

# A Brief Summary of Three Selected Projects

## Tasks

- ✓ Regression (IQA) and Classification (EEG)

## IQA Research Novelty

- ✓ [Definition] Non-local Modeling and Local Modeling
- ✓ [Definition] Global Distortions and Local Distortions
- ✓ [Motivation] Human Visual System (HVS) perceives Image Quality:  
Adaptive to local content + Long-range Dependency constructed among different regions
- ✓ [Method] Superpixel-based Graph Neural Network to explore Non-local Interactions

## EEG Research Novelty

- ✓ [Motivation] Graph Modeling for EEG Electrodes System
- ✓ [Method] Graph Representation Learning of EEG Signals
- ✓ [Motivation] Spatial-Temporal Analysis of EEG Signals
- ✓ [Method] Deep Feature Mining of EEG Signals



## No-reference Image Quality Assessment via Non-local Modeling



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



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

# Selected Research Projects

Shuyue Jia

January 10th, 2023

<https://github.com/SuperBruceJia>

# No-reference Image Quality Assessment via Non-local Modeling

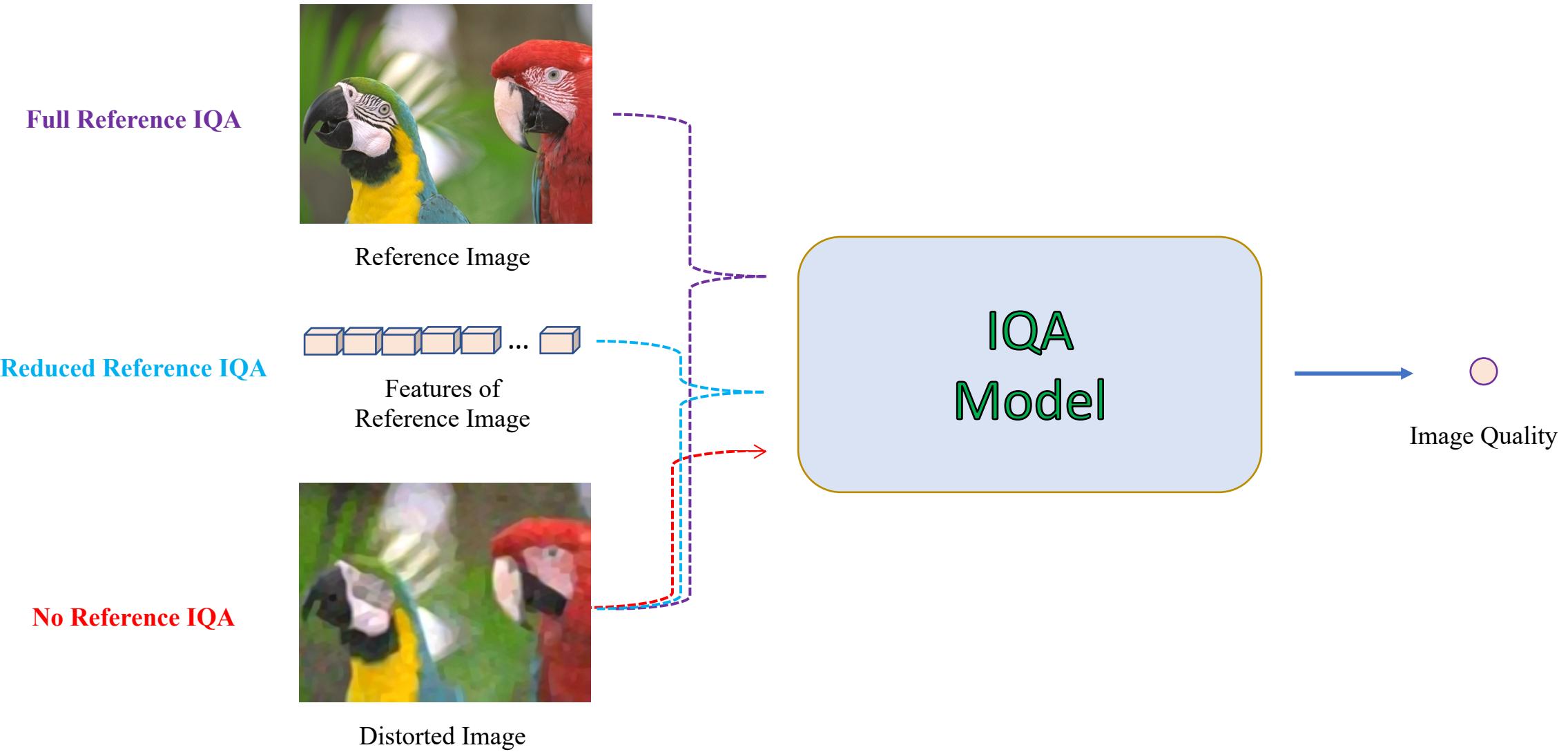
Shuyue Jia<sup>1</sup>, Baoliang Chen<sup>1</sup>, Dingquan Li<sup>2</sup>, and Shiqi Wang<sup>1\*</sup>

<sup>1</sup> Department of Computer Science, City University of Hong Kong

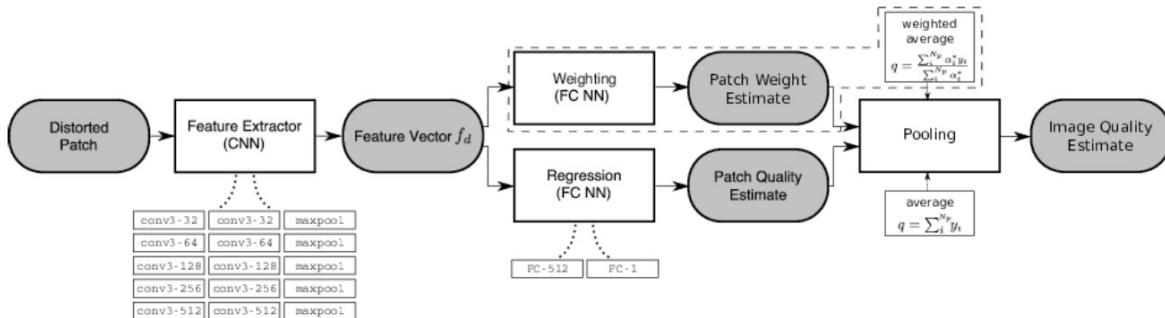
<sup>2</sup> Peng Cheng Laboratory

Project: <https://github.com/SuperBruceJia/NLNet-IQA>

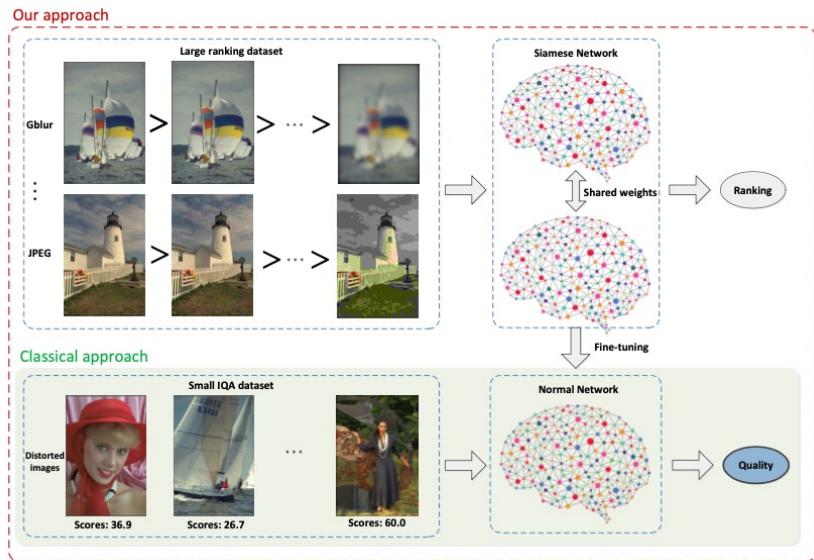
# Image Quality Assessment (IQA)



