



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

Yimin Hou¹, Shuyue Jia^{1, 2}, Xiangmin Lun¹, Ziqian Hao³, Yan Shi¹,
Yang Li⁴, Rui Zeng⁵, 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, The University of Sydney

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

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



A potential market

Key Points in dealing with EEG time series

► Individual Variability → Lower Classification Accuracy

- Low SNR
- Different brain electrical conductivity ← different anatomical structure of brain
- Electrodes' positional error

Feature Extraction

EEG Electrodes'
Structure Modeling

► Slow 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)

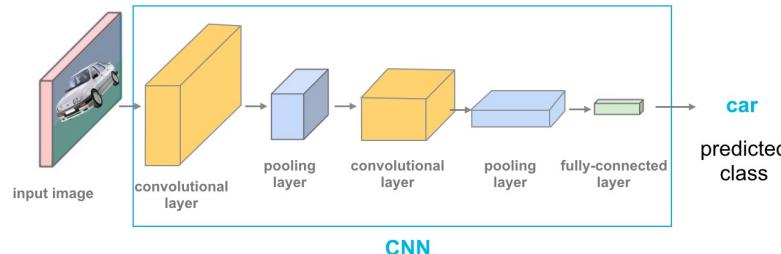
Time-resolved or Window-based
Signal Sampling

► Lower Group-level Accuracy → Hard to develop Applications for a Group of People

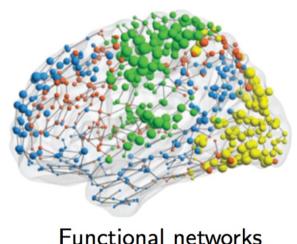
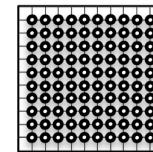
- [most literature] Subject-level prediction (Our Work)
- Group-level prediction (Our Work)

Motivation

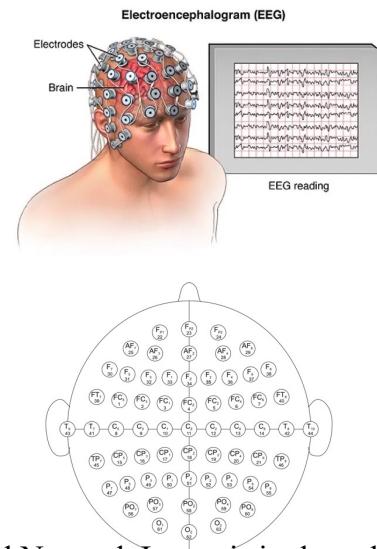
Convolutional Neural Networks:



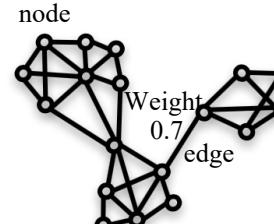
- **Module:** Convolution → Pooling → Fully-connected
- **Pros:** Translation Equivalence, Translation Invariance, Weight Sharing
- **Modeling:** Euclidean-Structured Data (e.g. Image, Speech, Natural Language)
- Neuroscience research has increasingly emphasized **Brain Network Dynamics**
 - Model **Functional Topological Connectivity** of EEG Electrodes → **Graph** (Non-Euclidean Structure)



measure
→
← interpret



model
→
← reflect



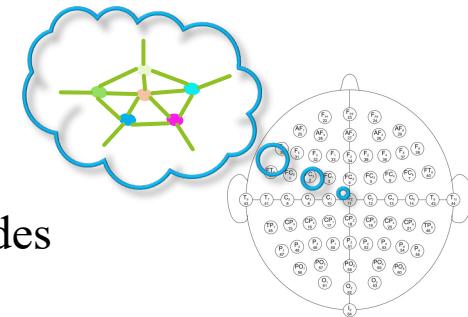
Our Question

How to model the EEG System
as a **Graph**?

How can we process EEG Signals
via **Graph Representation Learning**?

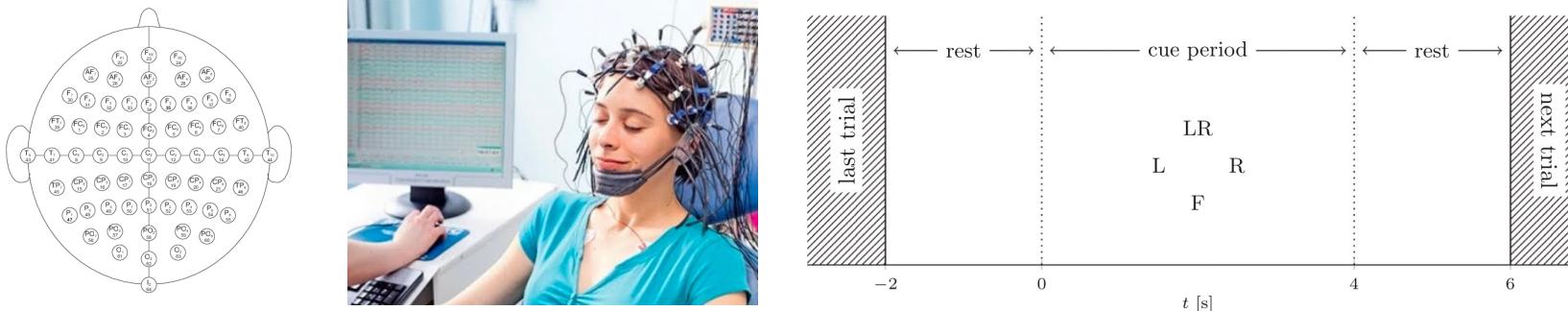
Can we directly apply convolutions on graphs?

- ▶ Traditional CNN **cannot** directly process graph signals
 - ▶ **Graphs are irregular** (*i.e.*, unordered and vary in size)
 - ▶ Convolutions **cannot keep Translation Invariance on 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
 - Analyze and interpret **Brain Network Dynamics**



Benchmark Dataset

- ▶ 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)

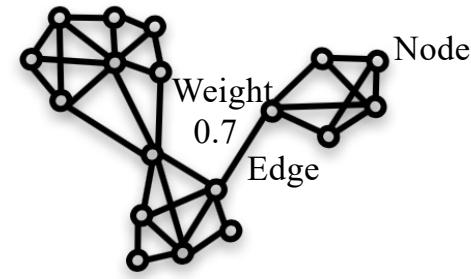


- ▶ **109 subjects** (the largest number of participants in the field of EEG Motor Imagery)
- ▶ Task: **4-class EEG Motor Imagery Classification**
 - ✓ Imagining (Task 1) left fist, (Task 2) right fist, (Task 3) both fists, (Task 4) both feet
- ▶ For each subject, **3 runs, 7 trials, 4 classes** → 84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

Preliminary: Graph Representation

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

- \mathbf{V} : nodes (vertices), $|\mathbf{V}| = N$
- \mathbf{E} : edges (links) that connect nodes
- \mathbf{A} : weights (correlations) between nodes



Nodes Correlations: Pearson Matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ (denotes as PCC matrix)

- Measure the linear correlations between node \mathbf{x} and node \mathbf{y}
- μ is the mean, σ is the standard deviation, and $P_{x,y}$ is the Pearson Correlation Coefficient between node \mathbf{x} and node \mathbf{y}

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

- Absolute Pearson Matrix: $|\mathbf{P}| \in \mathbb{R}^{N \times N}$ and $|P_{ij}| \in [0, 1]$ → **Note:** in this work, we only consider scale

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

Graph Degrees: Degree Matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$

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

Graph Representation: Combinatorial Laplacian $\mathbf{L} \in \mathbb{R}^{N \times N}$

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

Normalized:

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

Preliminary:

Spectral Theorem for Graph Laplacian \mathbf{L}

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

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

- \mathbf{U} : Fourier basis → **real** and **orthonormal** eigenvectors of \mathbf{L}
- Λ : Fourier modes → the diagonal is the **ordered** and **real nonnegative** eigenvalues of \mathbf{L}

Graph Fourier Transform of Signal f

can be seen as the $e^{-j\omega t}$
in Fourier Transforms

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

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

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

Preliminary: Graph Convolution via Graph Fourier Transform

Convolution in the spatial domain \Leftrightarrow point-wise multiplication of two transformed signals in the frequency domain

