$	$\sum_{i=1}^{\frac{N}{16}-1} i$	–	2	–	–	–
C5	Convolution	F_5	$\frac{N}{16}$	$\sum_{i=1}^{\frac{N}{16}-1} i$	K	–	Softplus	$F_4 \times F_5 \times K$	$\frac{N}{16} \times F_5$
P4	Max-pooling	F_4	$\frac{N}{8}$	$\sum_{i=1}^{\frac{N}{8}-1} i$	–	2	–	–	–
C4	Convolution	F_4	$\frac{N}{8}$	$\sum_{i=1}^{\frac{N}{8}-1} i$	K	–	Softplus	$F_3 \times F_4 \times K$	$\frac{N}{8} \times F_4$
P3	Max-pooling	F_3	$\frac{N}{4}$	$\sum_{i=1}^{\frac{N}{4}-1} i$	–	2	–	–	–
C3	Convolution	F_3	$\frac{N}{4}$	$\sum_{i=1}^{\frac{N}{4}-1} i$	K	–	Softplus	$F_2 \times F_3 \times K$	$\frac{N}{4} \times F_3$
P2	Max-pooling	F_2	$\frac{N}{2}$	$\sum_{i=1}^{\frac{N}{2}-1} i$	–	2	–	–	–
C2	Convolution	F_2	$\frac{N}{2}$	$\sum_{i=1}^{\frac{N}{2}-1} i$	K	–	Softplus	$F_1 \times F_2 \times K$	$\frac{N}{2} \times F_2$
P1	Max-pooling	F_1	N	$\sum_{i=1}^{N-1} i$	–	2	–	–	–
C1	Convolution	F_1	N	$\sum_{i=1}^{N-1} i$	K	–	Softplus	$1 \times F_1 \times K$	$N \times F_1$
Input	Input	1	N	$\sum_{i=1}^{N-1} i$	–	–	–	–	–

Model Optimization

- **Ablation Study**: Optimal Model Structure (64-electrode EEG system)
  - C6-P6-K2 with [16, 32, 64, 128, 256, 512] filters
- **Gradient Iterative Solver**: Adam Optimizer with the Stochastic Gradient Descent (SGD) algorithm
  - Learning Rate: 0.01
  - Batch Size: 1,024
- **Activation Function**: Softplus (Smooth Rectified Linear Unit)

$$f(\mathbf{x}) = \log(1 + e^{\mathbf{x}})$$

- **Model Output**: via Softmax: **y** is labels, $\hat{\mathbf{y}}$ is the final output tasks

$$\hat{y}_i = \operatorname{argmax}\left(\frac{e^{y_i}}{\sum_{i=1}^{4} e^{y_i}}\right)$$

- **Loss Function**: Cross-entropy Loss with L2 regularization

$$\text{Loss} = -\sum_{i=1}^{4} y_i \log(\hat{y}_i) + \lambda\left(\sum_{j=1}^{n} w_j^2 + b_j^2\right)$$

$\lambda = 1 \times 10^{-6}$ is the coefficient of the L2 regularization.

Experimental Results
Groupwise Prediction and Subject-specific Adaptation

TABLE IV

PERFORMANCE COMPARISONS ON THE PHYSIONET DATASET

Related Work	Max. Accuracy	Avg. Accuracy	*p*-value	Level	Approach	Num. of Subjects
Dose *et al.* (2018) [22]	–	58.58%	–	Group	CNNs	105
	80.38%	68.51%	< 0.05	Subject		1
Ma *et al.* (2018) [60]	82.65%	68.20%	–	Group	RNNs	12
Hou *et al.* (2020) [20]	94.50%	–	–	Group	ESI-CNNs	10
	96.00%	–	> 0.05	Subject		1
Hou *et al.* (2022) [34]	94.64%	–	–	Group	BiLSTM-GCN	20
	98.81%	95.48%	> 0.05	Subject		1
Jia *et al.* (2022) [40]	94.16%	93.78%	–	Group	Graph ResNet	20
	98.08%	94.18%	> 0.05	Subject		1
Author	**89.39%**	**88.57%**	–	**Group**	**GCNs-Net**	**20**
	88.14%	–				**100**
	98.72%	**93.06%**		**Subject**		**1**

Conclusions and Future Work

- We have developed a novel GCNs-Net method that processes EEG signals through **Graph Representation Learning** to deeply extract *network patterns of brain dynamics* for EEG data classification.
- The introduced method has been proven to converge for both personalized and groupwise predictions, indicating that the GCNs-Net is able to build a generalized representation of EEG time-series signals against both personalized and groupwise variations.
- We are looking forward to modeling EEG signals as ***dynamic graphs*** and processing them via ***dynamic graph learning methods***.

Deep Feature Mining via Attention-based BiLSTM-GCN for Human Motor Imagery Recognition

Yimin Hou [1], Shuyue Jia [1, 2], Xiangmin Lun [1], Shu Zhang [3], Tao Chen [1], Fang Wang [1], and Jinglei Lv [4]

[1] School of Automation Engineering, Northeast Electric Power University

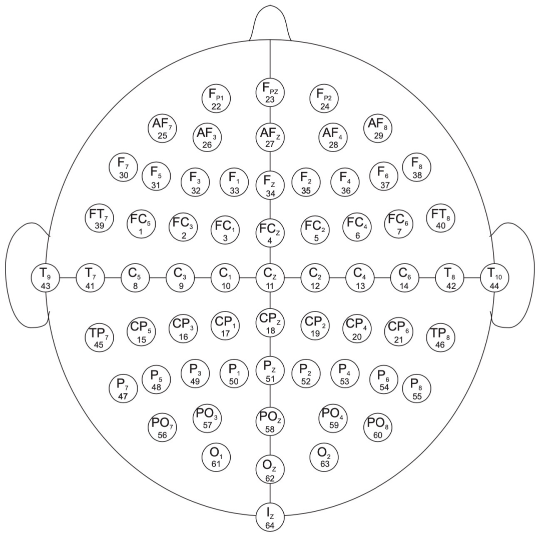
[2] Department of Computer Science, City University of Hong Kong

