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

School of Automation Engineering, NEEPU, China
Shuyue Jia
shuyuej@ieee.org
Supervisor: Yimin Hou

Research Background

- **BCI**: connects the brain with machines, **acquires and analyzes brain signals** regarding actual or imagery tasks, and then **commands machines**.
- **Significance**: help the disabled (e.g., strokes) and understand our brains.
- **Types of BCI**:
  - Electroencephalography (EEG)
  - Magnetoencephalography (MEG)
  - Functional Magnetic Resonance Imaging (fMRI)
  - Invasive BCI Technologies (e.g., Neuralink)
  - etc.
- **Reasons for using EEG for this project**:
  - Non-invasiveness
  - High Temporal Resolution
  - Portability
  - Inexpensive Equipment
  - etc.

  Potentially have a broad market.

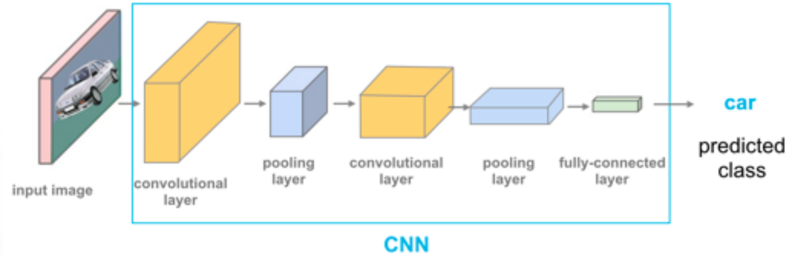
- **Related Applications**: Wheelchair (Nature Machine Intelligence, 2019), Spoken Sentences (Nature 2019)
- **Specific Task**: EEG Motor Imagery Tasks Classification (e.g., control a wheelchair through only brain signals)
- **Our Goal**: develop **EEG-based BCI applications** that could potentially be used to improve the current stroke rehabilitation strategies.

Difficulties in dealing with EEG Signals

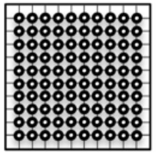
- **Individual Variability → Lower Classification Accuracy**
  - Low Signal-noise ratio
  - Different brain electrical conductivity ← different anatomical structure of brain
  - Electrodes' position error
  - etc.
- **Slow Real-time Responding → Hard to develop Real-life Applications**
  - Trial-level prediction (e.g., 4 s) (most literature)
  - Window-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**
  - Subject-level prediction (most literature) (Our Work)
  - Group-level prediction (Our Work)

Intuition & Motivation

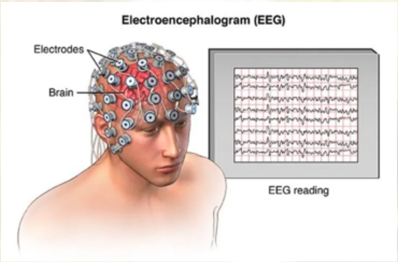
- Traditional CNN-based approaches:

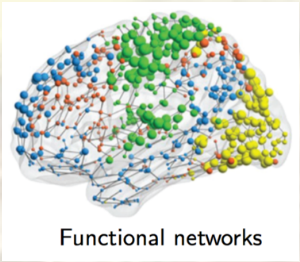


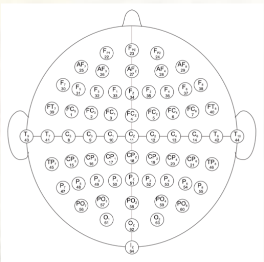
  - Most recent published literature in the field of EEG (MI) introduced **CNN-based approaches**.
  - Check our previous work *(Journal of Neural Engineering, SCI, IF=4.551)*, which used CNN method and achieved competitive results (**96%** accuracy at the subject level, and **94.50%** at the group level (10 subjects)).
  - **Local connectivity, weights sharing, translation invariance, hierarchical, low dimensionality, etc.**
  - **Implemented on the Euclidean-structured data** (e.g. Image, voice, natural languages)



- Neuroscience research has increasingly emphasized **Brain Network Dynamics**.
  - The **functional topological connectivity** of EEG electrodes → Graph instead of Euclidean structure





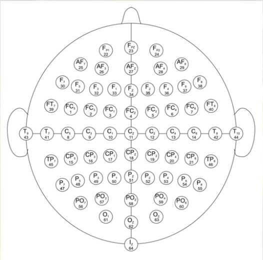


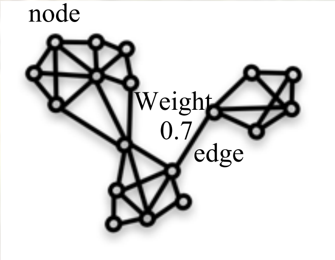
Our Question:

Can we implement CNNs on graphs directly?

Can we use Traditional CNNs on graphs directly?