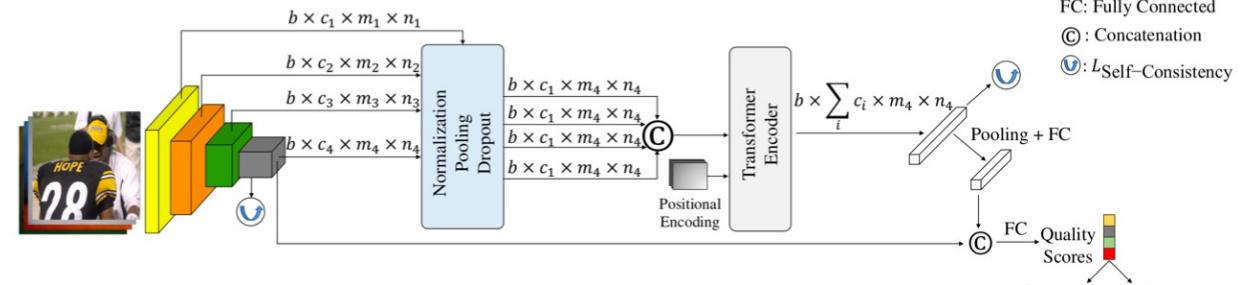
# Recent Progress on No-reference IQA



CNN-based Methods [1]



Ranking-based Methods [2]



Transformer-based Methods [3]

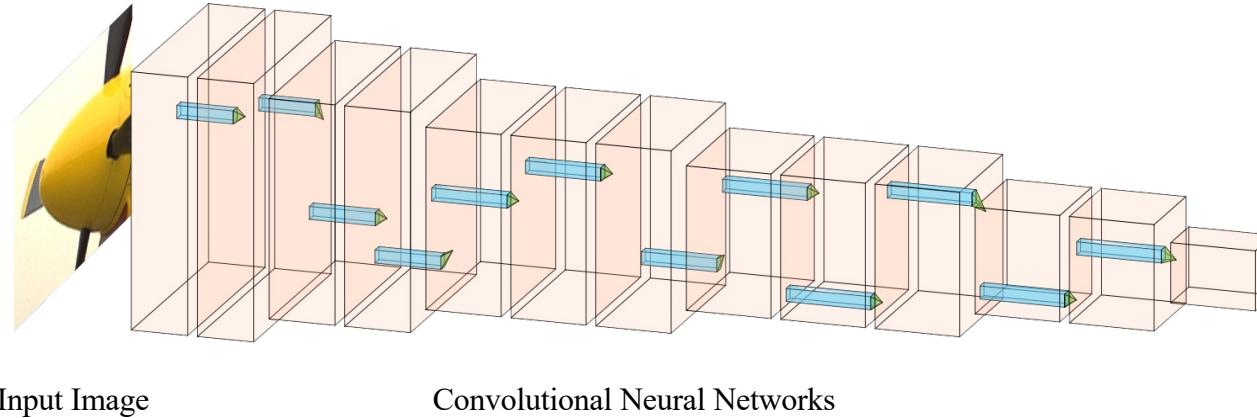
Credit:

[1] Bosse *et al.*, Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment, In TIP 2018

[2] Liu *et al.*, RankIQA: Learning from Rankings for No-reference Image Quality Assessment, In ICCV 2017

[3] Golestaneh *et al.*, No-Reference Image Quality Assessment via Transformers, Relative Ranking, and Self-Consistency, In WACV 2022

# Challenges



- **Local Modeling** (Convolutional Neural Networks):
  - ✓ Translation Invariance (Pooling)
  - ✓ Translation Equivalence (Convolution)
  - ✓ Sharable Trainable Parameters (Weight Sharing)
- **Limitations:**
  - ✓ Small-sized receptive field → **Extracted features are too local**
  - ✓ Parameters fixed across the whole image → **Image content is equally treated**
  - ✓ Lack of geometric and relational modeling → **Missing complex relations and dependencies**

# Motivation



**Local Feature Extraction** is critical



**Non-local Dependency**  
learned by the NLNet

- ✓ HVS is **adaptive to the local content**
  - ***Local feature extraction*** via a pre-trained CNN
- ✓ HVS perceives image quality with **long-range dependency constructed among different regions**
  - ***Non-local feature extraction*** for long-range dependency and relational modeling

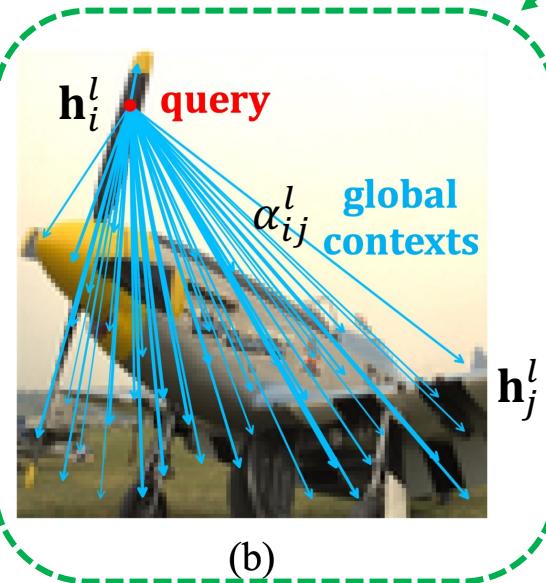
# Definition



**Convolution:  
Pixel-to-Pixel  
Modeling**

Local feature extraction is critical

Figure 2: Local region feature extraction and non-local dependency feature extraction



**Non-local:  
Object-to-Pixel  
Modeling**

*Simplicity is Beauty*

**Spatial Integration of Information**

$$\mathbf{h}_i^l = \text{ELU} \left( \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^l \mathbf{W}^l \mathbf{h}_j^l \right)$$

**Spatial Weighting Functions**

$$\alpha_{ij}^l = \frac{\exp(a_{ij}^l)}{\sum_{k \in \mathcal{N}(i)} a_{ik}^l}$$

$$a_{ij}^l = \text{LeakyReLU}(\text{FC}([\mathbf{W}^l \mathbf{h}_i^l \parallel \mathbf{W}^l \mathbf{h}_j^l]))$$

- ✓ **Local Modeling:** encodes spatially proximate **local neighborhoods**.
- ✓ **Non-local Modeling:** establishes **spatial integration of information** by long- and short-range communications with different **spatial weighting functions**.

# Non-local Behavior

Object-to-Pixel  
Region Feature Extraction

Non-local  
Dependency & Relational  
Modeling



Semantics and Content  
Understanding

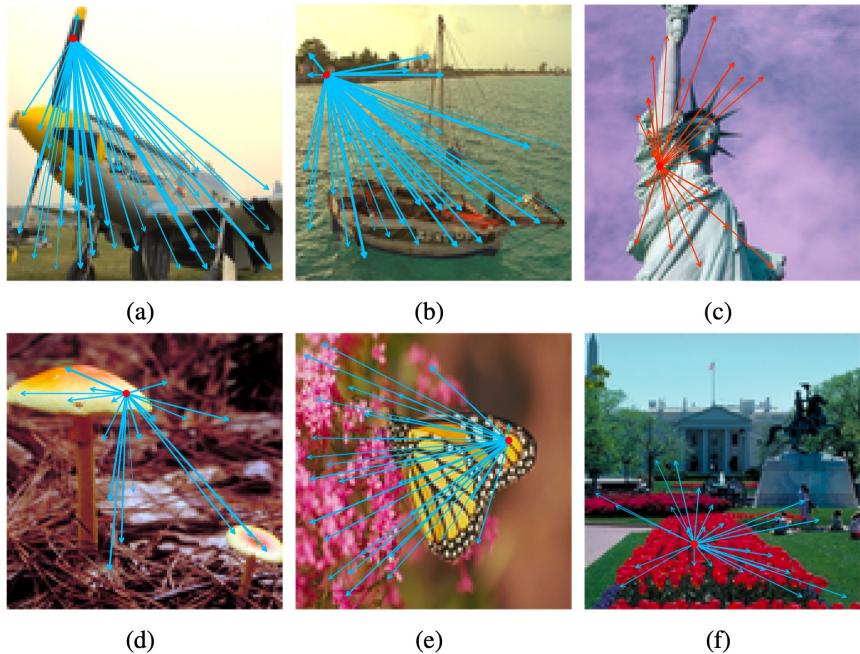


Figure 3.1: The non-local behavior of the long-range dependency and relational modeling. (a) The plane image with a query on wings. (b) The boat image with a query on nearby river bank. (c) The Statue of Liberty image with a query on the lady. (d) The shrooms image with a query on one shroom. (e) The butterfly image with a query on the wing. (f) The Lafayette Square, Washington, D.C. image with a query on flowers.