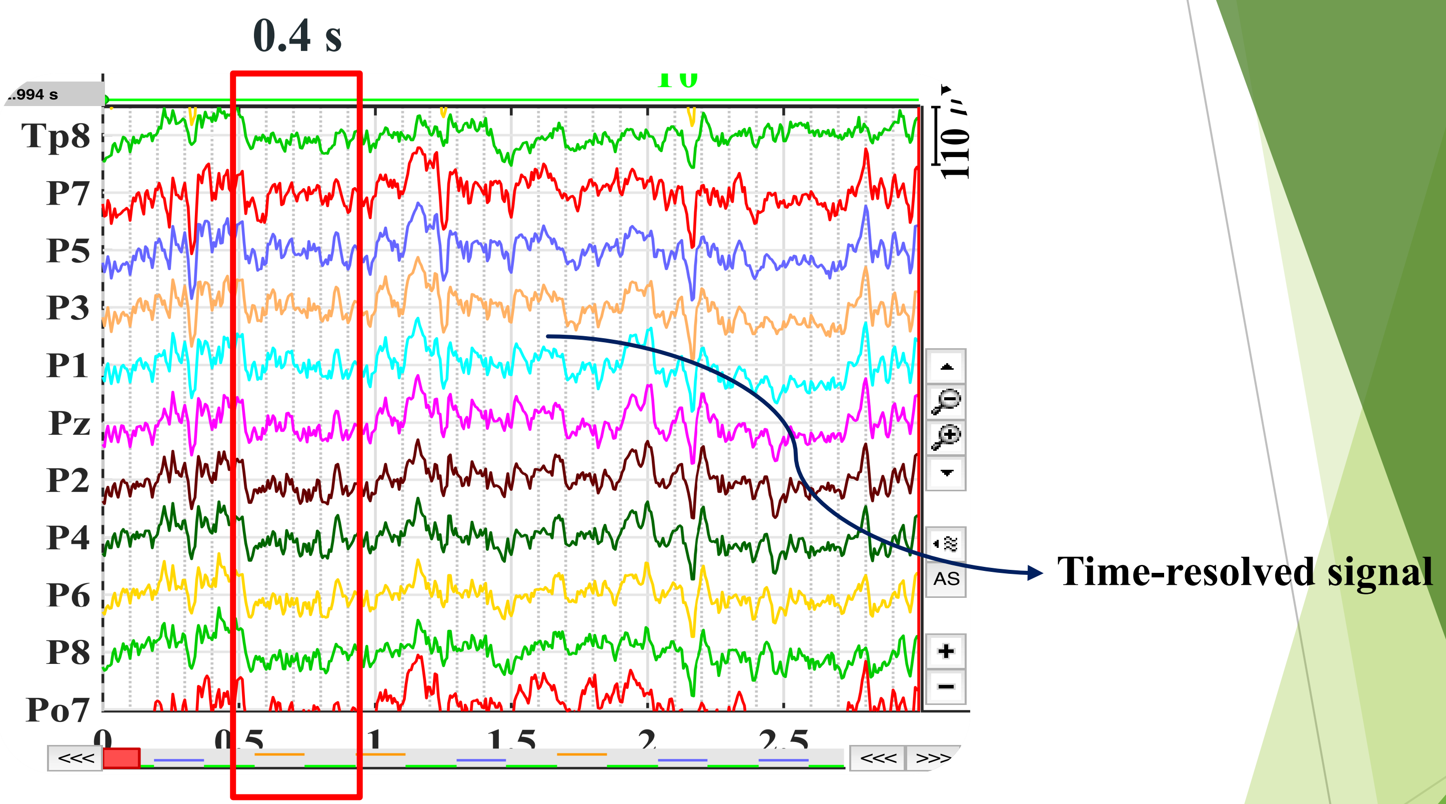
[3] School of Computer Science, Northwestern Polytechnical University

[4] School of Biomedical Engineering and Brain and Mind Center, University of Sydney