- Traditional CNNs **cannot** process graphs directly:
  - **Graphs are irregular** (1. unordered, 2. vary in size)
  - → Convolution **cannot** keep **translation invariance** on the non-Euclidean signals
- Can we implement CNNs on Graphs? → Graph Convolutional Neural Networks (GCNs / Graph CNN)
  - Can process Graph-structured Signals directly
  - Consider the relationship properties (e.g., correlations) between nodes
  - Consider the functional topological relationships of EEG electrodes

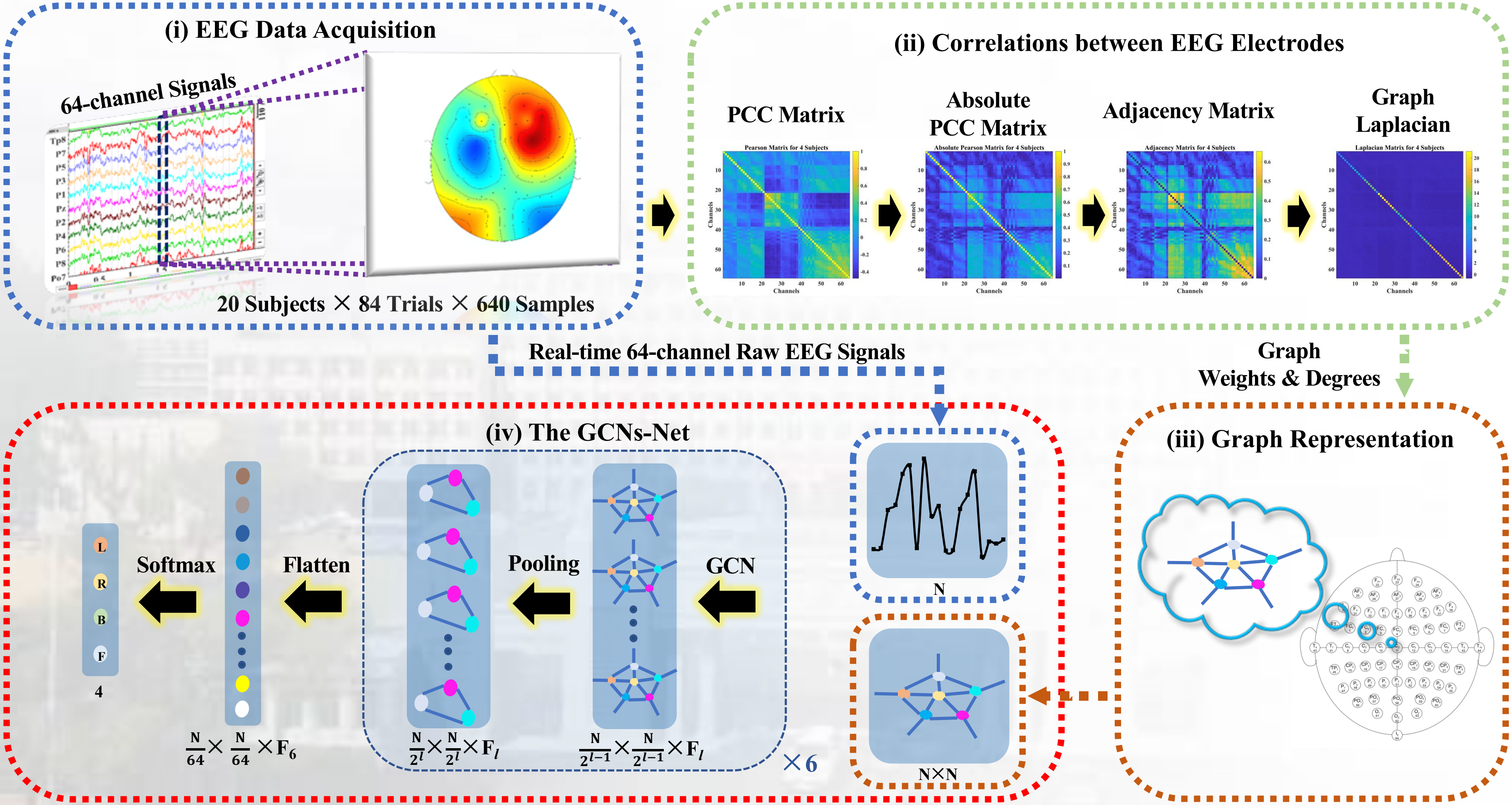




Our Question:

How to implement CNNs on graphs?
(How to implement GCNs?)

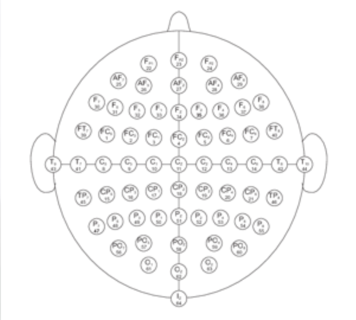
Our presented **GCNs-Net** for EEG Signals Classification