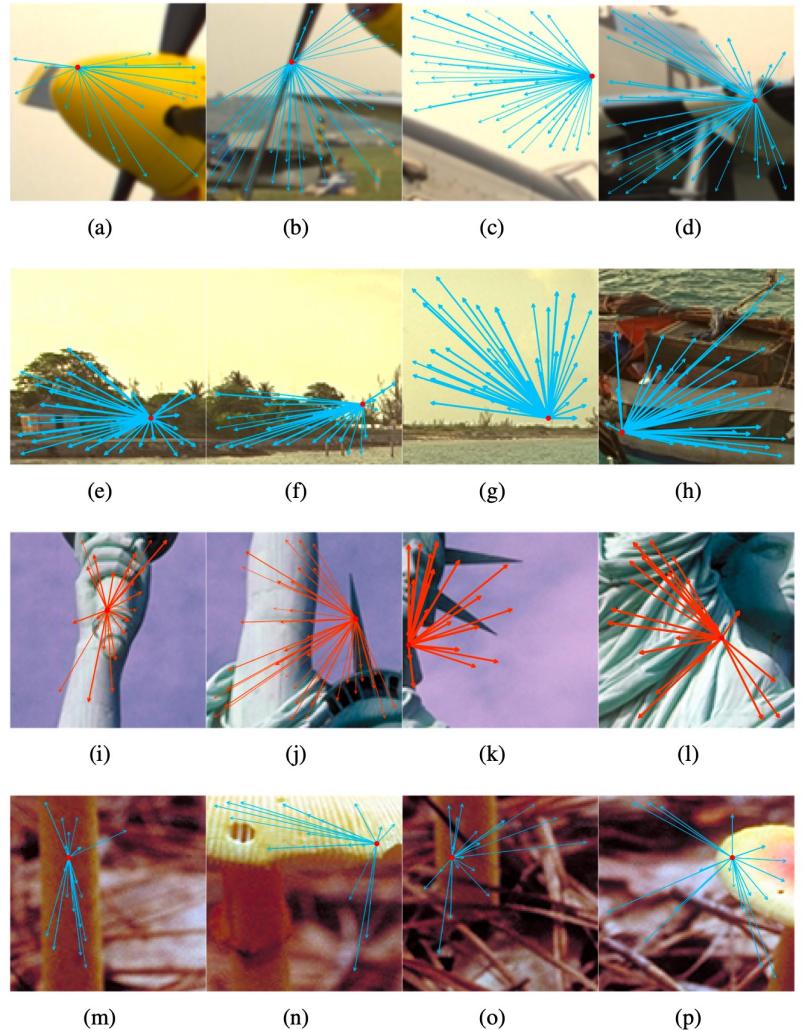


Figure 3.2: Selected demonstrations of the non-local behavior and long-range dependencies with regard to the cropped image patches from the illustrated images.  
The details of Figure (a) to (p) are described in the thesis.

- ✓ **Non-local Modeling:** establishes the **spatial integration of information**

by **long- and short-range communications** with different **spatial weighting functions**.

# Definition

## Non-local Recurrence

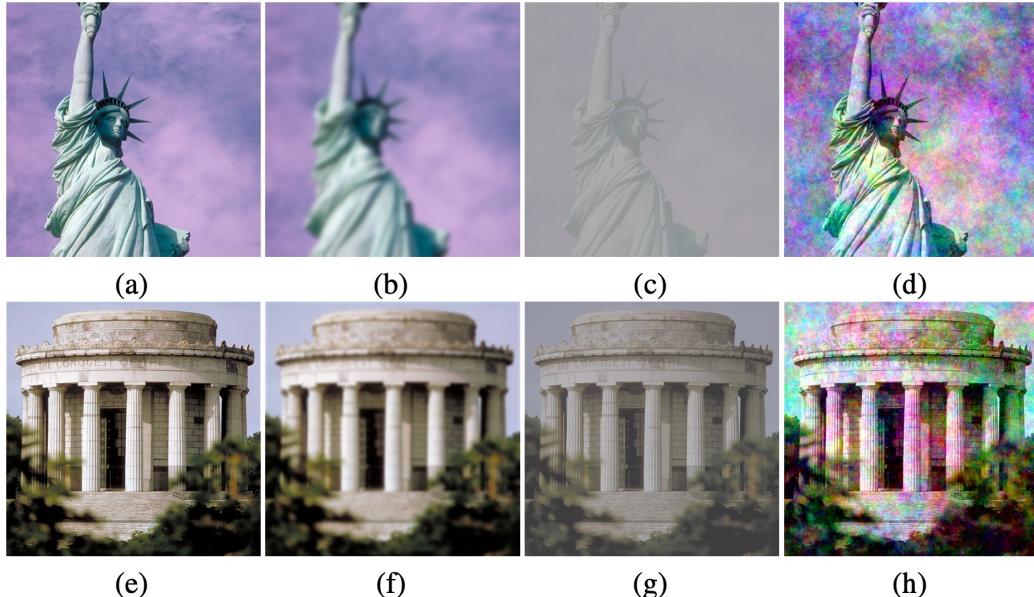


Figure 4.9: Demonstrations of the global distortions (b/f: GB, c/g: CC, d/h: PN) contaminating the Statue of Liberty and George Rogers Clark Memorial images.

Figure (a) and Figure (e) are reference images from the CSIQ database.

## Global Distortion

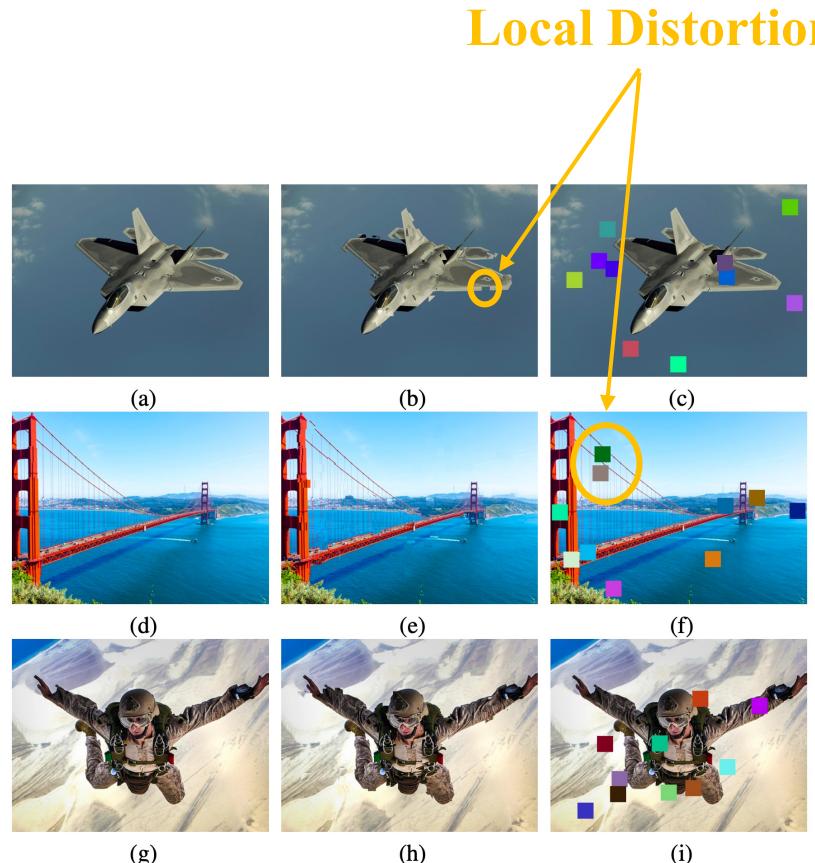


Figure 4.11: Demonstrations of the local distortions (b/e/h: non-eccentricity patch and c/f/i: color block). Figure (a), Figure (d), and Figure (g) are reference images from the KADID-10k database.