Signal f
Signal h

$$F((f * h)_G) = \hat{f}(w) \times \hat{h}(w)$$

F: Fourier Transforms
 F^{-1} : Inverse Fourier Transforms
 $\hat{f}(w)$: $F(f)$
 $\hat{h}(w)$: $F(h)$

$$(f * h)_G = F^{-1}(\hat{f}(w) \times \hat{h}(w))$$

Hadamard Product
(Element-wise Multiplication)

$$\hat{f}(\lambda) = \mathbf{U}^T f$$

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

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

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

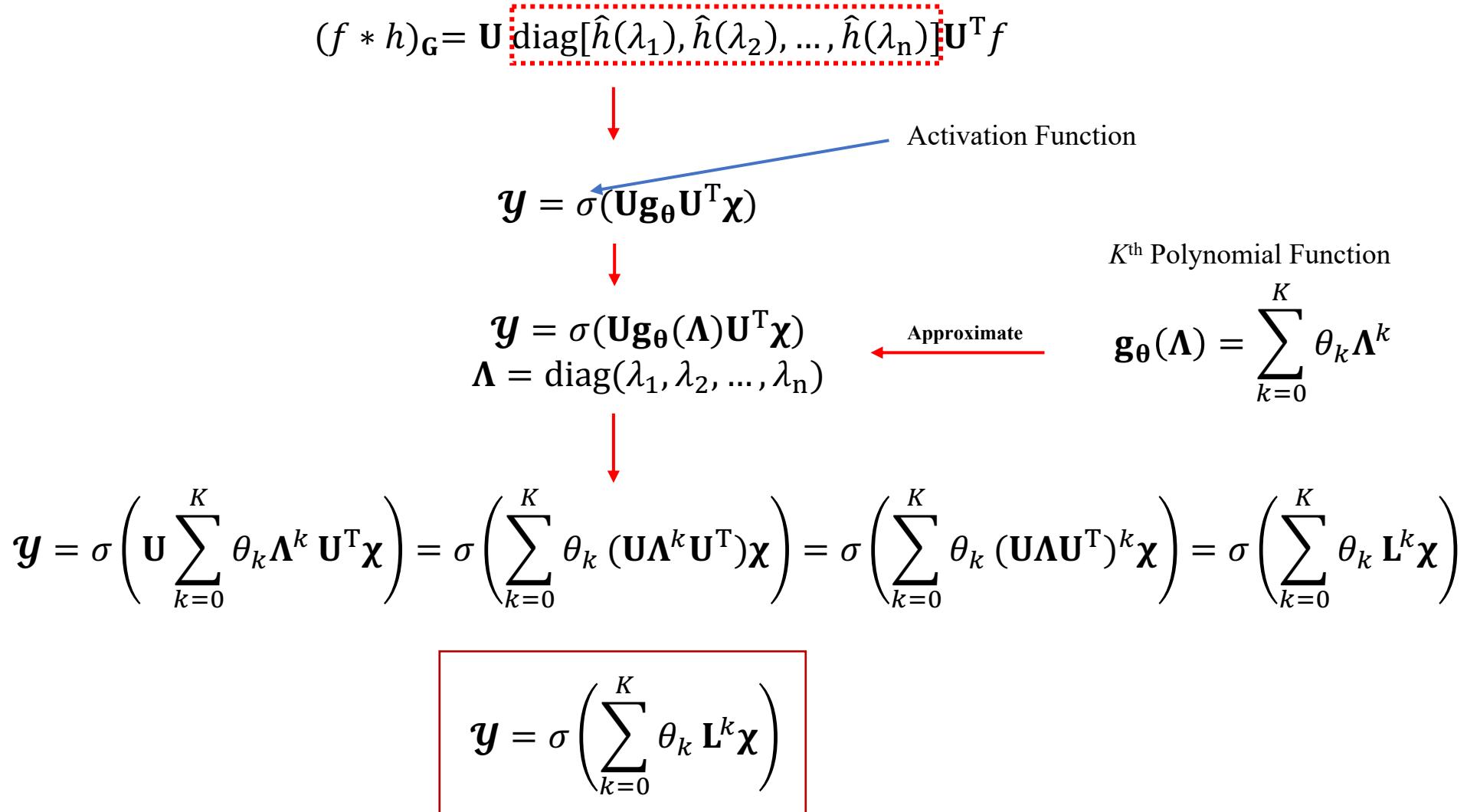
$[n \times n]$

$[n \times n]$

$[n \times n]$

$$(f * h)_G = \mathbf{U} \text{diag}[\hat{h}(\lambda_1), \hat{h}(\lambda_2), \dots, \hat{h}(\lambda_n)] \mathbf{U}^T f \quad [n \times d]$$

Graph Convolution



Graph Convolution

Node Aggregation
 K is Filter Size

Convolution:
Weighted Sum

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

Weight Sharing

No need for Fourier Transform

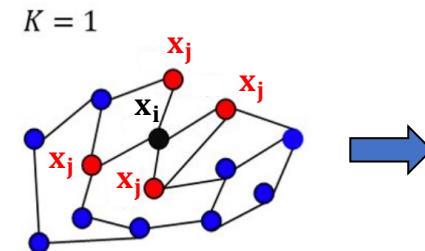
Beauty is in Simplicity

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

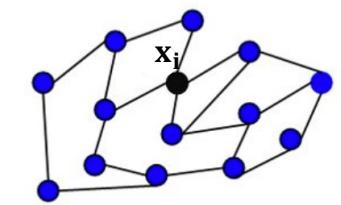
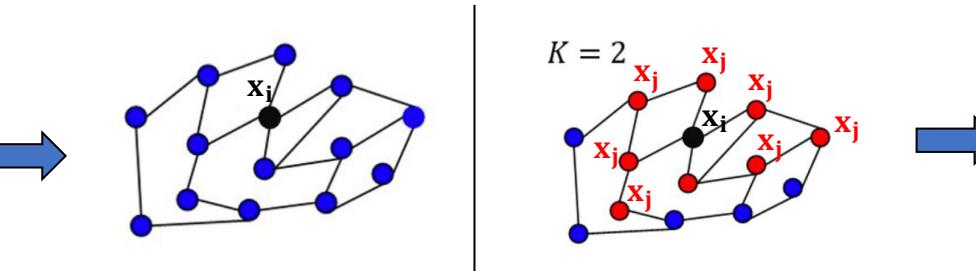
Laplace Operator

Local connectivity

$$\mathbf{x}_{\text{new}} \leftarrow \mathbf{L} \mathbf{x}_i = \sum_j A_{ij} (\mathbf{x}_i - \mathbf{x}_j)$$



Localize in Space



Pros:

1. No need for Spectral Decomposition of \mathbf{L}
2. Less number of parameters (decrease model complexity) $\rightarrow K \ll N$

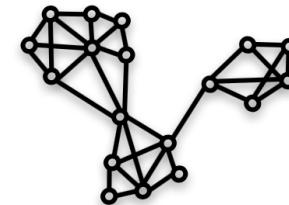
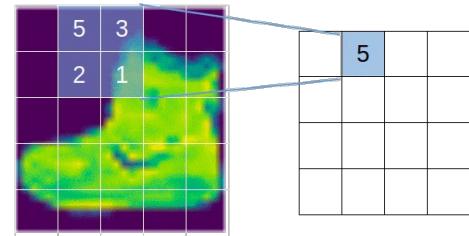
Cons: Need to compute \mathbf{L}^k

Pooling on Graphs (Graph Coarsening)

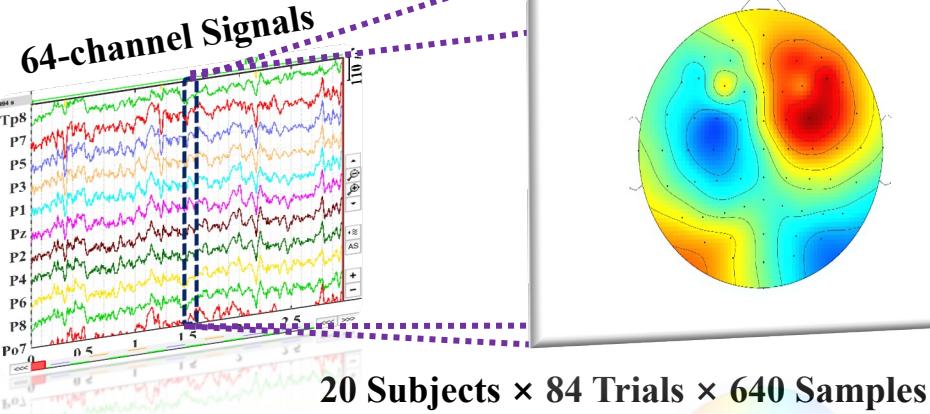
- Traditional CNN doesn't need to consider **neighbors** after convolutions
 - [Euclidean Structure] The output Feature Maps are “**regular**”
 - The neighbor is “**meaningful**”
- GCNs need to consider neighbors after convolutions
 - [Non-Euclidean Structure] The output **graphs' nodes are not arranged in any meaningful way**
 - Use **Graclus Multilevel Clustering Algorithm** to find “meaningful” neighbors
 - Minimize the **Local Normalized Cut** (a cluster grouping method)

