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.
- **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

(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

(iii) Graph Representation

- Graph Weights & Degrees

(iv) The GCNs-Net

- Softmax
- Flatten
- Pooling
- GCN

Real-time 64-channel Raw EEG Signals

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

- **Correlations representation**: Pearson Matrix
  - Measure the linear correlations between nodes
  - Below, $\mu$ is the expectation, $\sigma$ 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**:
  $$L = I_N - D^{-\frac{1}{2}} A D^2$$
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

\[ g_{\theta}(\Lambda) = \text{diag}(\theta) \]

- \( \theta \) is a vector of Fourier coefficients.
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
    \[
    y = U \left( \sum_{k=1}^{K} \theta_k \Lambda^k \right) U^T x = \left( \sum_{k=1}^{K} \theta_k U \Lambda^k U^T \right) x = \left( \sum_{k=1}^{K} \theta_k L^k \right) x = \sum_{k=1}^{K} \theta_k (L^k x)
    \]
    \( x_{\text{new}} \leftarrow Lx_i = \sum_j A_{ij}(x_i - x_j) \)
  - \( \theta \) are the trainable parameters \( \leftarrow \) 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"

\[
K = 1
\]
\[
K = 2
\]
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, \( \hat{y} \) is the final output probability
  \[ \hat{y} = \text{argmax} \left( \frac{e^{y_i}}{\sum_{i=1}^{4} e^{y_i}} \right) \]

- **Loss Function: Cross-entropy with L2 regularization**
  \[ \text{Loss} = -\sum_{i=1}^{4} y_i \log(\hat{y}_i) + \lambda \left( \sum_{j=1}^{n} w_j^2 + b_j^2 \right) \]

- \( \lambda (1 \times 10^{-6}) \) is the coefficient of L2 norm.
Which Chebyshev polynomial order should we use?

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

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

**TABLE V: Performance comparison on the High Gamma Dataset**

<table>
<thead>
<tr>
<th>Related Work</th>
<th>Avg. GAA</th>
<th>p-value</th>
<th>Level</th>
<th>Approach</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schirrmeister et al. (2017) [22]</td>
<td>92.50%</td>
<td>&lt; 0.05</td>
<td>Subject</td>
<td>CNNs</td>
<td>1 subjects</td>
</tr>
<tr>
<td>Li et al. (2019) [54]</td>
<td>93.70%</td>
<td>&lt; 0.05</td>
<td>Subject</td>
<td>CP-MixedNet</td>
<td>1 subjects</td>
</tr>
<tr>
<td>Tang et al. (2019) [27]</td>
<td>95.30%</td>
<td>&gt; 0.05</td>
<td>Subject</td>
<td>DAN</td>
<td>1 subject</td>
</tr>
<tr>
<td>Author</td>
<td>80.89%</td>
<td></td>
<td>Group</td>
<td>GCNs-Net</td>
<td>14 subject</td>
</tr>
<tr>
<td>Author</td>
<td>96.24%</td>
<td></td>
<td>Subject</td>
<td>GCNs-Net</td>
<td>1 subject</td>
</tr>
</tbody>
</table>
To summarize my undergraduate studies

• **Research Topics:** EEG Signals/Tasks Classification
  – 5 Papers (All selected by SCI, 1 accepted, 4 under review)
  – 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


* 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.