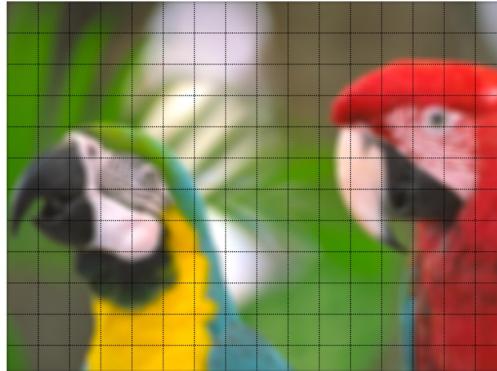
## Local Distortion

- ✓ **Global Distortion:** **globally and uniformly distributed** distortions with **non-local recurrences** over the image.
- ✓ **Local Distortion:** **local nonuniform-distributed** distortions **in a local region**.

# Superpixel Segmentation



(a)



(b)



(c)



(d)

## Superpixel vs. Square Patch

- ✓ Adherence to boundaries and **Visually meaningful**
- ✓ **Accurate feature extraction**

Figure 4.2: The superpixel vs. square patch representation (with size of  $\approx 32 \times 32$ ) of the plane image from the TID2013 database.

# Superpixel Segmentation

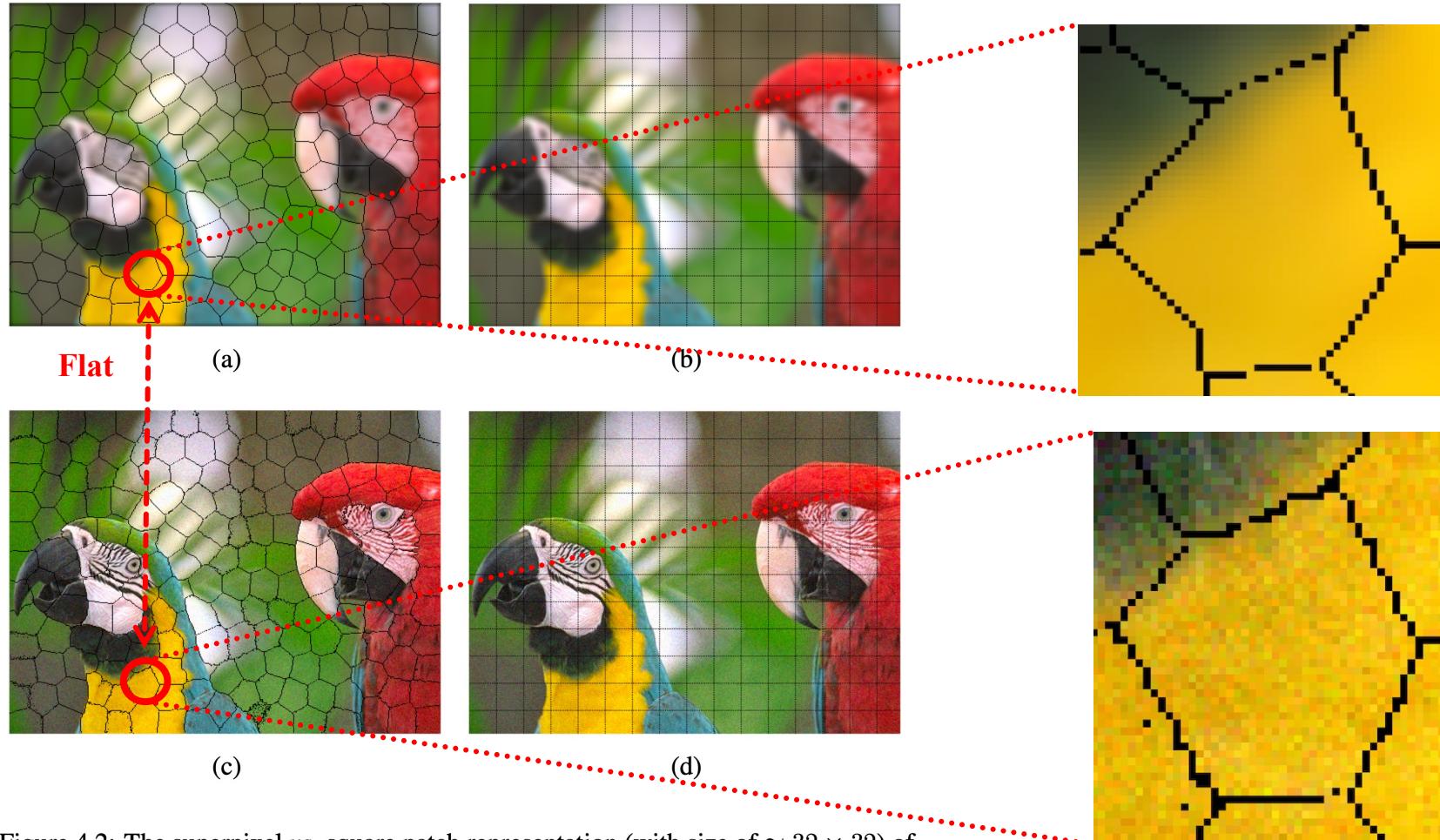


Figure 4.2: The superpixel vs. square patch representation (with size of  $\approx 32 \times 32$ ) of the plane image from the TID2013 database.



Reference

Gaussian Blur

Gaussian Noise

# Superpixel Segmentation

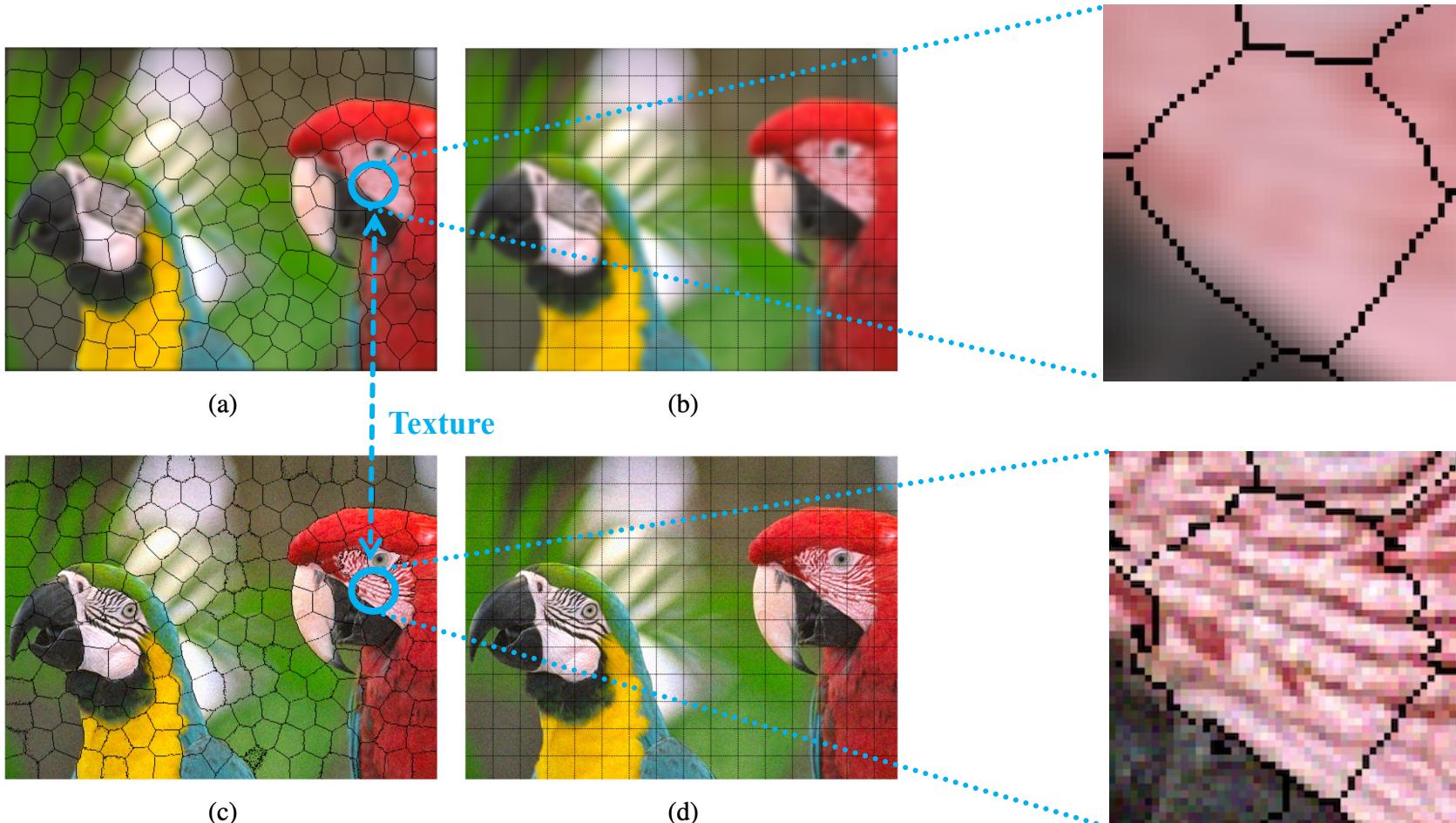


Figure 4.2: The superpixel vs. square patch representation (with size of  $\approx 32 \times 32$ ) of the plane image from the TID2013 database.