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

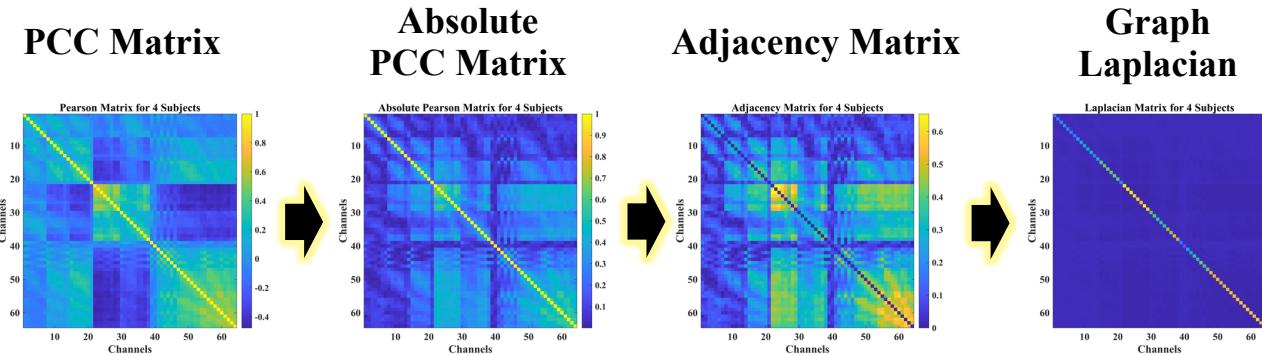
- i and j denote node i and node j
- W_{ij} is the **learned weight** between node i and node j



(i) EEG Data Acquisition

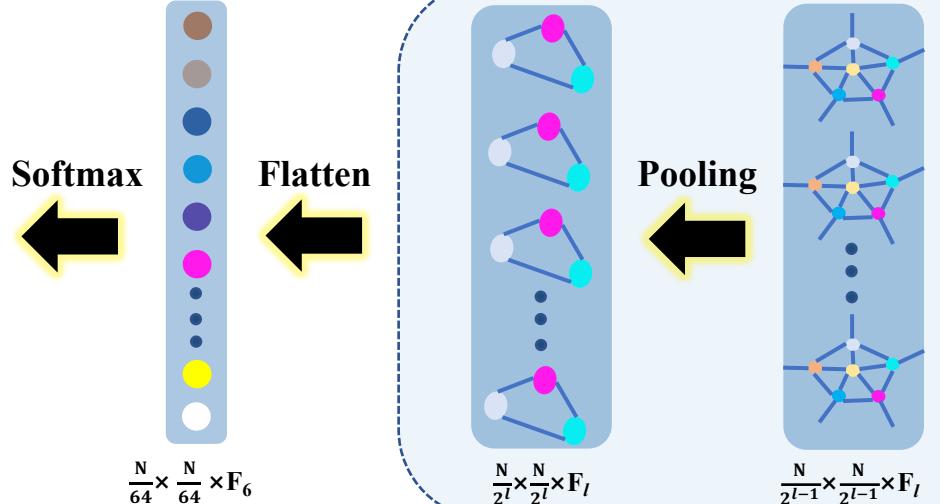


(ii) Correlations between EEG Electrodes

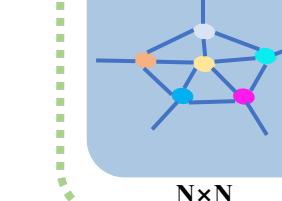
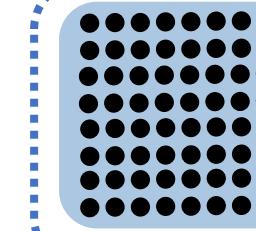


Real-time 64-channel Raw EEG Signals

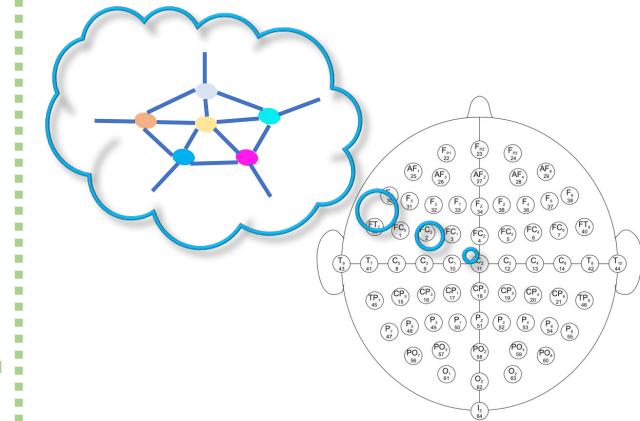
(iv) The GCNs-Net



GCN



(iii) Graph Representation



Correlation among EEG electrodes

Two Subjects: Subject 10 and 5

Problem: Individual Variability

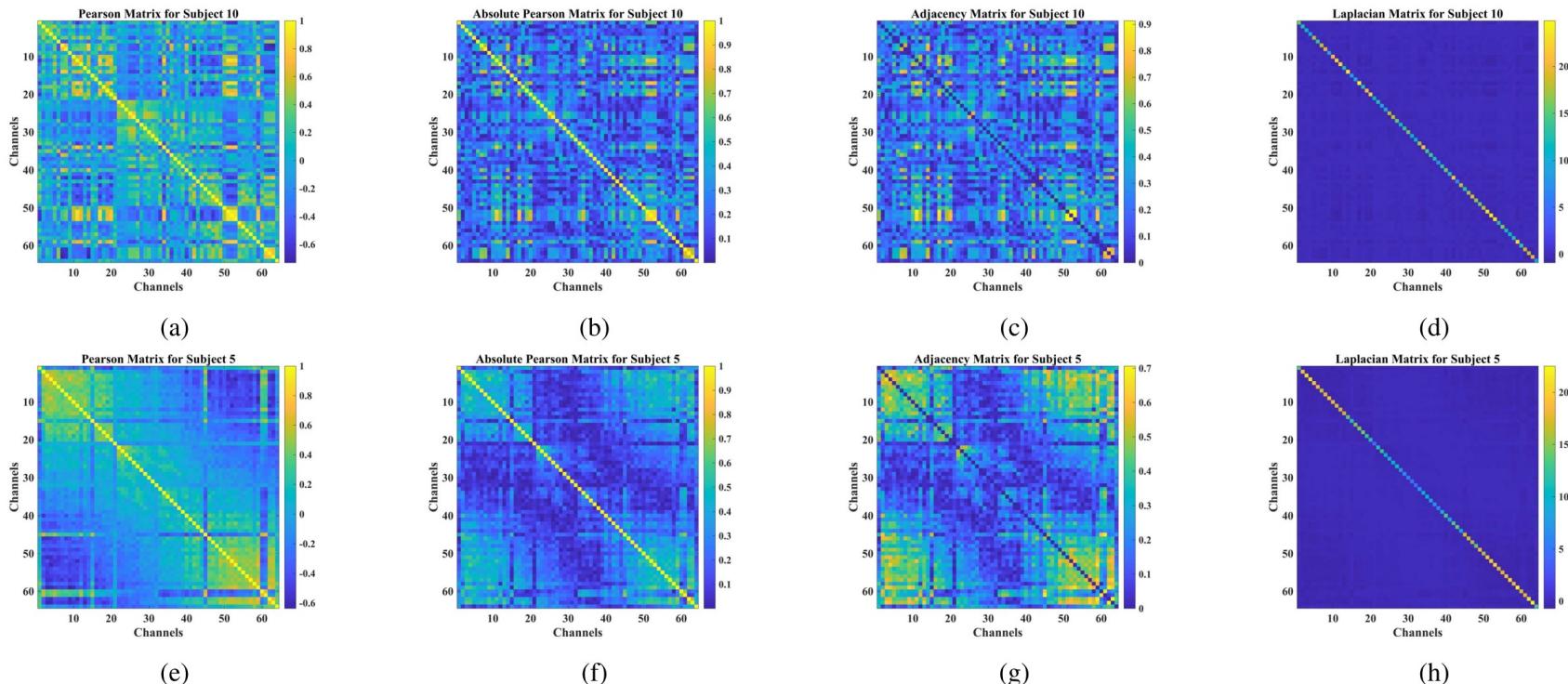


Fig. 6. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for Subjects 10 and 5 from the PhysioNet dataset. (a) PCC matrix for Subject 10. (b) Absolute PCC matrix for Subject 10. (c) Adjacency matrix for Subject 10. (d) Graph Laplacian for Subject 10. (e) PCC matrix for Subject 5.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