EEG Deep Learning Library: https://github.com/SuperBruceJia/EEG-DL

One Problem of the GCNs-Net





Spatial information

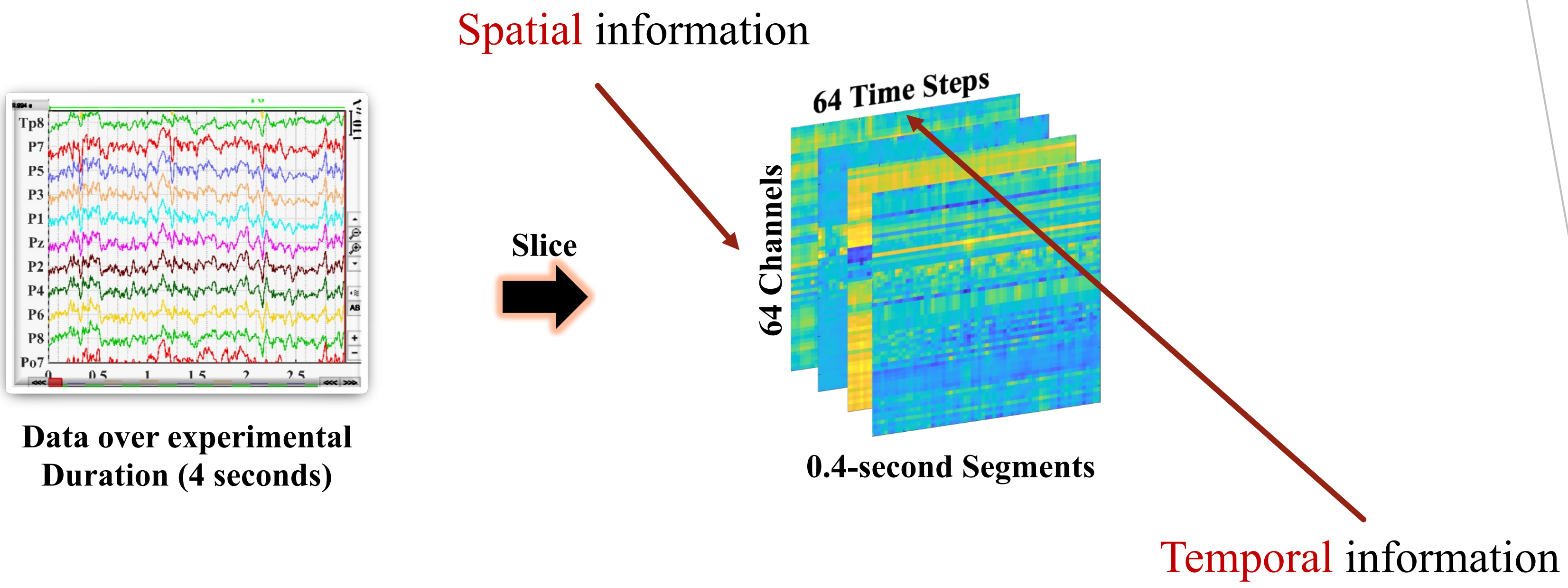
Temporal information

- GCNs-Net uses the Time-resolved Signal → doesn’t consider the **temporal information**

Motivation:

- ✓ **Consider temporal and spatial information at the same tims from EEG signals**
- ✓ **Maintain high responding time**

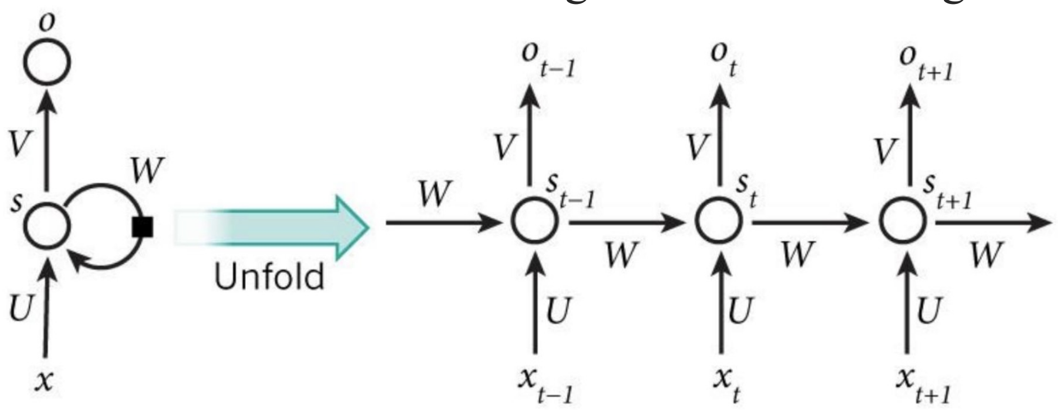
64-channel Raw EEG Signals Acquisition



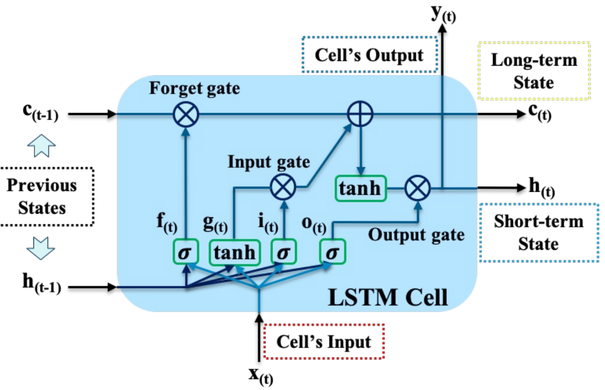
- 4-s (experimental duration) signals → 0.4-s segments over time
- Each segment: 64 channels × 64 time steps
- Pre-processed Data: Temporal information + Spatial information