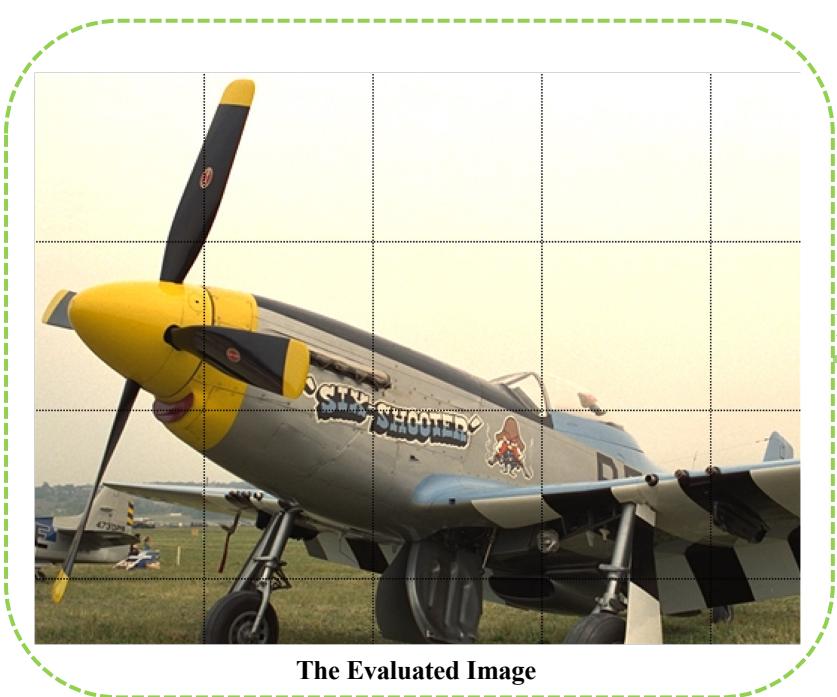
Reference

Gaussian Blur

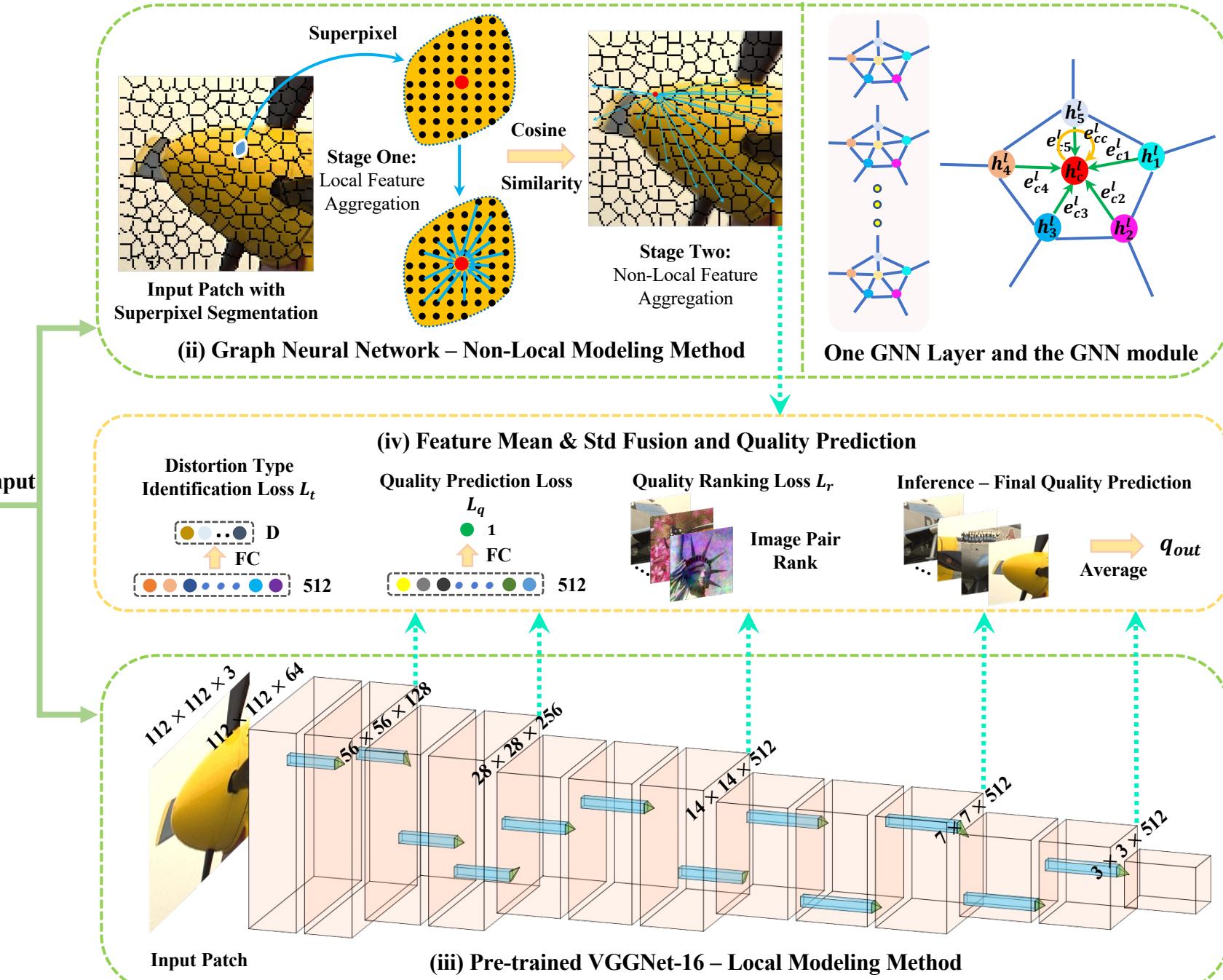


Gaussian Noise

# NLNet Architecture



(i) Image Preprocessing



# Experimental Setup

- **Databases:**
  - LIVE, CSIQ, TID2013, and KADID-10k
- **Evaluation Metrics:**
  - SRCC (Spearman Rank-order Correlation Coefficient)
  - PLCC (Pearson Linear Correlation Coefficient)
- **Experimental Setting:**
  - Intra-database Experiments:
    - 60% training, 20% validation, and 20% testing, with seeds from 1 to 10
  - Cross-database Experiments:
    - One database as the training set, and the other databases as the testing set
    - Report the last epoch's performance

Screen Content

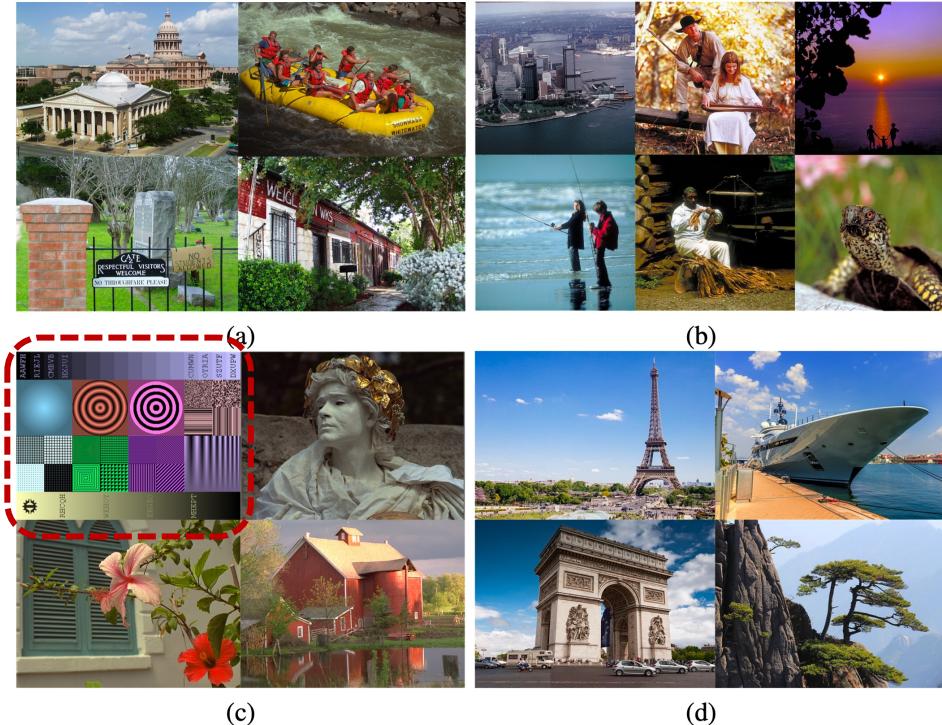


Figure 1.1: Natural images and a screen content image from the constructed databases.