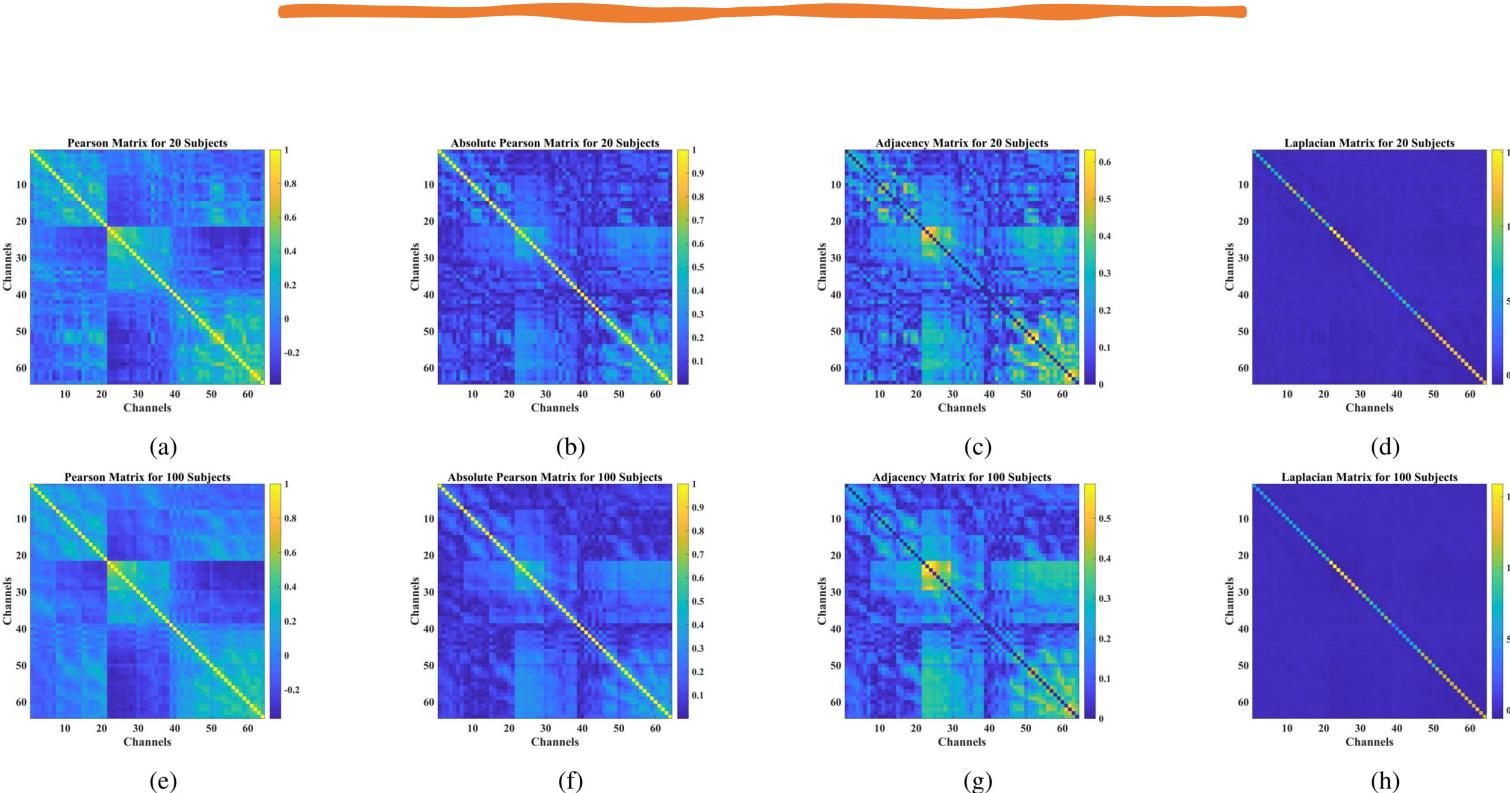


Fig. 2. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for 20 and 100 subjects, respectively, from the PhysioNet dataset. (a) PCC matrix for 20 subjects. (b) Absolute PCC matrix for 20 subjects. (c) Adjacency matrix for 20 subjects. (d) Graph Laplacian for 20 subjects. (e) PCC 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 number 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 Stochastic Gradient Descent (SGD) algorithm
 - Learning Rate: 0.01
 - Batch Size: 1,024
- **Activation Function:** Softplus (Smooth Rectified Linear Unit)

$$F(x) = \log(1 + e^x)$$

- **Model Output:** Softmax: y are labels, \hat{y} are the final output tasks

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

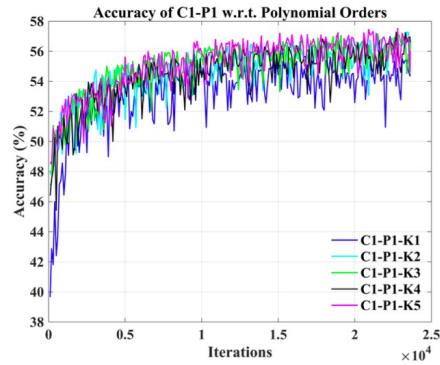
- **Loss Function:** Cross-entropy Loss with L2 regularization

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

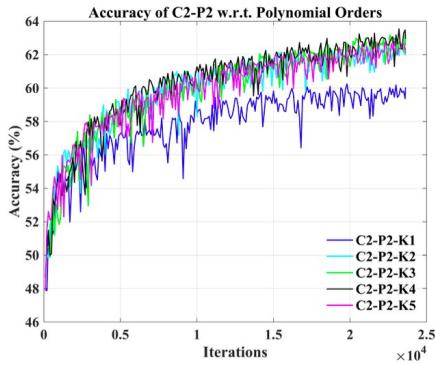
$\lambda = 1 \times 10^{-6}$ is the coefficient of the L2 regularization.

Ablation Study

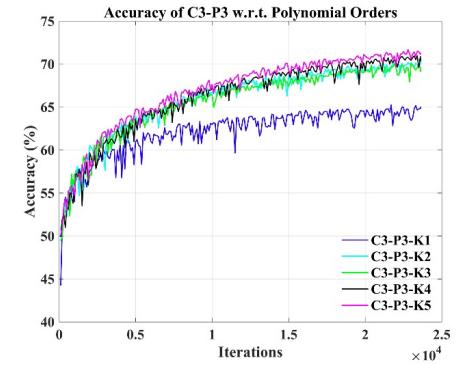
K1
Poor Performance



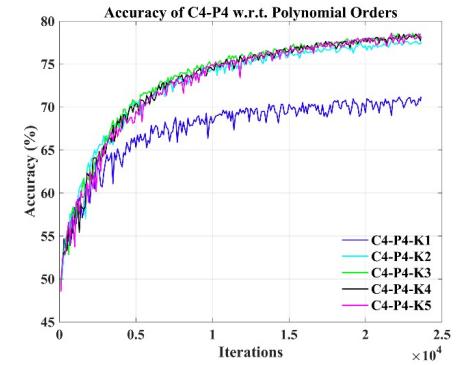
(a)



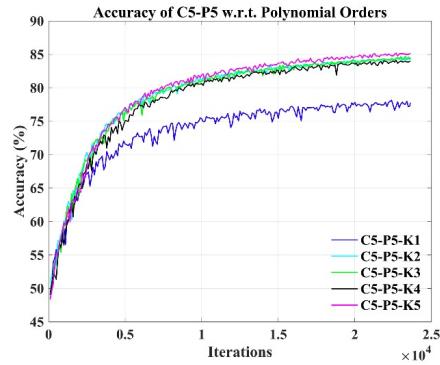
(b)



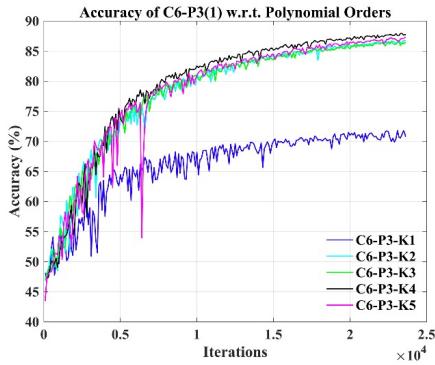
(c)



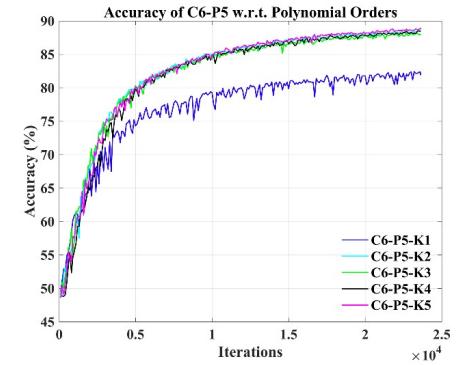
(d)



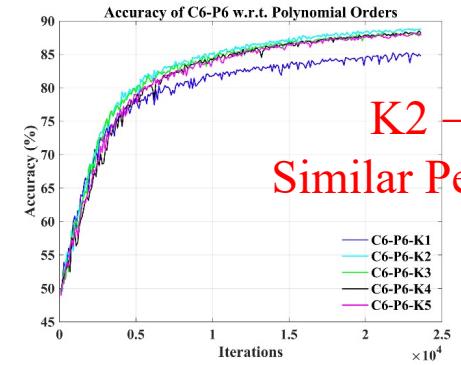
(e)



(f)



(g)



(h)

K2 → K5
Similar Performance

Fig. 3. Accuracy of some selected models regarding different polynomial approximation order. The models are selected from Table II. (a) Accuracy of the model C1-P1 (model 1). (b) Accuracy of the model C2-P2 (model 3). (c) Accuracy of the model C3-P3 (model 6). (d) Accuracy of the model C4-P4 (model 10). (e) Accuracy of the model C5-P5 (model 14). (f) Accuracy of the model C6-P3 (model 16). (g) Accuracy of the model C6-P5 (model 19). (h) Accuracy of the model C6-P6 (model 20).