Temporal Information Extraction

Unrolling the network through time



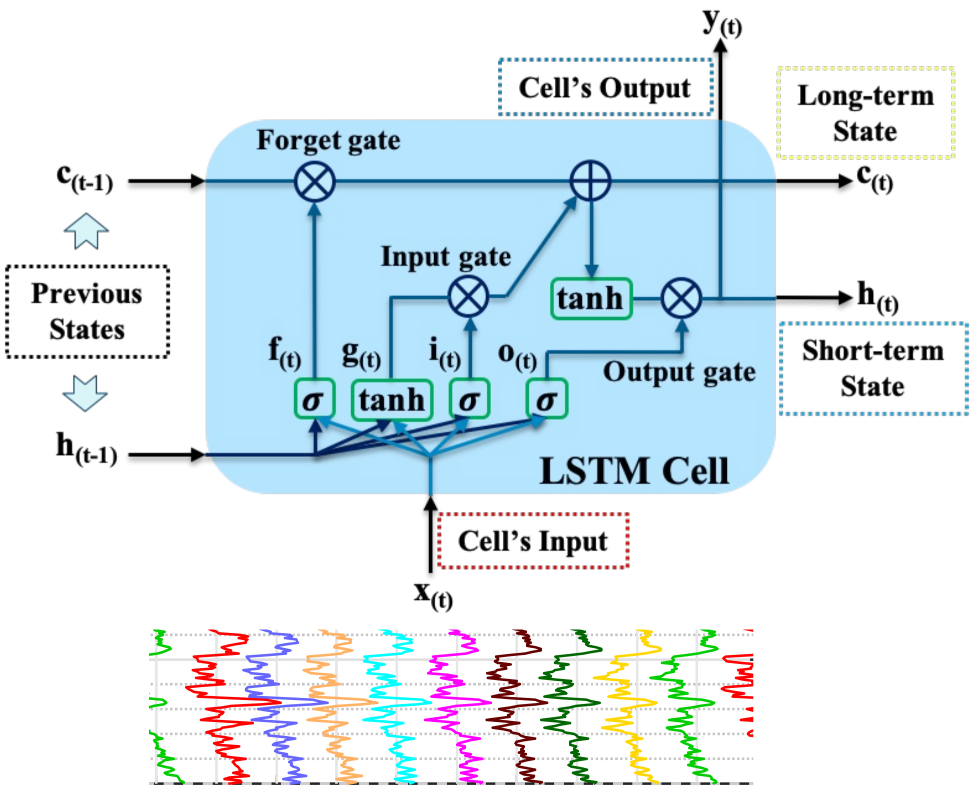
Recurrent Neural Network (RNN)



Long Short-term Memory (LSTM)

- Designed for order-mattered sequential data, *e.g.*, time point signals
- The learned features at *time step t* are affected by $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ → continuly learn from time series
- LSTM: better capture long-range sequence dependencies
- GRU: lightweight architecture with comparable performance

Long Short-term Memory (LSTM)



64-channel Raw EEG Signals at Time Step $x_{(t)}$

- Capture Long-range Dependency

  by the long-term state path $c_{t-1} \rightarrow c_t$
- Input Gate: store x_t and control c_t's input
- Forget Gate: control c_{t-1}
- Output Gate: control c_t's output

  $\rightarrow$ short-term state h_t (Cell's Output)
- More parameters to store information
- Bidirectional:

  (1) $x_1 \rightarrow x_t$

  (2) $x_t \rightarrow x_1$

Attention Mechanism

- Signals/Outputs

equally treated/contributed

vs.

differently treated/contributed with importance

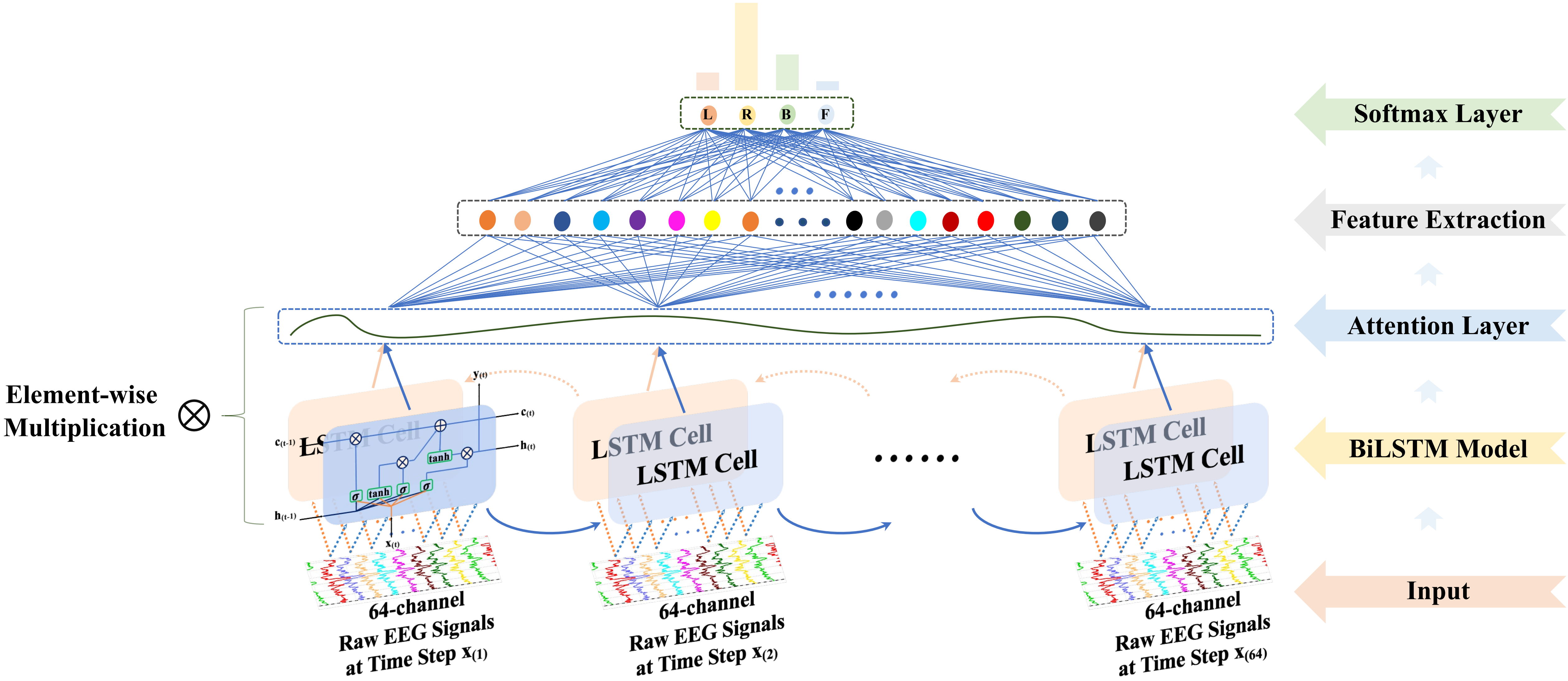
FC Layer $$\mathbf{U}_t = \tanh(\mathbf{W}_w \mathbf{y}_t + \mathbf{b}_w)$$