(a) LIVE Database [13] (b) CSIQ Database [14] (c) TID2013 Database [15] (d)  
KADID-10k Database [16].

Table 4.1: Brief summary of the LIVE, CSIQ, TID2013, and KADID-10k databases.

Database	LIVE [13]	CSIQ [14]	TID2013 [15]	KADID-10k [16]
Num. of Reference Images	29	30	25	81
Num. of Distorted Images	779	866	3,000	10,125
Num. of Distortion Types	5	6	24	25
Num. of Distortion Levels	5 ~ 8	3 ~ 5	5	5
Annotation	DMOS	DMOS	MOS	MOS
Range	[0, 100]	[0, 1]	[0, 9]	[1, 5]

Natural Images

# Intra-Database Experiments

Table 4.2: Performance comparisons on the LIVE, CSIQ, and TID2013 databases.  
Top two results are highlighted in bold.

Method	LIVE		CSIQ		TID2013	
	SRCC	PLCC	SRCC	PLCC	SRCC	PLCC
BRISQUE (2012) [10]	0.939	0.935	0.746	0.829	0.604	0.694
CORNIA (2012) [104]	0.947	0.950	0.678	0.776	0.678	0.768
M3 (2015) [105]	0.951	0.950	0.795	0.839	0.689	0.771
HOSA (2016) [103]	0.946	0.947	0.741	0.823	0.735	0.815
FRIQUEE (2017) [90]	0.940	0.944	0.835	0.874	0.68	0.753
DIQaM-NR (2018) [35]	0.960	<b>0.972</b>	-	-	0.835	0.855
DB-CNN (2020) [64]	0.968	<b>0.971</b>	<b>0.946</b>	<b>0.959</b>	0.816	0.865
HyperIQA (2020) [65]	0.962	0.966	0.923	0.942	0.729	0.775
GraphIQA (2022) [86]	<b>0.968</b>	0.970	0.920	0.938	-	-
TReS (2022) [87]	<b>0.969</b>	0.968	0.922	0.942	<b>0.863</b>	<b>0.883</b>
NLNet	0.962	0.963	<b>0.941</b>	<b>0.958</b>	<b>0.856</b>	<b>0.880</b>

SOTA  
Transformer

Fewer Training Data  
(- 20% Total Data)

Highly Competitive Performance

Table 4.3: Performance comparisons on the KADID-10k database.

Top two results are highlighted in bold.

Method	BRISQUE [10]	CORNIA [104]	HOSA [103]	InceptionResNetV2 [16]	DB-CNN [64]	HyperIQA [65]	TReS [87]	NLNet
SRCC	0.519	0.519	0.609	0.731	0.851	<b>0.852</b>	<b>0.859</b>	0.846
PLCC	0.554	0.554	0.653	0.734	<b>0.856</b>	0.845	<b>0.858</b>	0.850

# Cross-Database Evaluations

Table 4.9: Cross-database performance comparisons.

Training	LIVE		CSIQ		TID2013	
Testing	CSIQ	TID2013	LIVE	TID2013	LIVE	CSIQ
BRISQUE (2012) [10]	0.562	0.358	0.847	0.454	0.790	0.590
CORNIA (2012) [104]	0.649	0.360	0.853	0.312	0.846	0.672
M3 (2015) [105]	0.621	0.344	0.797	0.328	0.873	0.605
HOSA (2016) [103]	0.594	0.361	0.773	0.329	0.846	0.612
FRIQUEE (2017) [90]	0.722	0.461	0.879	0.463	0.755	0.635
DIQaM-NR (2018) [35]	0.681	0.392	-	-	-	0.717
DB-CNN (2020) [64]	<b>0.758</b>	<b>0.524</b>	0.877	<b>0.540</b>	<b>0.891</b>	<b>0.807</b>
HyperIQA (2020) [65]	0.697	<b>0.538</b>	<b>0.905</b>	<b>0.554</b>	0.839	0.543
<b>NLNet</b>	<b>0.771</b>	0.497	<b>0.923</b>	0.516	<b>0.895</b>	<b>0.730</b>

Similar  
Distortions

TID:  
More Distortion Types & Levels

# Single Distortion Type Evaluation

Table 4.4: The average SRCC and PLCC results of the individual distortion type on the LIVE database. Top two results are highlighted in bold.

SRCC	Global Distortion				Local Distortion FF
	JPEG	JP2K	WN	GB	
BRISQUE (2012) [10]	0.965	0.929	0.982	<b>0.964</b>	0.828
CORNIA (2012) [104]	0.947	0.924	0.958	0.951	0.921
M3 (2014) [105]	0.966	0.930	<b>0.986</b>	0.935	0.902
HOSA (2016) [103]	0.954	0.935	0.975	0.954	<b>0.954</b>
FRIQUEE (2017) [90]	0.947	0.919	0.983	0.937	0.884
dipIQ (2017) [82]	0.969	<b>0.956</b>	0.975	0.940	-
WaDIQaM (2018) [35]	0.953	0.942	0.982	0.938	0.923
DB-CNN (2020) [64]	<b>0.972</b>	0.955	0.980	0.935	0.930
HyperIQA (2020) [65]	0.961	0.949	0.982	0.926	0.934
<b>NLNet</b>	<b>0.979</b>	<b>0.958</b>	<b>0.990</b>	<b>0.964</b>	<b>0.941</b>
PLCC	Global Distortion				Local Distortion FF
	JPEG	JP2K	WN	GB	
BRISQUE (2012) [10]	0.971	0.940	0.989	<b>0.965</b>	0.894
CORNIA (2012) [104]	0.962	0.944	0.974	0.961	0.943
M3 (2014) [105]	0.977	0.945	<b>0.992</b>	0.947	0.920
HOSA (2016) [103]	0.967	0.949	0.983	<b>0.967</b>	<b>0.967</b>
FRIQUEE (2017) [90]	0.955	0.935	0.991	0.949	0.936
dipIQ (2017) [82]	0.980	<b>0.964</b>	0.983	0.948	-
DB-CNN (2020) [64]	<b>0.986</b>	<b>0.967</b>	0.988	0.956	<b>0.961</b>
<b>NLNet</b>	<b>0.986</b>	0.961	<b>0.993</b>	0.964	0.951