Experimental Results

Groupwise Prediction and Subject-specific Adaptation

TABLE IV
PERFORMANCE COMPARISONS ON THE PHYSIONET DATASET

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

Note: ***p*-value < 0.05** → Statistically Significant Difference

Takeaways and Future Work

- ✓ [Graph Representation]

Graph Representation Learning to deeply extract **Network Patterns of Brain Dynamics** for EEG classification.

- ✓ [Model Converge]

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**.

- ✓ [Future Work]

Modeling EEG signals as **Dynamic Graphs** and processing them via **Dynamic Graph Representation Learning**.

Deep Feature Mining via Attention-based BiLSTM-GCN

for Human Motor Imagery Recognition

Yimin Hou¹, Shuyue Jia^{1, 2 *}, Xiangmin Lun¹, Shu Zhang³, Tao Chen¹, Fang Wang¹, and Jinglei Lv⁴

¹ School of Automation Engineering, Northeast Electric Power University

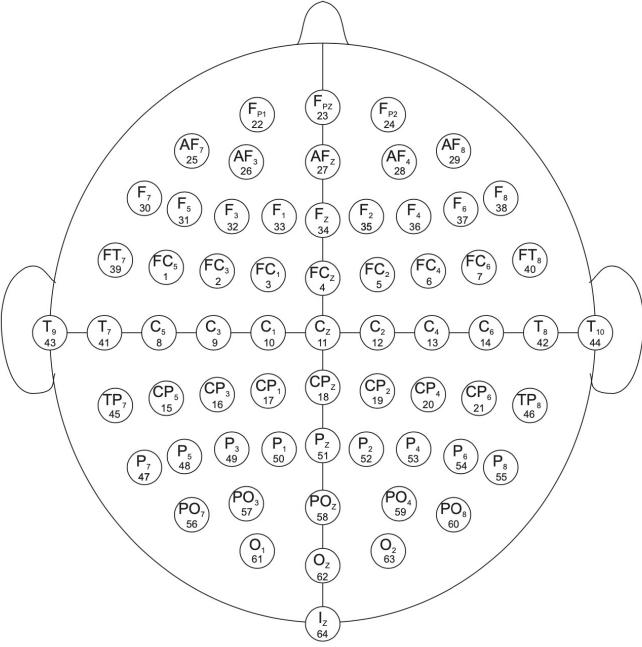
² Department of Computer Science, City University of Hong Kong

³ School of Computer Science, Northwestern Polytechnical University

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

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

One Problem of the GCNs-Net



Spatial information

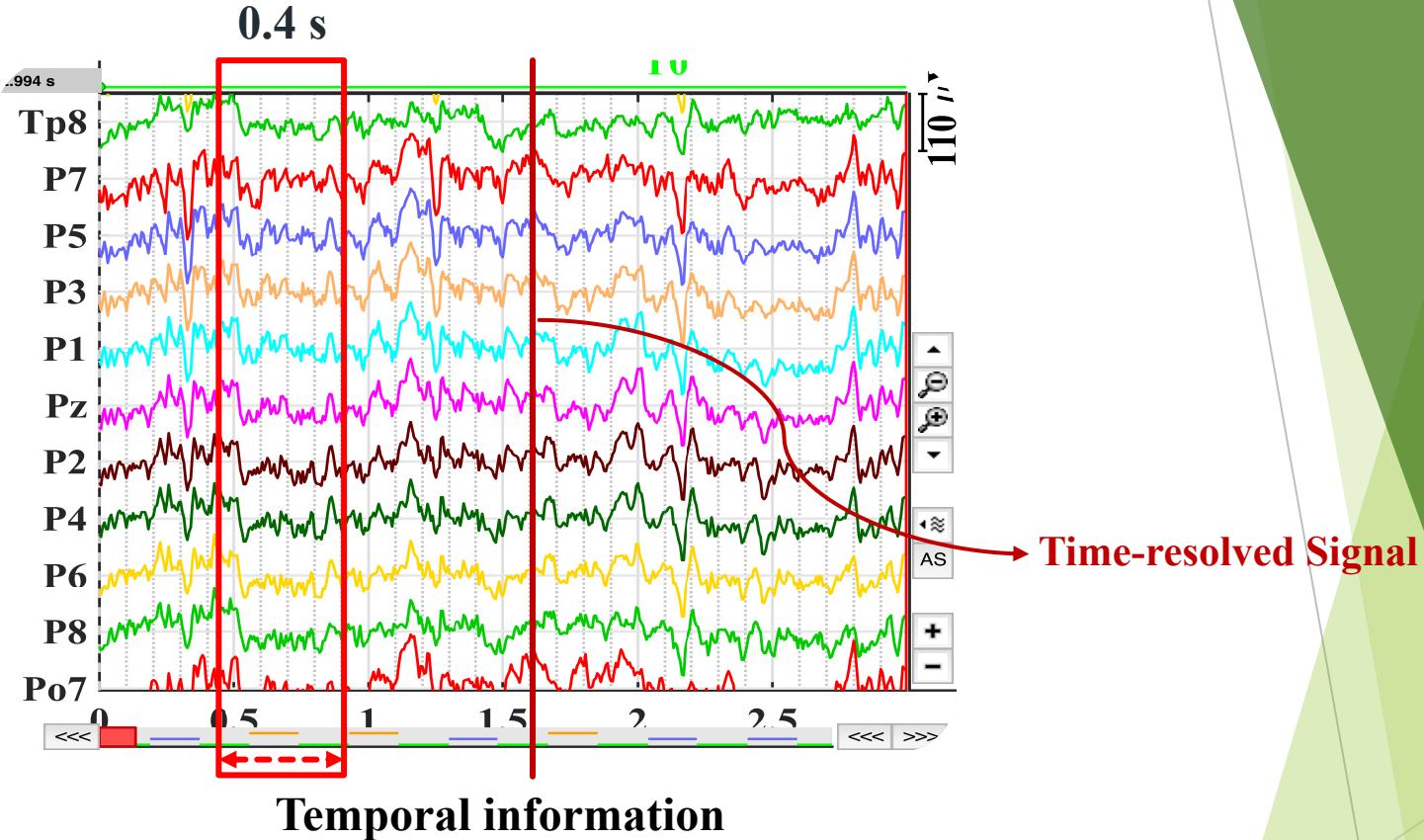
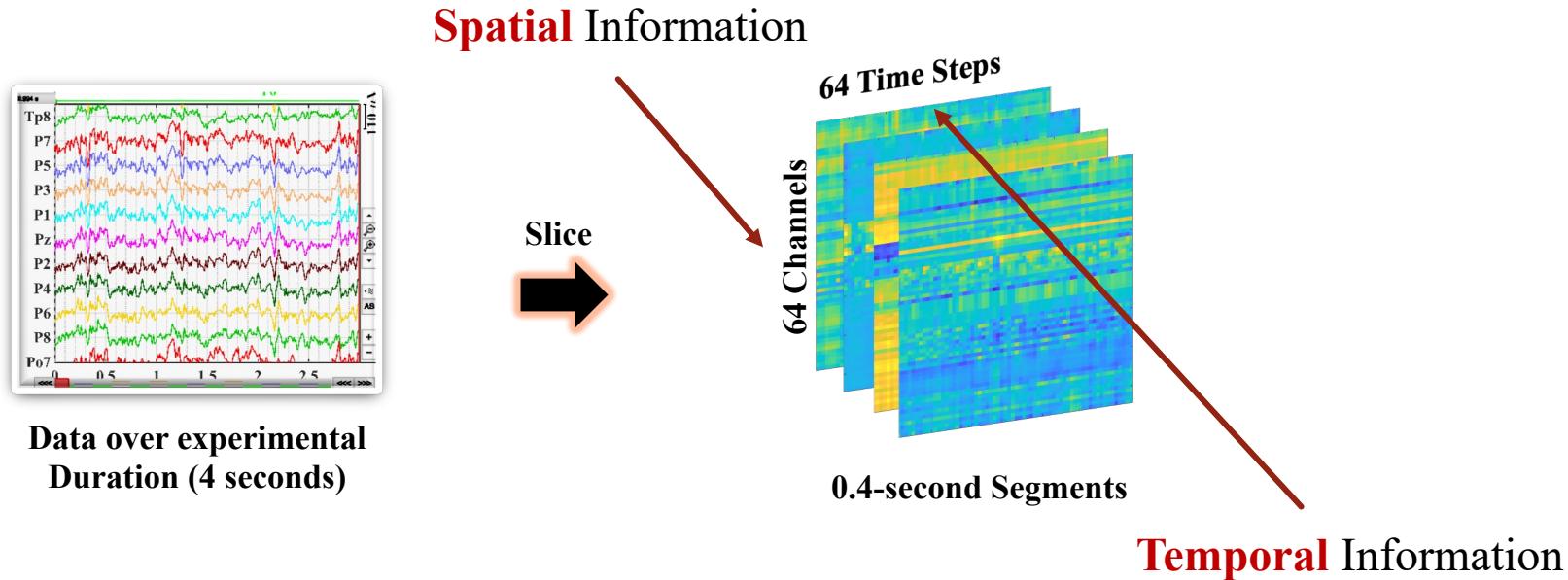


Image Credit: The PhysioNet Dataset.