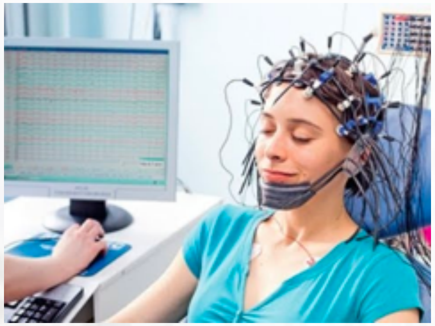


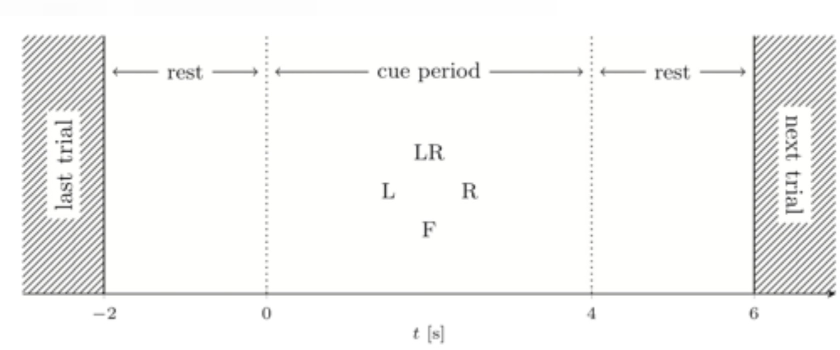
Codes available at https://github.com/SuperBruceJia/EEG-DL.

Benchmark Dataset Description

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

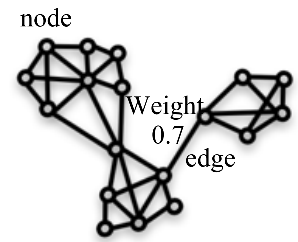






- 109 subjects (largest number of participants in the field of EEG MI)
- 4-class EEG Motor Imagery Classification
  - Imagining left fist, right fist, both fists, and both feet
- For each subject, 3 runs, 7 trials, 4 classes → 84 trials in total
- For each trial, 4 seconds experimental duration, 160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

Graph Representation: **Laplacian Matrix** in Graph Theory



- Undirected and Weighted Graph: $G = \{V, E, A\}$
  – V: nodes, $|V| = N$
  – E: edges that connected nodes
  – A: weights / correlations between nodes

  1. Weights
  2. Degrees
- Correlations representation: Pearson Matrix
  – Measure the linear correlations between nodes
  – Below, μ is the expectation, σ is the standard deviation, and $P_{x,y}$ is the Pearson Correlation Coefficient (PCC) between two nodes

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

  – Absolute Pearson Matrix: $|P_{x,y}|$
- Graph Weights representation: Adjacency Matrix: $A = |P_{x,y}| - I$, where I is an Identity Matrix
- Graph Degrees representation: Degree Matrix

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

- Graph representation: Graph Laplacian (Laplacian Matrix, Combinatorial Laplacian)

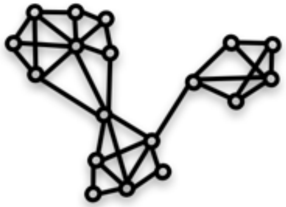
$$L = D - A$$

- Normalized Graph Laplacian:

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

Convolutions on Graphs (*Spectral Graph Filtering*):

- Convolutions on graphs in the spatial domain
  - No convinced mathematical definition
  - Hard to match local neighborhoods
- Convolutions on graphs with the spectral graph theory
  - Have a solid mathematical definition
  - Have a well-defined localized operator on graphs



- Spectral Theorem: Fourier Transform for graphs, i.e., for the graph Laplacian
  - U: Fourier basis, which is a complete set of orthonormal eigenvectors of L
  - Λ: a Diagonal Matrix, where the diagonal is the ordered real nonnegative eigenvalues of L

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

- Fourier Transform of Signal x

$$\hat{x} = U^{T}x$$

- Spectral filtering of graph signal x (feature vector of graph nodes)

$$y = g_{\theta}(L)x = g_{\theta}(U\Lambda U^{T})x = U g_{\theta}(\Lambda) U^{T}x$$

- A non-parametric filter, i.e., a filter whose parameters are all free, would be defined as

$$g_{\theta}(\Lambda) = \mathrm{diag}(\theta)$$

- θ is a vector of Fourier coefficients.

Problem 2: No Local connectivity

Convolution in the spatial domain

frequency domain

Spatial domain

Problem 1

non-parametric filter

not localized in space

Solution:

Polynomial approximation

Convolutions on Graphs (*Spectral Graph Filtering*):

- One commonly used filter is the Chebyshev polynomial
  - K^{th} Chebyshev polynomial

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$
$$T_0 = 1$$
$$T_1 = x$$

  - The filter after approximation

$$g_\theta(\Lambda) = \sum_{k=1}^{K} \theta_k \Lambda^k$$

  - Defined Convolutional Operation

$$y = U\left(\sum_{k=1}^{K} \theta_k \Lambda^k\right)U^T x = \left(\sum_{k=1}^{K} \theta_k \overset{L^k}{\boxed{U\Lambda^k U^T}}\right)x = \left(\sum_{k=1}^{K} \theta_k\, L^k\right)x = \boxed{\sum_{k=1}^{K} \boxed{\theta_k} \boxed{(L^k x)}}$$

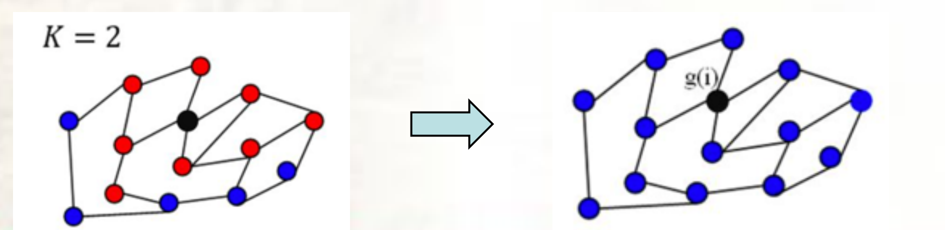
$$x_{new} \leftarrow Lx_i = \sum_j A_{ij}(x_i - x_j)$$