Attentional Weights $$\boxed{\boldsymbol{\alpha}_t = \frac{\exp(\mathbf{U}_t^T \mathbf{U}_w)}{\sum_t \exp(\mathbf{U}_t^T \mathbf{U}_w)}}$$

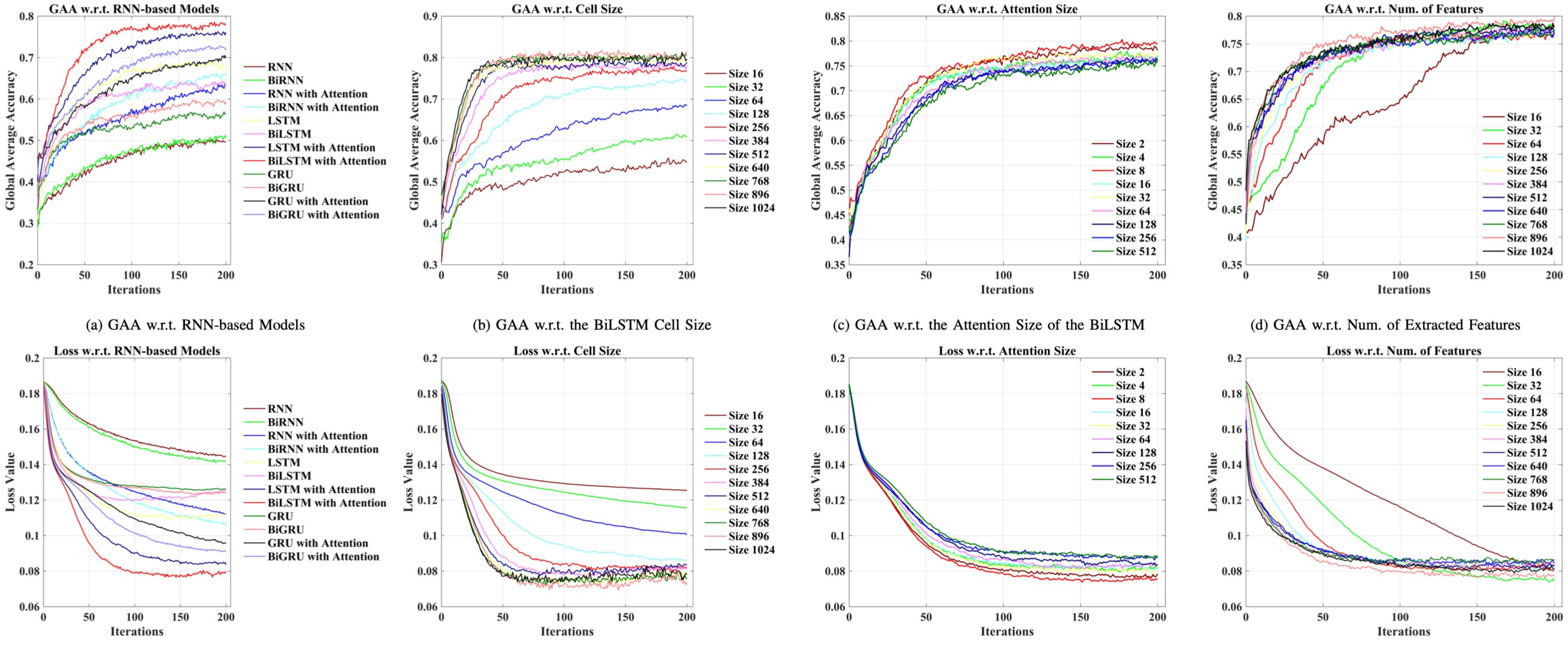
Weighted Sum $$\mathbf{U}_t = \sum_t \boldsymbol{\alpha}_t \mathbf{y}_t$$

Image Credit: Prof. Bolei Zhou.

Attention-based Bidirectional Long Short-term Memory (Bi-LSTM)