Motivation:

- ✓ GCNs-Net is based on **Time-Resolved Signal** → doesn't consider **Temporal Information**
- ✓ [Spatial-Temporal Analysis] Consider **Temporal** and **Spatial Information** from EEG signals
- ✓ [Responsive] Maintain **High Responding Time**

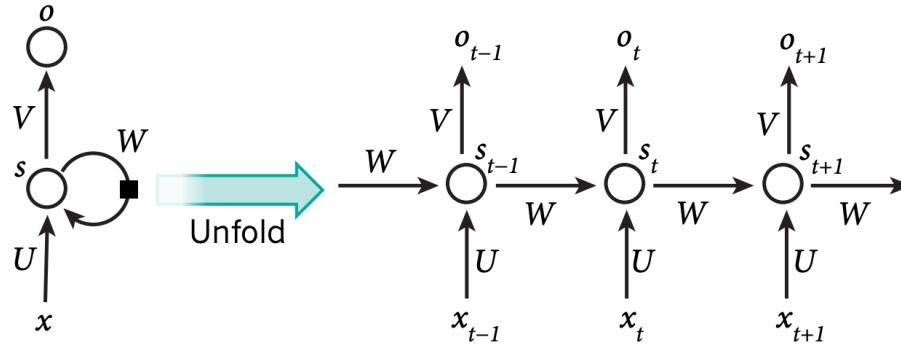
64-channel Raw EEG Signals Acquisition



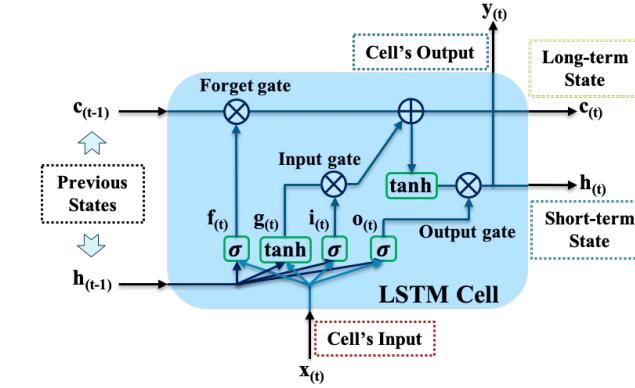
- ✓ 4-s (experimental duration) Signals → **0.4-s segments** over time
- ✓ Each Segment: **64 channels × 64 time steps**
- ✓ Pre-processed Data: **Temporal** Information + **Spatial** Information

Temporal Information Extraction

unrolling the network through time



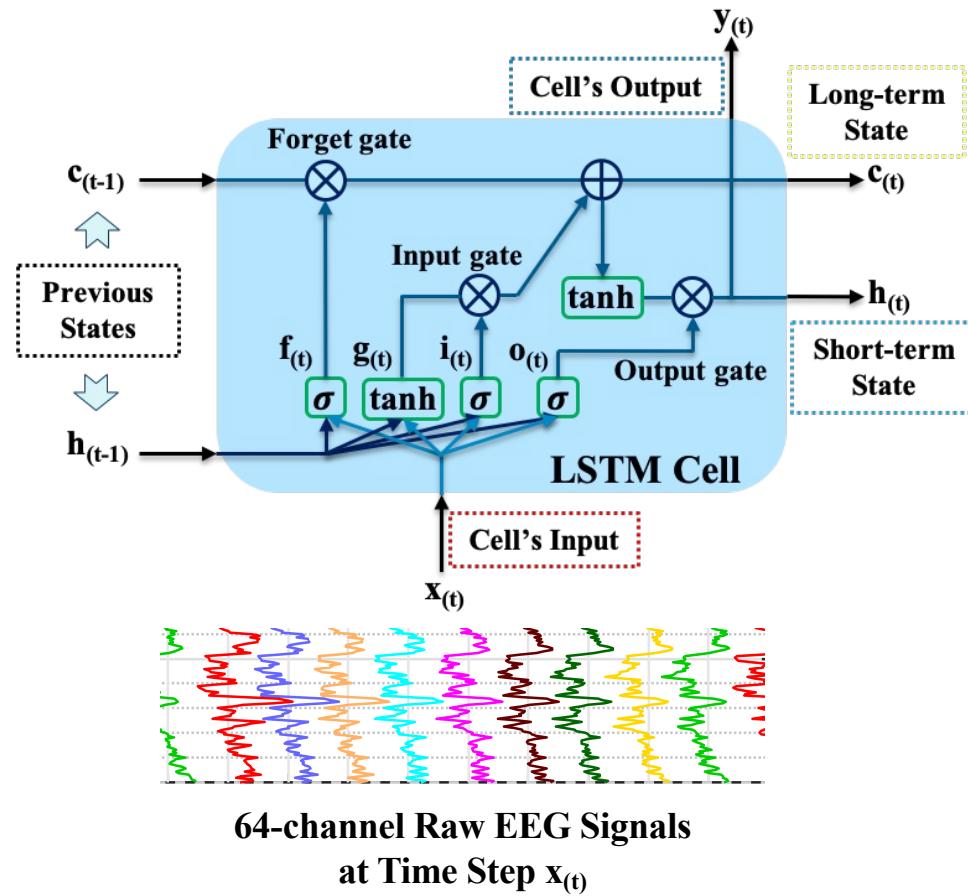
Recurrent Neural Network (RNN)



Long Short-Term Memory (LSTM)

- ✓ Designed for ***order-mattered sequential data***, e.g., time series
- ✓ 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**
- ✓ Gated Recurrent Units (**GRU**): **lightweight** architecture with comparable performance

Long Short-term Memory (LSTM)



- ✓ Capture **Long-range Dependencies**
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
→ 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 or Outputs
Equally treated/contributed

vs.

Differently treated/contributed with preference/importance

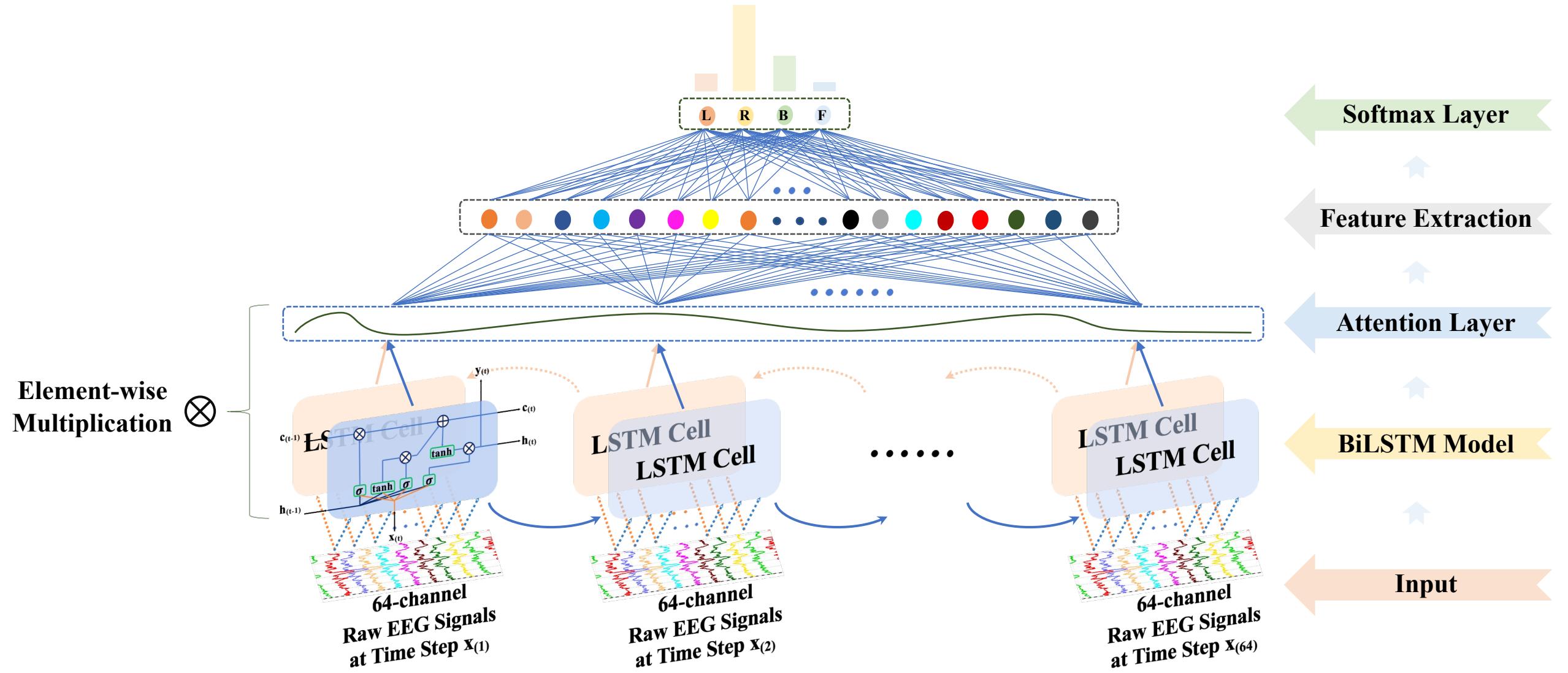
FC Layer $\mathbf{U}_t = \tanh(\mathbf{W}_w \mathbf{y}_t + \mathbf{b}_w)$