Convolution: Weighted Sum

Weights Sharing → Translation Invariance

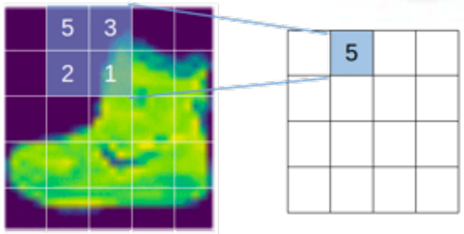
Local connectivity
No need for Fourier
$O(n^2) \rightarrow O(n)$

  - θ are the trainable parameters ← Back-propagation Algorithm
  - K-hop Neighbor features divergence/average, K is the size of the repetitive field
  - GCN Key Idea: Use "edge information" to "aggregate" "node information" to generate a new "node representation"



Pooling on Graphs (*Graph Coarsening + 1D Pooling*)

- Traditional CNNs **don't** need to consider neighbors after convolutions
  - The output feature maps are regular (Euclidean Structure)
  - The neighbors are "meaningful"



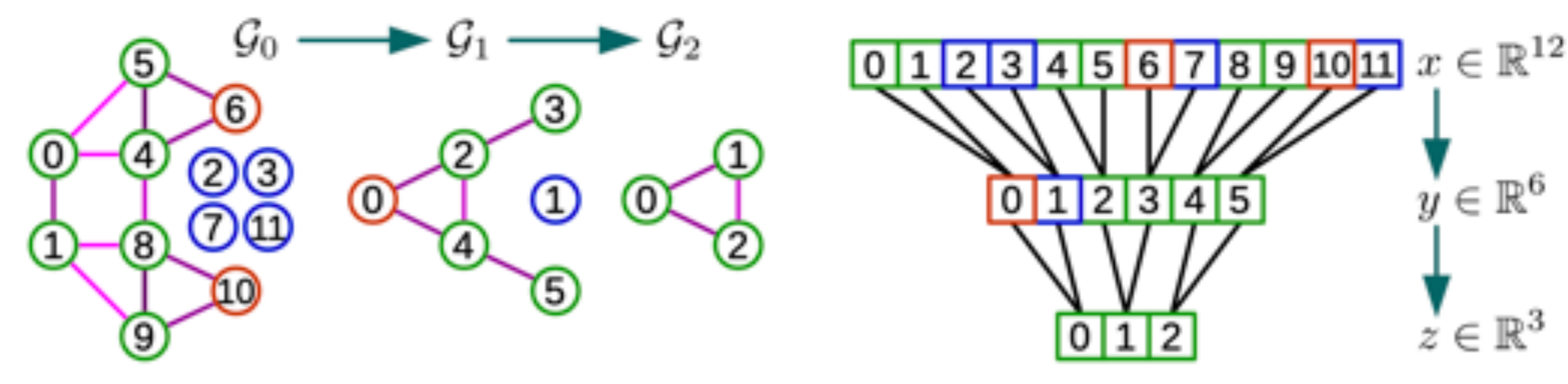
- GCNs need to consider neighbors after convolutions
  - The output graphs' nodes are not arranged in any meaningful way
  - So, we have to find meaningful neighbors of the graph nodes after convolution to carry out pooling
  - We will use Graclus multilevel clustering algorithm, a cluster algorithm to find meaningful neighbors
  - a.k.a. Graph Coarsening
  - Minimize the local normalized cut

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

  - i and j are two nodes. After the Coarsening, W_{ij} will be their new weight

Pooling on Graphs (*Graph Coarsening + 1D Pooling*)

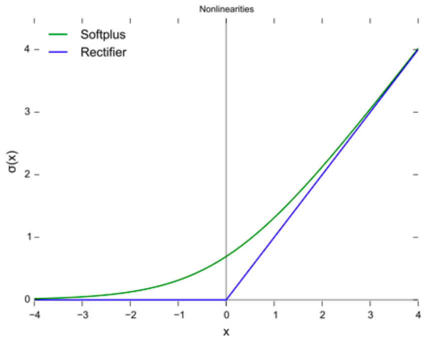
- Along the way, a balanced binary tree was used to store nodes of the coarsened graph
  - Memory Efficient
- Then carry out one-dimensional pooling
- If there is singletons (non-matched nodes) → Cannot pool based on a size two → We will use a fake node.