Ablation Study

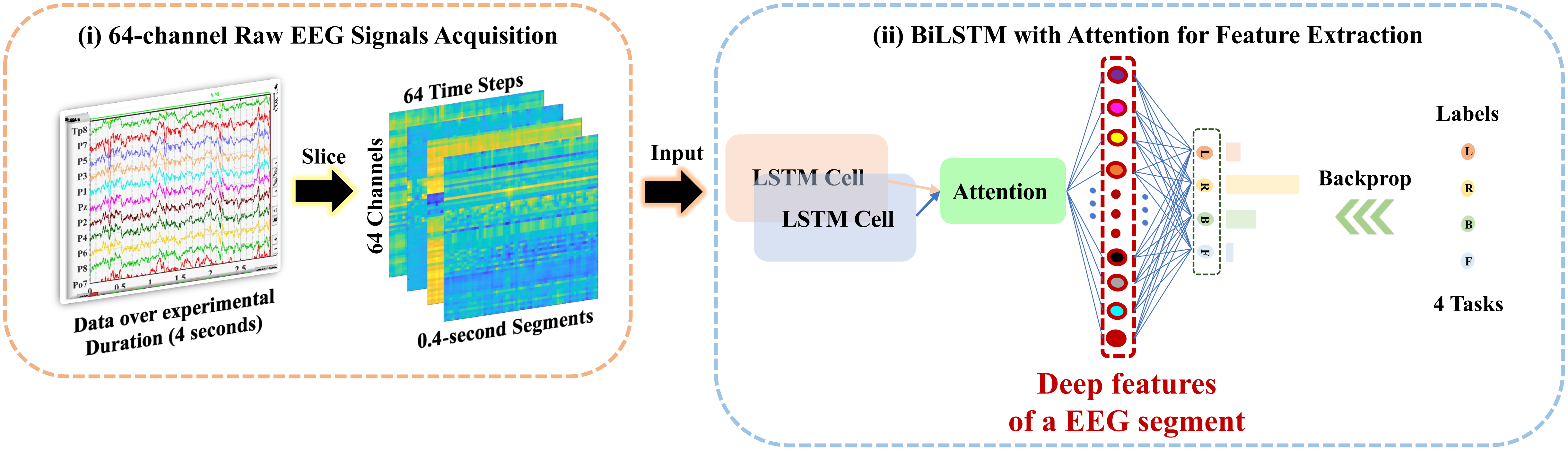


(a) GAA w.r.t. RNN-based Models (b) GAA w.r.t. the BiLSTM Cell Size (c) GAA w.r.t. the Attention Size of the BiLSTM (d) GAA w.r.t. Num. of Extracted Features

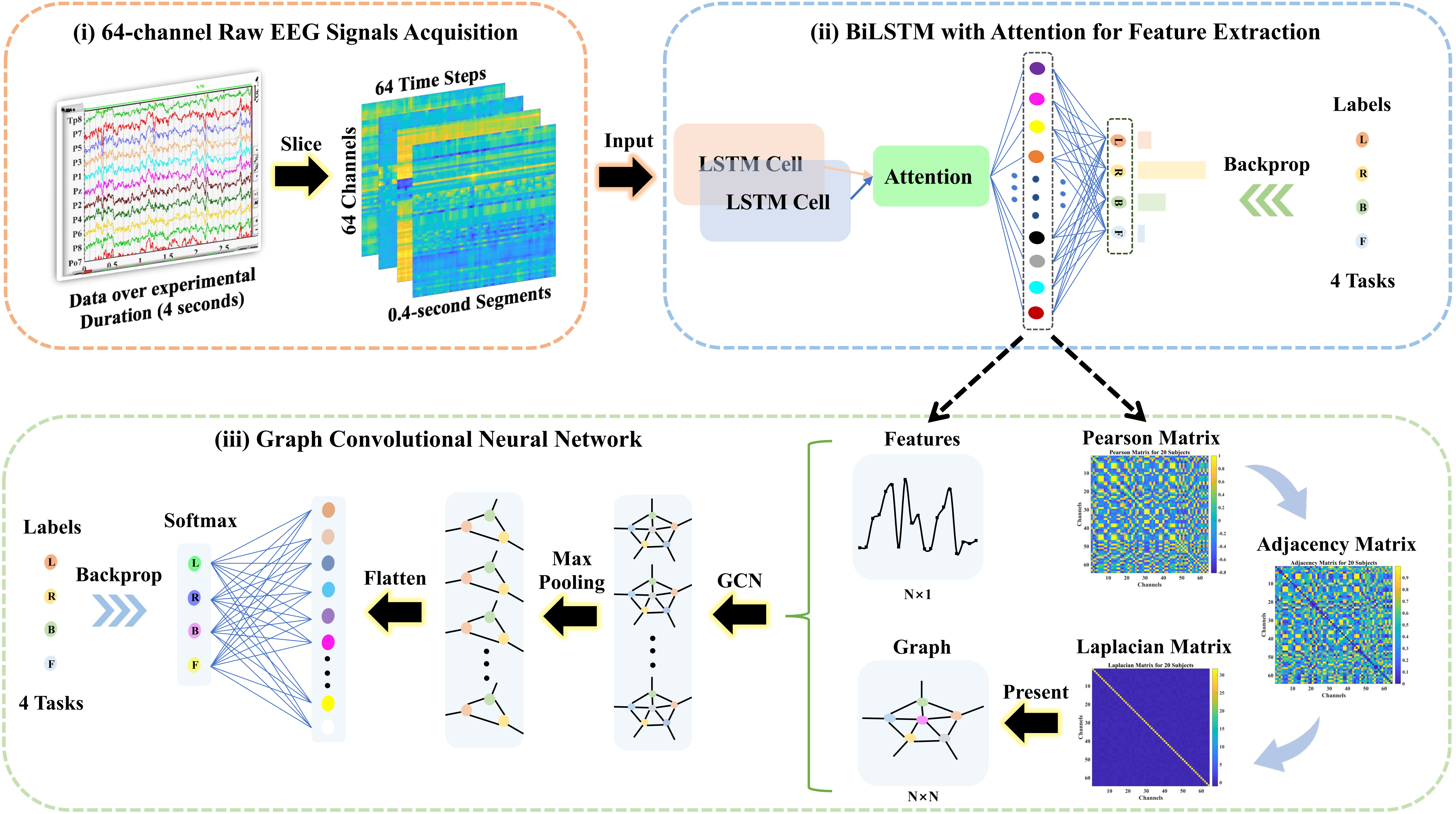
(e) Loss w.r.t. RNN-based Models (f) Loss w.r.t. the BiLSTM Cell Size (g) Loss w.r.t. the Attention Size of the BiLSTM (h) Loss w.r.t. Num. of Extracted Features