Attentional
Weights $\alpha_t = \frac{\exp(\mathbf{U}_t^T \mathbf{W}_U)}{\sum_t \exp(\mathbf{U}_t^T \mathbf{W}_U)}$

Weighted
Sum

$$\hat{\mathbf{U}}_t = \sum_t \alpha_t \mathbf{y}_t$$

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



Model Design Ablation Study

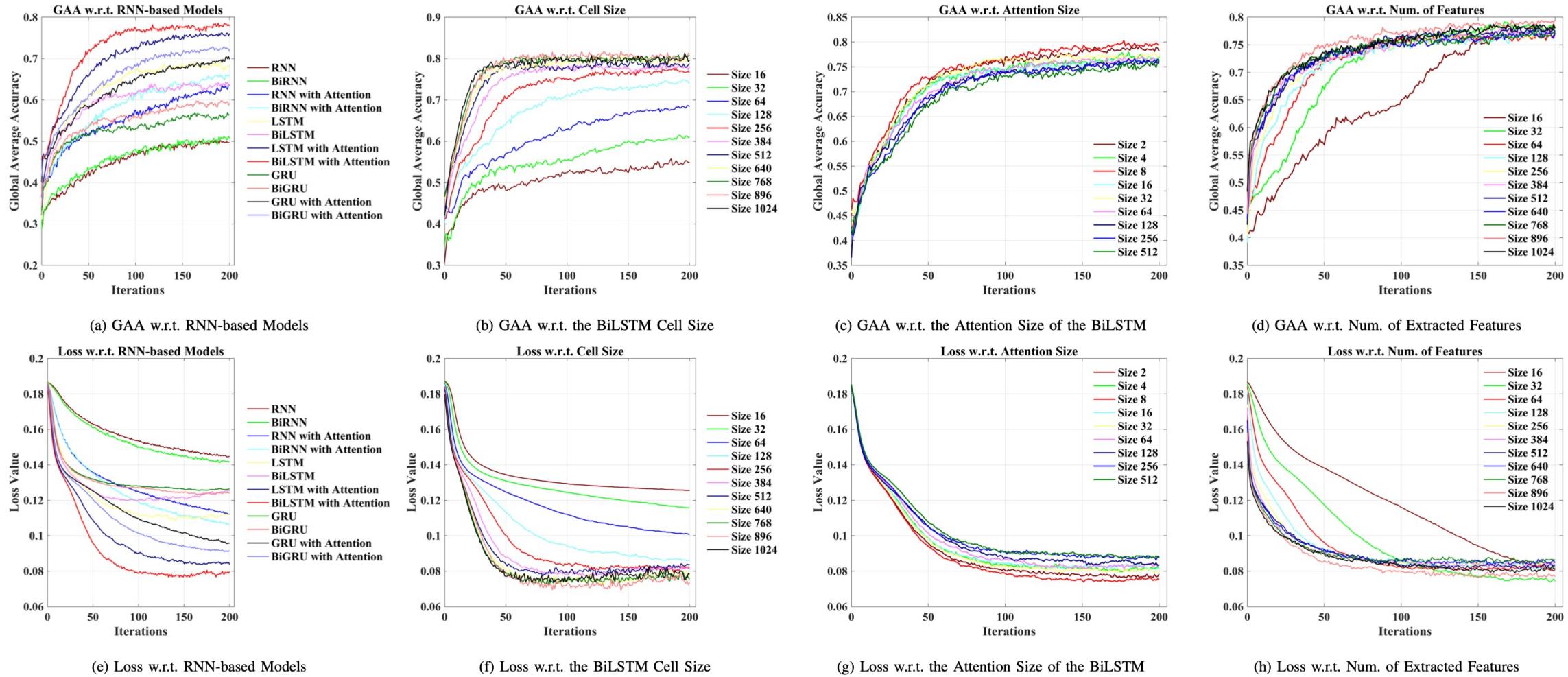
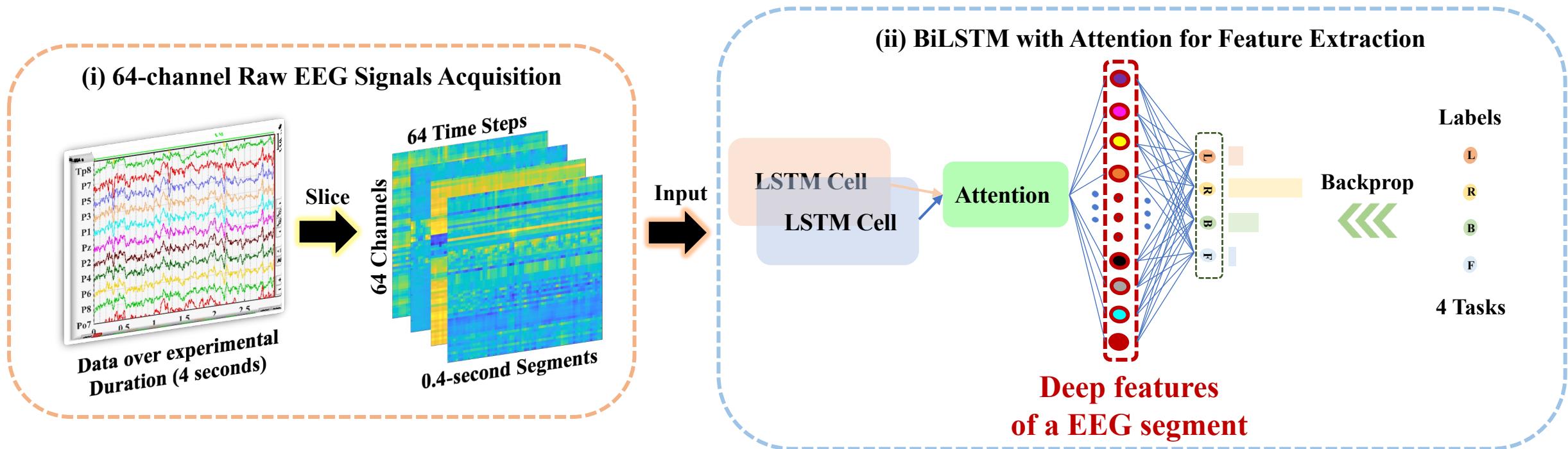


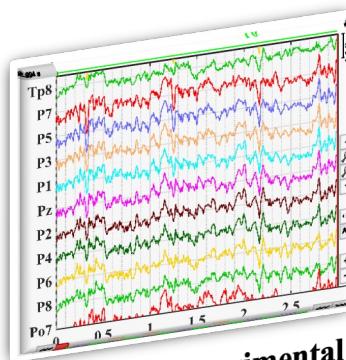
Fig. 3: Models and Hyperparameters Comparison w.r.t. the RNN-based Methods for Feature Extraction

Topological Structure of Features

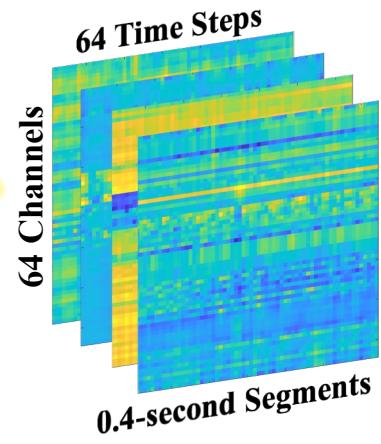


✓ Deep Feature Mining → Intra-feature Relationship → Intra-feature Modeling

(i) 64-channel Raw EEG Signals Acquisition



Slice



Data over experimental Duration (4 seconds)