Model Initialization

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

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



- Softmax Output: y is the label, ŷ is the final output probability

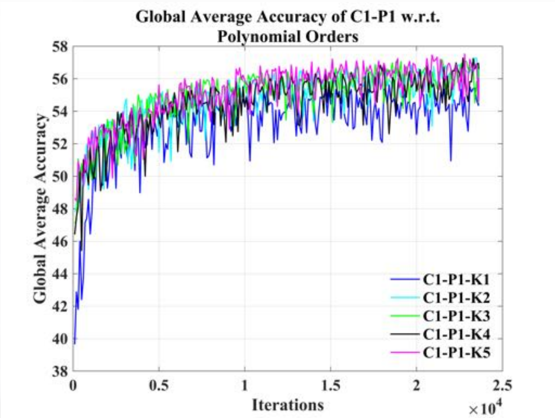
$$\hat{y} = \operatorname{argmax}\left(\frac{e^{y_i}}{\sum_{i=1}^{4} e^{y_i}}\right)$$

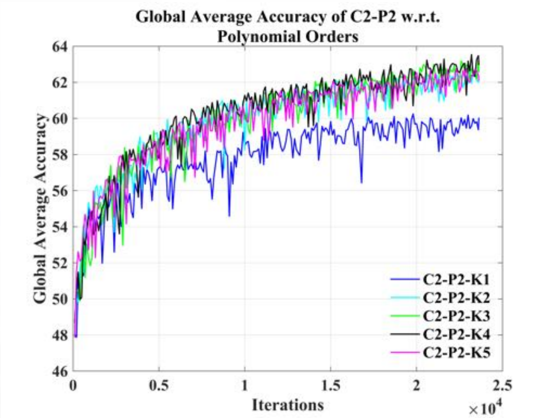
- Loss Function: Cross-entropy with L2 regularization

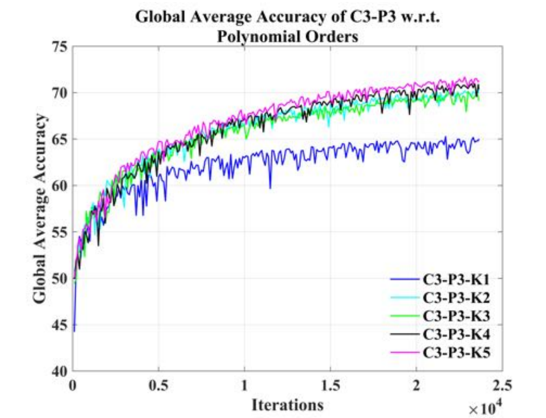
$$\text{Loss} = -\sum_{i=1}^{4} y_i \log(\hat{y}_i) + \lambda\left(\sum_{j=1}^{n} w_j^2 + b_j^2\right)$$

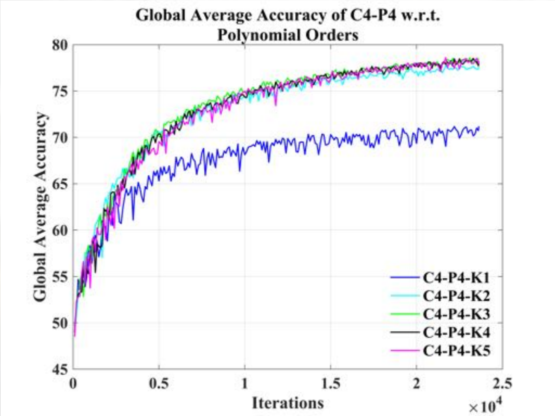
- λ (1×10^{-6}) is the coefficient of L2 norm.

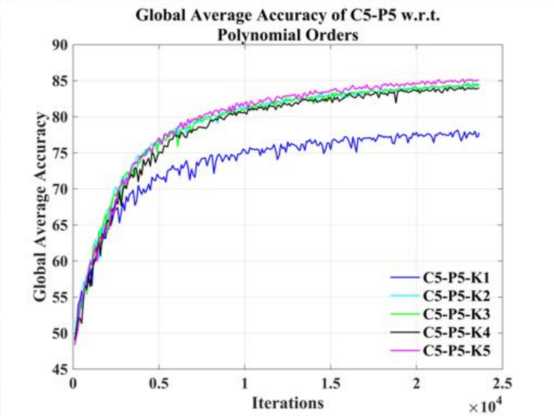
Which Chebyshev polynomial order should we use?