Fig. 3: Models and Hyperparameters Comparison w.r.t. the RNN-based Methods for Feature Extraction

Topological structure of features

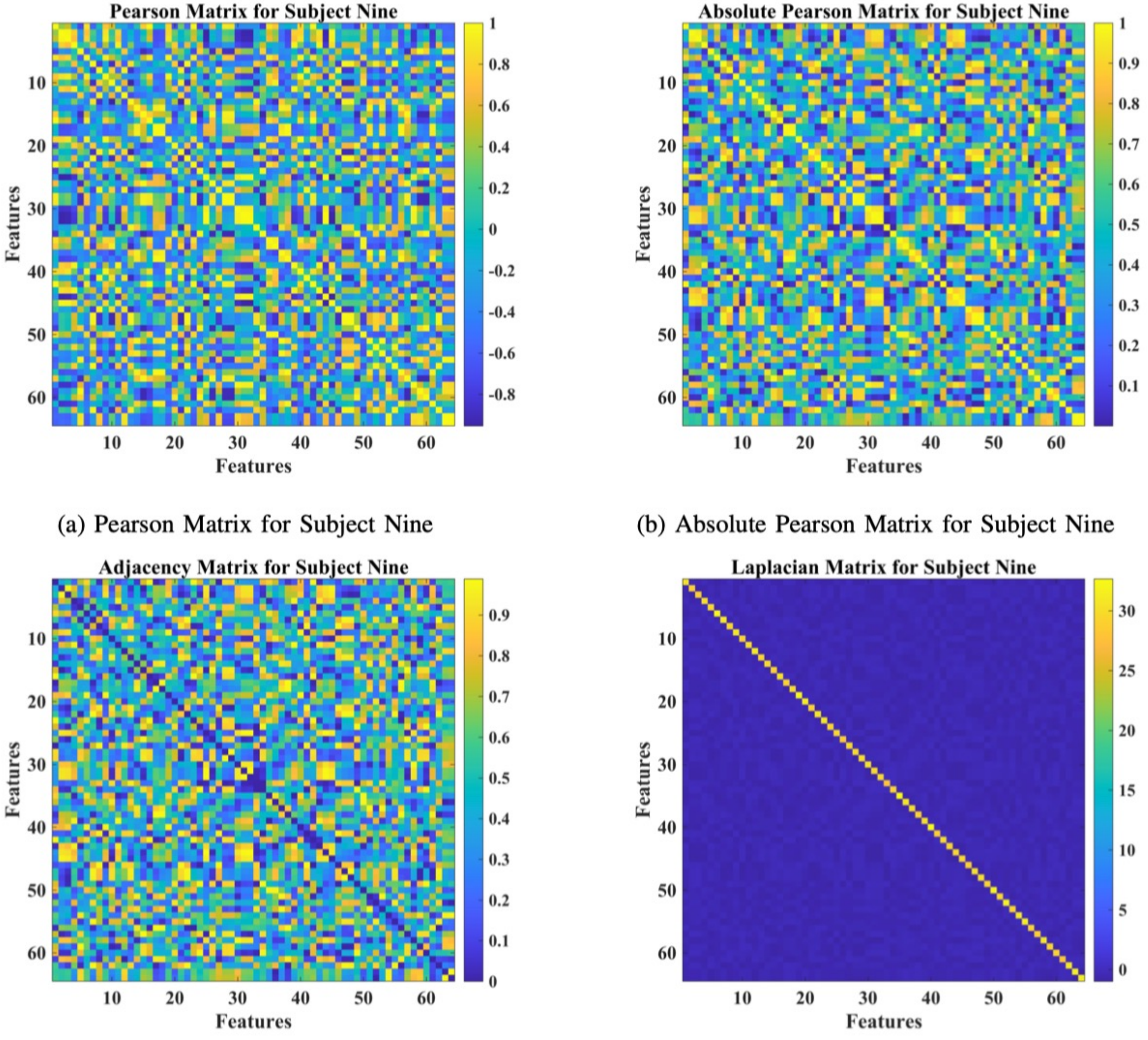


- **Deep feature mining** → Intra-feature relationship → Intra-feature Modeling



(i) 64-channel Raw EEG Signals Acquisition
Slice
64 Time Steps
64 Channels
Data over experimental Duration (4 seconds)
0.4-second Segments
Input
(ii) BiLSTM with Attention for Feature Extraction
LSTM Cell
LSTM Cell
Attention
Labels
Backprop
4 Tasks
(iii) Graph Convolutional Neural Network
Features
N×1
Pearson Matrix
Adjacency Matrix
Graph
N×N
Present
Laplacian Matrix
GCN
Max Pooling
Flatten
Softmax
Labels
Backprop
4 Tasks