Noisy  
and  
Compressed  
Images

Global  
Distortion

Non-local  
Recurrence

Local  
Distortion

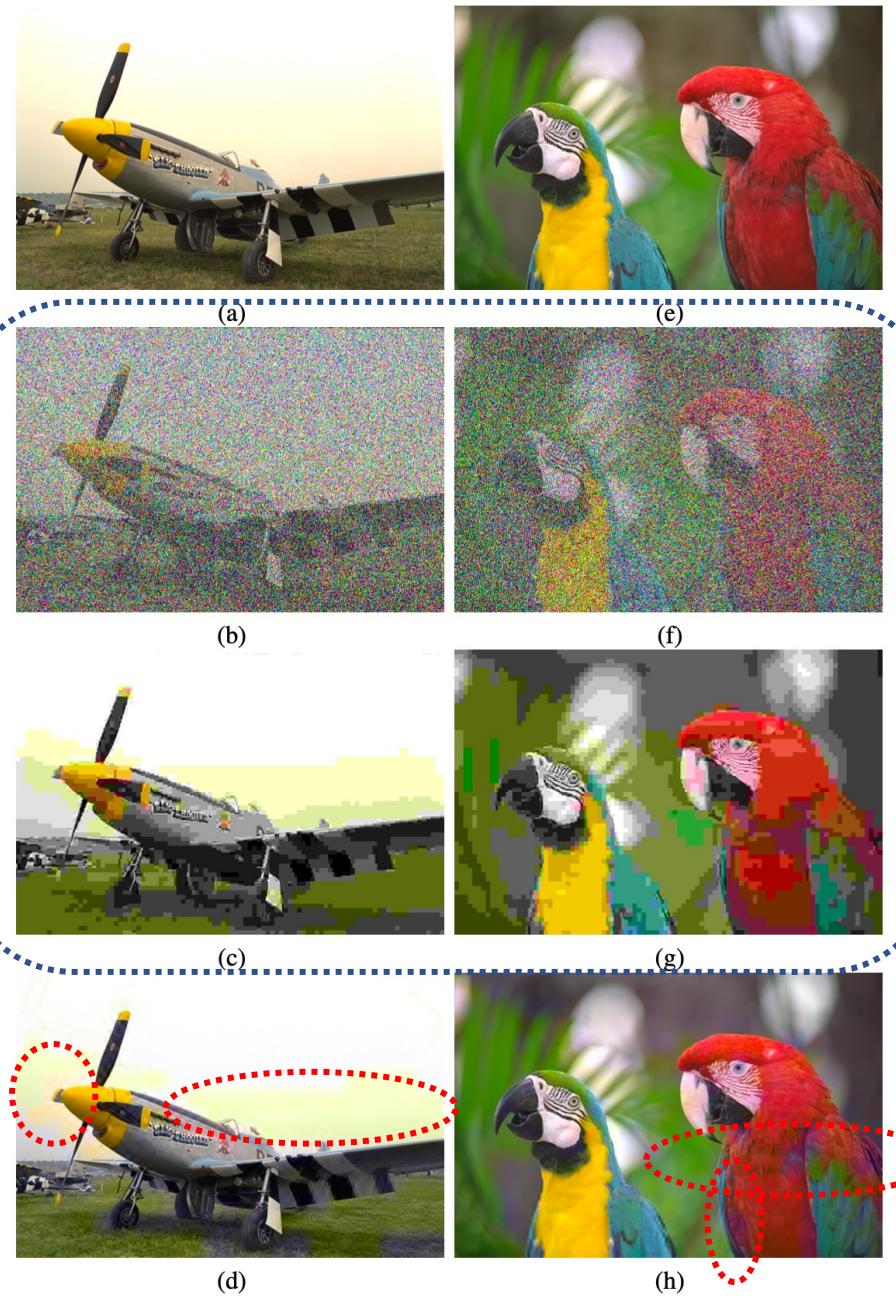


Figure 4.7: Demonstrations of the global distortions (b/f: WN and c/g: JPEG) and local distortions (d/h: FF) contaminating the plane and parrot images. Figure (a) and Figure (e) are reference images from the LIVE database.

# Single Distortion Type Evaluation

## Noise-Related Distortions

Table 4.5: The average SRCC and PLCC results of the individual distortion type on the CSIQ database. Top two results are highlighted in bold.

SRCC	JPEG	JP2K	WN	GB	PN	CC
BRISQUE (2012) [10]	0.806	0.840	0.723	0.820	0.378	0.804
CORNIA (2012) [104]	0.513	0.831	0.664	0.836	0.493	0.462
M3 (2014) [105]	0.740	0.911	0.741	0.868	0.663	0.770
HOSA (2016) [103]	0.733	0.818	0.604	0.841	0.500	0.716
FRIQUEE (2017) [90]	0.869	0.846	0.748	0.870	0.753	0.838
dipIQ (2017) [82]	0.936	0.944	0.904	0.932	-	-
MEON (2018) [71]	<b>0.948</b>	0.898	<b>0.951</b>	0.918	-	-
WaDIQaM (2018) [35]	0.853	0.947	0.974	0.979	0.882	<b>0.923</b>
DB-CNN (2020) [64]	0.940	0.953	0.948	<b>0.947</b>	<b>0.940</b>	0.870
HyperIQA (2020) [65]	0.934	<b>0.960</b>	0.927	0.915	0.931	0.874
<b>NLNet</b>	<b>0.972</b>	<b>0.963</b>	<b>0.965</b>	<b>0.955</b>	<b>0.969</b>	<b>0.968</b>
PLCC	JPEG	JP2K	WN	GB	PN	CC
BRISQUE (2012) [10]	0.828	0.887	0.742	0.891	0.496	0.835
CORNIA (2012) [104]	0.563	0.883	0.687	0.904	0.632	0.543
M3 (2014) [105]	0.768	0.928	0.728	0.917	0.717	0.787
HOSA (2016) [103]	0.759	0.899	0.656	0.912	0.601	0.744
FRIQUEE (2017) [90]	0.885	0.883	0.778	0.905	0.769	0.864
dipIQ (2017) [82]	0.975	<b>0.959</b>	0.927	0.958	-	-
MEON (2018) [71]	0.979	0.925	<b>0.958</b>	0.946	-	-
DB-CNN (2020) [64]	<b>0.982</b>	0.971	0.956	<b>0.969</b>	<b>0.950</b>	<b>0.895</b>
<b>NLNet</b>	<b>0.991</b>	<b>0.976</b>	<b>0.967</b>	<b>0.9746</b>	<b>0.966</b>	<b>0.969</b>

Global  
Distortion



Figure 4.9: Demonstrations of the global distortions (b/f: GB, c/g: CC, d/h: PN) contaminating the Statue of Liberty and George Rogers Clark Memorial images.

Figure (a) and Figure (e) are reference images from the CSIQ database.

# Single Distortion Type Evaluation

Table 4.6: The average SRCC results of the individual distortion type on the TID2013 database. Top two results are highlighted in bold.

SRCC	Distortion Type	BRISQUE [10]	FRIQUEE [90]	HOSA [103]	MEON [71]	M3 [105]	DB-CNN [64]	CORNIA [104]	NLNet
Global Distortion	Additive Gaussian noise	0.711	0.730	<b>0.833</b> ↑ <b>8.4%</b>	0.813	0.766	0.790	0.692	<b>0.917</b>
	Lossy compression of noisy images	0.609	0.641	0.838	0.772	0.692	<b>0.860</b> ↑ <b>7.5%</b>	0.712	<b>0.935</b>
	Additive noise in color components	0.432	0.573	0.551	<b>0.722</b> ↑ <b>12.8%</b>	0.740	0.700	0.137	<b>0.850</b>
	Comfort noise	0.196	0.318	0.622	0.406	0.353	<b>0.752</b> ↑ <b>11.8%</b>	0.617	<b>0.870</b>
	Contrast change	-0.001	<b>0.585</b>	0.362	0.252	0.155	0.548	0.254	<b>0.793</b>
	Change of color saturation	0.003	0.589	0.045	<b>0.684</b>	-0.199	0.631	0.169	<b>0.827</b>
	Spatially correlated noise	0.746	0.866	0.842	<b>0.926</b> ↑ <b>3.2%</b>	0.82	0.826	0.741	<b>0.958</b>
	High frequency noise	0.842	0.847	0.897	<b>0.911</b> ↑ <b>1.0%</b>	0.000	0.879	0.815	<b>0.921</b>
	Impulse noise	0.765	0.730	0.809	<b>0.901</b> ↑ <b>1.2%</b>	0.738	0.708	0.616	<b>0.913</b>
	Quantization noise	0.662	0.764	0.815	<b>0.888</b> ↑ <b>4.1%</b>	0.832	0.825	0.661	<b>0.929</b>
	Gaussian blur	0.871	0.881	0.883	0.887	<b>0.896</b>	0.859	0.850	<b>0.912</b>
	Image denoising	0.612	0.839	0.854	0.797	0.709	<b>0.865</b> ↑ <b>1.7%</b>	0.764	<b>0.882</b>
	JPEG compression	0.764	0.813	0.891	0.850	0.844	<b>0.894</b> ↑ <b>1.1%</b>	0.797	<b>0.905</b>
	JPEG 2000 compression	0.745	0.831	<b>0.919</b> ↑ <b>1.1%</b>	0.891	0.885	0.916	0.846	<b>0.930</b>
	Multiplicative Gaussian noise	0.717	0.704	0.768	<b>0.849</b> ↑ <b>5.5%</b>	0.738	0.711	0.593	<b>0.904</b>
	Image color quantization with dither	0.831	0.768	0.896	0.857	<b>0.908</b>	0.833	0.683	<b>0.911</b>
	Sparse sampling and reconstruction	0.807	0.891	<b>0.909</b>	0.855	0.893	0.902	0.865	<b>0.940</b>
	Chromatic aberrations	0.615	0.737	0.753	<b>0.779</b>	0.570	0.732	0.696	<b>0.773</b>
	Masked noise	0.252	0.345	0.468	<b>0.728</b>	0.577	0.646	0.451	<b>0.700</b>
	Mean shift (intensity shift)	0.219	<b>0.254</b>	0.211	0.177	0.119	-0.009	0.232	<b>0.358</b>
Local Distortion	JPEG transmission errors	0.301	0.498	0.730	0.746	0.375	<b>0.772</b> ↑ <b>3.3%</b>	0.694	<b>0.805</b>
	JPEG 2000 transmission errors	0.748	0.660	0.710	0.716	0.718	<b>0.773</b> ↑ <b>10.2%</b>	0.686	<b>0.875</b>
	Non eccentricity pattern noise	0.269	0.076	0.242	0.116	0.173	<b>0.270</b> ↑ <b>34.6%</b>	0.200	<b>0.616</b>
	Local block-wise distortions with different intensity	0.207	0.032	0.268	<b>0.500</b>	0.379	0.444	0.027	<b>0.493</b>