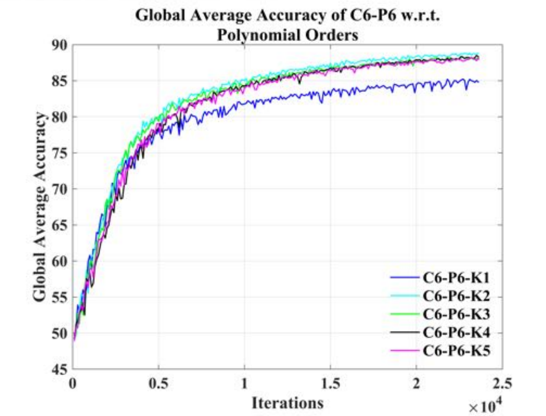






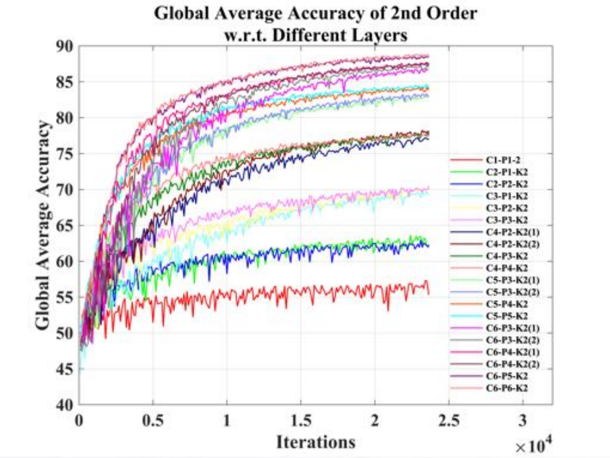


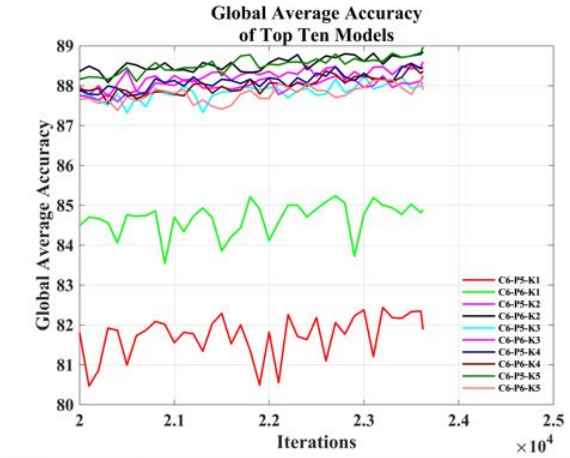


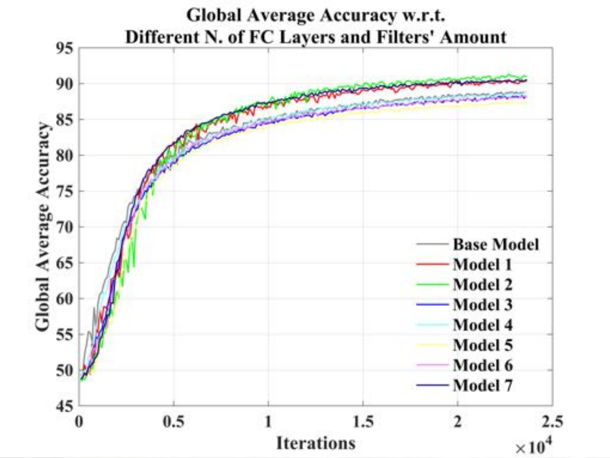


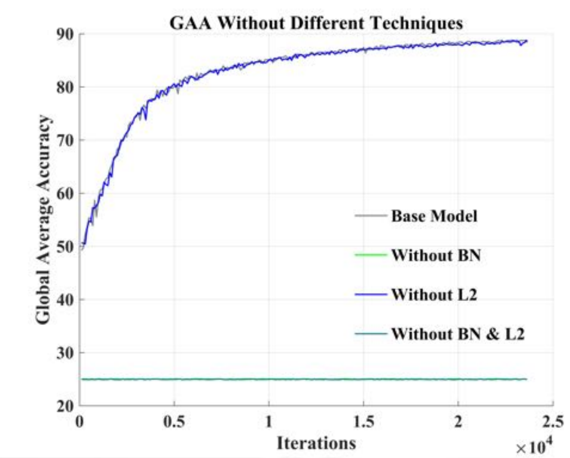
- The model using 1st order Chebyshev polynomial approximation performed worst (<58% accuracy), while the others using 2nd to 5th order performed nearly the same.
- **So, for our GCNs-Net, we will use 2nd Order Chebyshev to approximate filters.**

Which model should we use for EEG signals classification?