(ii) BiLSTM with Attention for Feature Extraction

LSTM Cell
LSTM Cell

Attention

Labels

L

R

B

F

Backprop



4 Tasks

Intra-feature Modeling

(iii) Graph Convolutional Neural Network

Labels

L

R

B

F

Backprop

Softmax

Flatten

Max Pooling

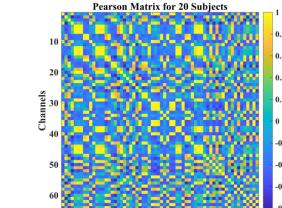
GCN

Features



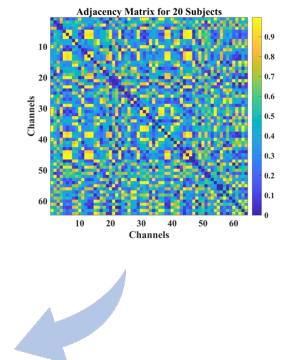
4 Tasks

Pearson Matrix

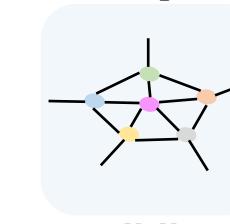


Intra-feature Relationship

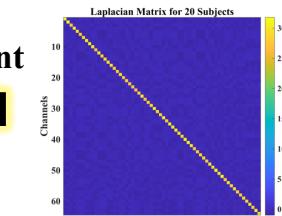
Adjacency Matrix



Graph



Laplacian Matrix



Present

N x N

Topological Structure of Features

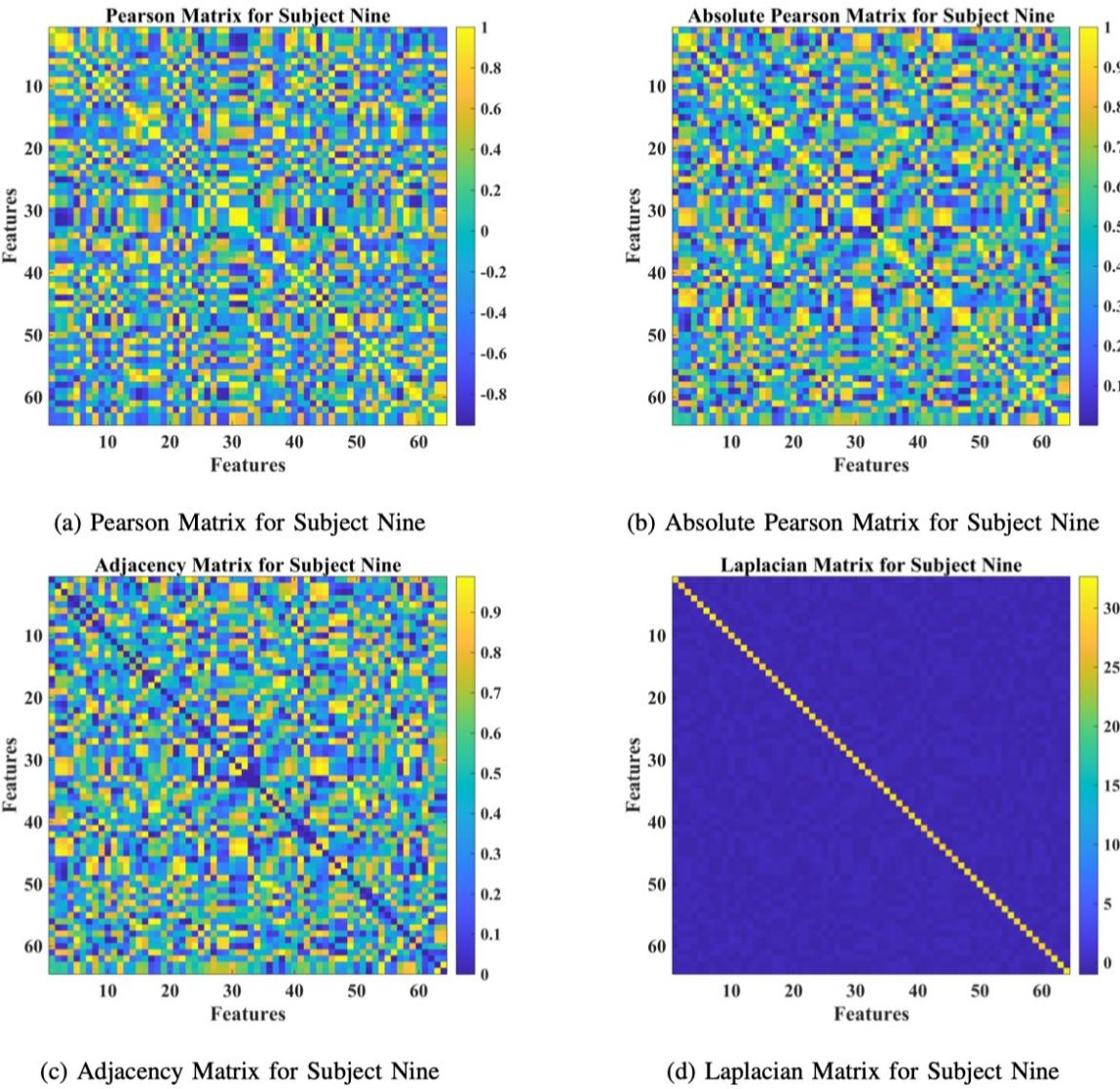
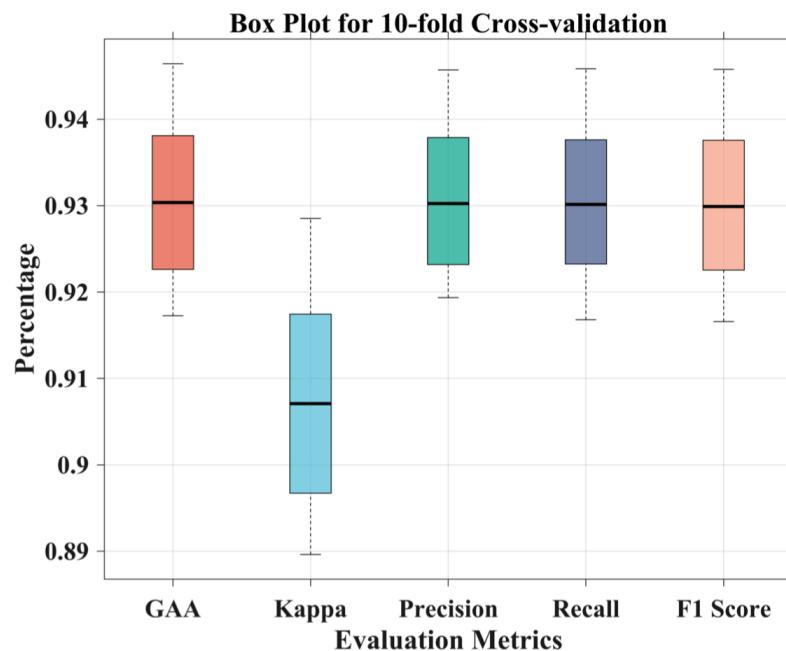
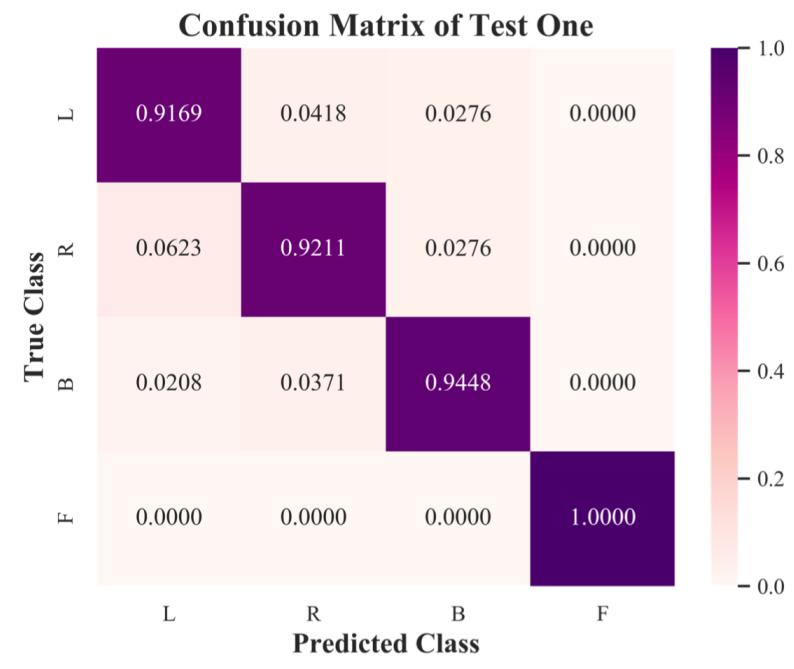


Fig. 4: The Pearson, Absolute Pearson, Adjacency, and Laplacian Matrices for Subject Nine.

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.

Note:

- (1) Box Plot (Maximum Score, Upper Quartile, Median, Lower Quartile, and Minimum Score)
- (2) Confusion Matrix: TP, TN, FP, and FN

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

Takeaways and Future Work

✓ [Spatial-Temporal Analysis]

- (1) Converge to both Subject-level and Groupwise Predictions and handle Individual Variability.
- (2) The 0.4-s sample size Time-Resolved Solution toward fast response.

✓ [Deep Feature Mining]

- (1) ↑ Highest Accuracy
- (2) Advance the Clinical Translation of the EEG MI-based BCI technology to meet diverse demands, such as those of paralyzed patients.

✓ [Future Work]

Long-range Dependencies among intra-subject or inter-subject EEG signals can be modeled via **Non-local Modeling**, **Self-attention Mechanism**, **Transformer**, and **AI foundation Models**.

Thank you!

Any question?