Topological structure of features



(a) Pearson Matrix for Subject Nine

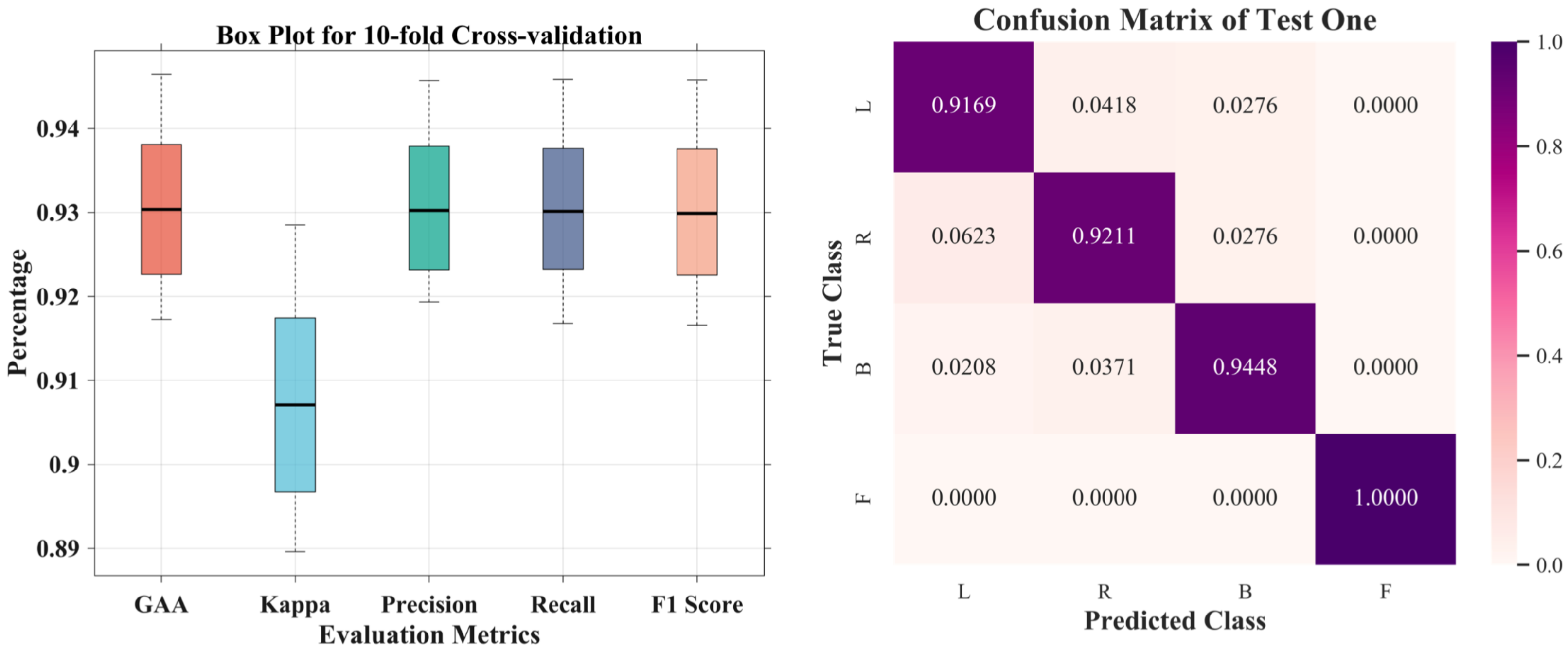
(b) Absolute Pearson Matrix for Subject Nine

(c) Adjacency Matrix for Subject Nine

(d) Laplacian Matrix for Subject Nine

Fig. 4: The Pearson, Absolute Pearson, Adjacency, and Laplacian Matrices for Subject Nine.

Experimental Results - Groupwise Prediction



(a) Box Plot for 10-fold cross validation

(b) Confusion Matrix of Test One

Fig. 5: Box plot and confusion matrix for 10-fold cross validation.

Experimental Results - Subject-Specific Adaptation

TABLE II: Subject-level Evaluation

No. of Subject	GAA	Kappa	Precision	Recall	F1 Score
1	94.05%	92.06%	94.20%	94.32%	94.16%
2	96.43%	95.19%	96.06%	96.06%	96.06%
3	97.62%	96.79%	97.33%	97.08%	97.18%
4	90.48%	87.34%	91.30%	91.11%	90.42%
5	95.24%	93.61%	95.96%	95.06%	95.38%
6	94.05%	92.02%	93.40%	94.96%	93.66%
7	98.81%	98.40%	98.81%	99.07%	98.92%
8	95.24%	93.60%	95.39%	95.04%	95.19%
9	98.81%	98.39%	99.11%	98.68%	98.87%
10	94.05%	91.98%	93.39%	94.70%	93.61%
Average	**95.48%**	**93.94%**	**95.50%**	**95.61%**	**95.35%**

TABLE III: Current studies comparison on subject-level prediction

Related Work	Max. GAA	Approach	Database
Ortiz-Echeverri *et al.* (2019)	94.66%	Sorted-fast ICA-CWT + CNNs	
Sadiq *et al.* (2019)	95.20%	EWT + LS-SVM	BCI Competition IV-a Dataset
Taran *et al.* (2018)	96.89%	TQWT + LS-SVM	
Zhang *et al.* (2019)	83.00%	CNNs-LSTM	
Ji *et al.* (2019)	95.10%	SVM	BCI Competition IV-2a Dataset
Amin *et al.* (2019)	95.40%	MCNNs	
Dose *et al.* (2018)	68.51%	CNNs	
Hou *et al.* (2019)	96.00%	ESI + CNNs	Physionet Database
This work	**98.81%**	**Attention-based BiLSTM-GCN**	

Conclusions and Future Work

- Extensive experimental results provide compelling evidence that the method has converged to both the **subject-level and groupwise predictions** and achieved the best state-of-the-art performance for handling individual variability. The 0.4-s sample size is proven effective and efficient in prediction compared with the traditional 4-s trial length, which means that our proposed framework can provide a **time-resolved solution toward fast response**.
- The proposed method is predicted to advance the **clinical translation** of the EEG MI-based BCI technology to meet the diverse demands, such as those of paralyzed patients. The unprecedented performance with the **highest accuracy** and **time-resolved** prediction is fulfilled via the introduced feature mining approach.
- **Long-range dependencies** among intra-subject or inter-subject EEG signals can be modeled via a **self-attention mechanism**, **Transformer** architecture, or even **AI foundation models**.

Thank you!

Any question?