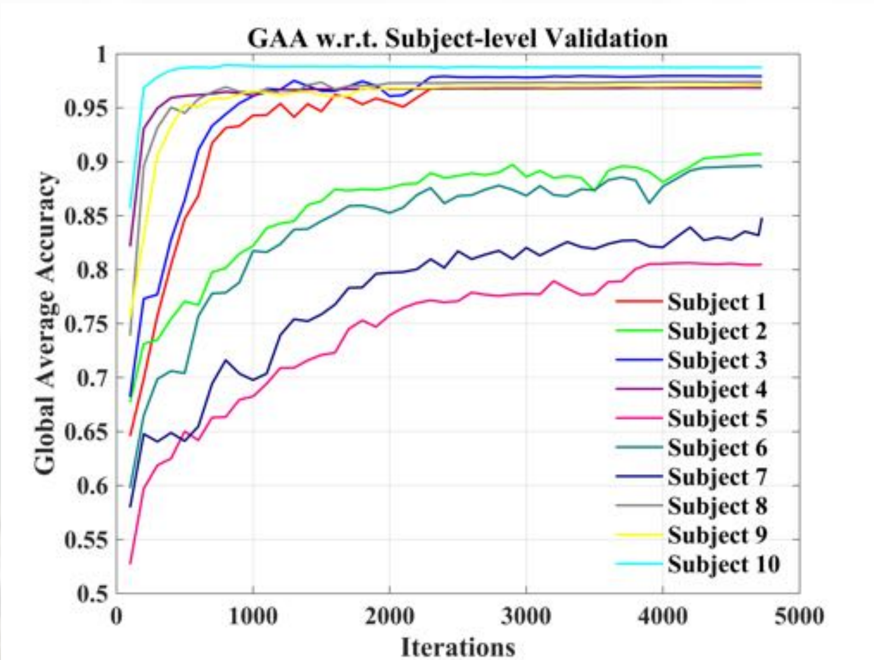


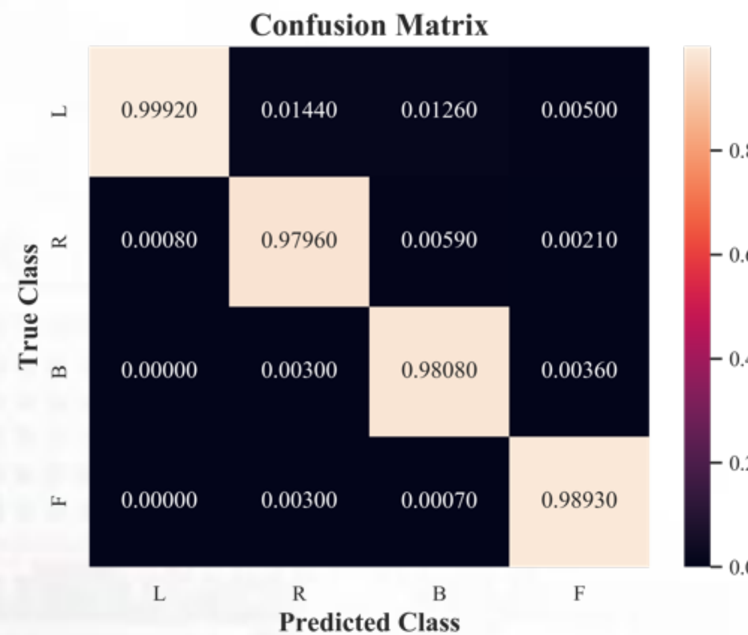




- For 64-electrode EEG system, the C6-P6-K2 model performed best (**88.85%** accuracy).
- **Six-layer graph convolutions, each followed by a graph pooling layer, and finally used a Softmax layer to predict the EEG tasks.**
- Used **Batch normalization (BN)**, and **L2 regularization** to prevent overfitting.

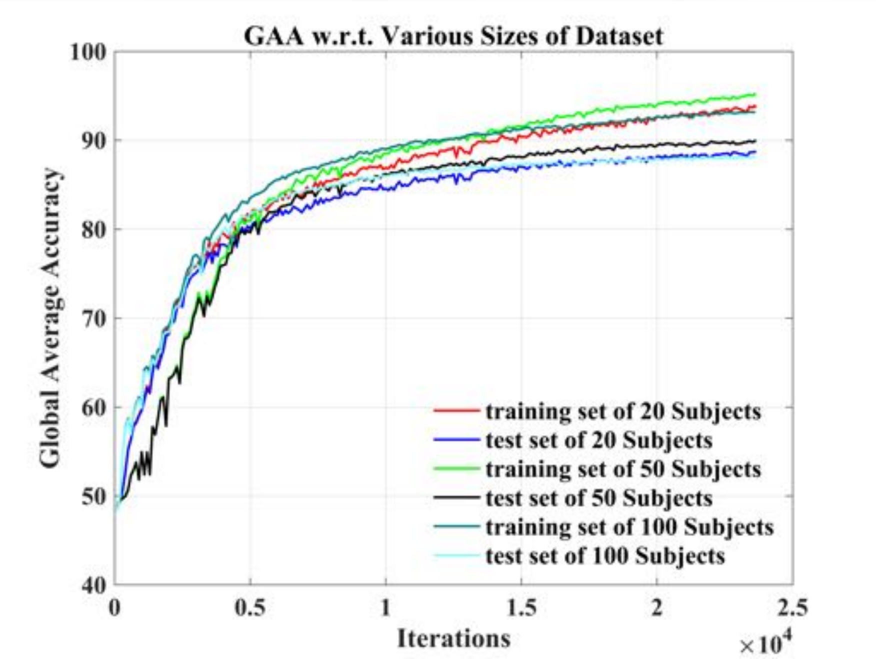
Results of the Subject-level Prediction:

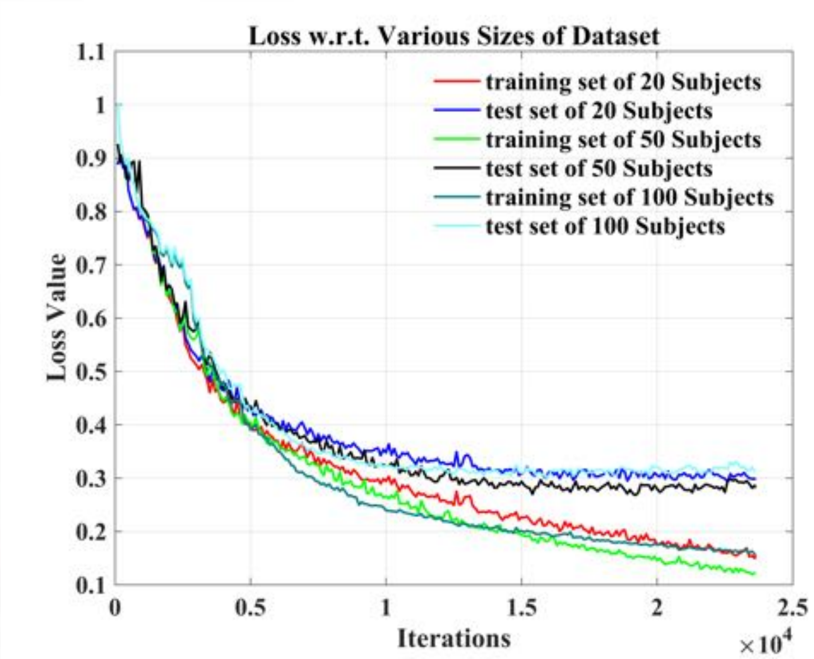




- For the Subject-level prediction, we used the first 10 Subjects ($S_1 \sim S_{10}$) from the PhysioNet Dataset.
- Averaged accuracy: **93.056%**, Maximum accuracy: **98.72%**.

Results of the Group-level Prediction:





- At the Group-level, we used the first 20, 50, 100 Subjects from the PhysioNet Dataset.
- For 20 subjects, averaged accuracy: **88.57%**, maximum accuracy: **89.387%**.
- For 50 subjects, accuracy: **89.75%**.
- For 100 subjects, accuracy: **88.14%**.

Compared with State-of-the-art Models:

TABLE IV: Performance comparison on the PhysioNet Dataset

Related Work	Max. GAA	Avg. GAA	*p*-value	Level	Approach	Num of Subjects
Dose *et al.* (2018) [20]	-	58.58%	–	Group	CNNs	105
	80.38%	68.51%	< 0.05	Subject		1
Ma *et al.* (2018) [53]	82.65%	68.20%	–	Group	RNNs	12
Hou *et al.* (2020) [18]	94.50%	–	–	Group	ESI-CNNs	10
	96.00%	–	> 0.05	Subject		1
Author	**89.387%**	**88.57%**	–	**Group**	**GCNs-Net**	**20**
	88.14%	-				**100**
	98.72%	**93.056%**		**Subject**		**1**

TABLE V: Performance comparison on the High Gamma Dataset

Related Work	Avg. GAA	*p*-value	Level	Approach	Dataset
Schirrmeister *et al.* (2017) [22]	92.50%	< 0.05		CNNs	
Li *et al.* (2019) [54]	93.70%	< 0.05	Subject	CP-MixedNet	1 subjects
Tang *et al.* (2019) [27]	95.30%	> 0.05		DAN	
Author	**80.89%**	–	**Group**	**GCNs-Net**	**14 subject**
	96.24%		**Subject**		**1 subject**

To summarize my undergraduate studies

- **Research Topics:** EEG Signals/Tasks Classification
  - 5 Papers (All selected by SCI, 1 accepted, 4 under review)
  - Open-source EEG-DL on GitHub, a Deep Learning (DL) Library written by TensorFlow for EEG signals classification, currently supports 22 DL algorithms, and keeps updating.
  - 36+ GitHub stars, 12+ forks
- 2017 Summer School at the *University of California, Irvine*, CA, USA
- 2019 Summer Intern at *Tsinghua University*, Beijing, China
- Student Member of IEEE, ACM and CCF, and attended a few CCF talks in Beijing, China

*** The projects' details can be found at my Homepage.**

Publications

1. A Novel Approach of Decoding EEG Four-Class Motor Imagery Tasks via Scout ESI and CNN. Yimin Hou, Lu Zhou, **Shuyue Jia**, and Xiangmin Lun. *Journal of Neural Engineering*, 2019; 17(1):016048. (Published)
2. GCNs-Net: A Graph Convolutional Neural Network Approach for Decoding Time-resolved EEG Motor Imagery Signals. Xiangmin Lun, **Shuyue Jia ***, Yimin Hou, Yan Shi, Yang Li, Hanrui Yang, Shu Zhang, and Jinglei Lv. *IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE)*, 2020. (Major Revision)
3. Deep Feature Mining via Attention-based BiLSTM-GCN for Human Motor Imagery Recognition. Yimin Hou, **Shuyue Jia ***, Shu Zhang, Xiangmin Lun, Yan Shi, Yang Li, Hanrui Yang, Rui Zeng, and Jinglei Lv. *IEEE Access*, 2020. (Minor Revision/Resubmit)
4. A Novel Synergetic Framework for Enhancing Electronic Nose Performance to measure the Quality Difference of Rice. Yan Shi, Xiaofei Jia, Hangcheng Yuan, **Shuyue Jia**, Jingjing Liu, and Hong Men. *Measurement Science and Technology*, 2020. (Major Revision)
5. Improving Performance: a Collaborative Strategy for Multi-data Fusion of Electronic Nose and Hyperspectral to Track the Quality Difference of Rice. Yan Shi, Hangcheng Yuan, Chenao Xiong, **Shuyue Jia**, Jingjing Liu, and Hong Men. *Sensors & Actuators: B. Chemical*, 2020. (Under Review)

*** denotes the Corresponding Author.**

Acknowledgements

***Supervisors*: Yimin Hou, Hanrui Yang, Yang Li, Yan Shi, and Jinglei Lv**

***Co-authors*: Xiangmin Lun, Shu Zhang, and Rui Zeng**

***Labmates*: Ziyu Huo, and Lu Zhou**

***Friends*: Shitu Zhang, Shichang Li, Xingyu Tong**

My parents

I would like to thank all the people who help and support me during my undergraduate studies. These research works could not have happened without you.