## Noise and Compression-Related Distortions

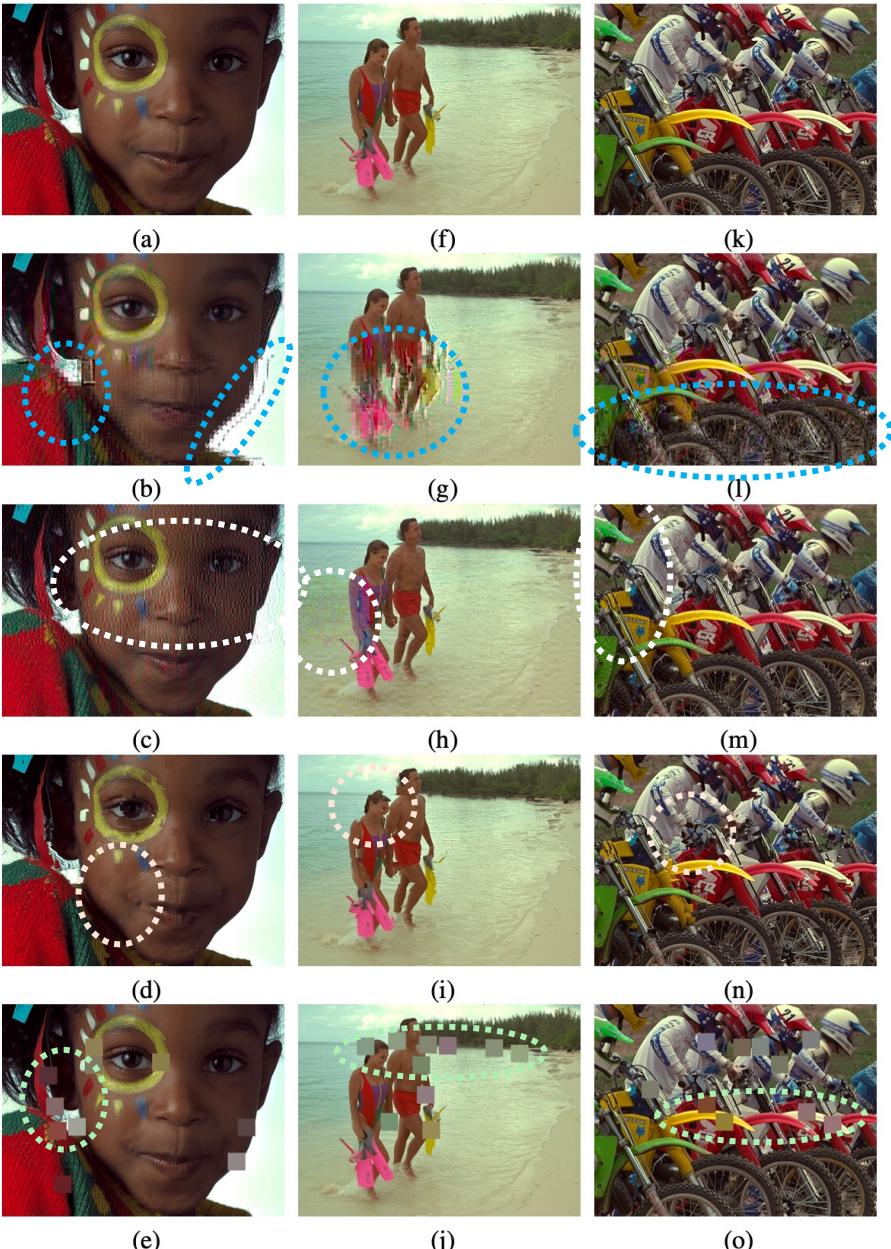


Figure 4.10: Demonstrations of the local distortions (b/g/l: JPEG transmission errors, c/h/m: JPEG2000 transmission errors, d/i/n: non eccentricity pattern noise, e/j/o: local block-wise distortions of different intensities). Figure (a), Figure (f), and Figure (k) are reference images from the TID2013 database.

# Single Distortion Type Evaluation

Table 4.7: The average SRCC results of the individual distortion type on the KADID-10k database. The local distortions are highlighted in blue and the top two results are highlighted in bold.

Distortion Type		BLIINDS-II [91]	BRISQUE [10]	ILNIQE [102]	CORNIA [104]	HOSA [103]	WaDIQaM [35]	NLNet
Blurs	Lens blur	0.781	0.674	<b>0.846</b>	0.811	0.715	0.730	<b>0.914</b>
	Gaussian blur	0.880	0.812	<b>0.883</b>	0.866	0.852	0.879	<b>0.914</b>
	Motion blur	0.482	0.423	<b>0.779</b>	0.532	0.652	0.730	<b>0.899</b>
Color distortions	Color diffusion	0.572	0.544	0.678	0.243	0.727	<b>0.833</b>	<b>0.916</b>
	Color saturation 2	0.602	0.375	0.677	0.120	<b>0.841</b>	0.836	<b>0.909</b>
	Color quantization	0.670	0.667	0.676	0.323	0.662	<b>0.806</b>	<b>0.853</b>
	Color shift	-0.139	-0.182	0.090	-0.002	0.050	<b>0.421</b>	<b>0.777</b>
Compression	Color saturation_1	<b>0.091</b>	<b>0.071</b>	<b>0.027</b>	<b>-0.019</b>	<b>0.216</b>	<b>0.148</b>	<b>0.604</b>
	JPEG compression	0.414	0.782	<b>0.804</b> ↑ <b>6.2%</b>	0.556	0.582	0.530	<b>0.866</b>
Noise	JPEG 2000 compression	0.655	0.516	<b>0.790</b> ↑ <b>6.3%</b>	0.342	0.608	0.539	<b>0.853</b>
	Denoise	0.457	0.221	<b>0.856</b> ↑ <b>9.7%</b>	0.229	0.247	0.765	<b>0.953</b>
	White noise in color component	0.757	0.718	0.841	0.418	0.745↑ <b>1.1%</b>	<b>0.925</b>	<b>0.936</b>
	Multiplicative noise	0.702	0.674	0.682	0.306	0.776↑ <b>5.0%</b>	<b>0.884</b>	<b>0.934</b>
	Impulse noise	0.547	-0.543	0.808	0.219	0.254↑ <b>10.2%</b>	<b>0.814</b>	<b>0.916</b>
Brightness change	White Gaussian noise	0.628	0.708	0.776	0.357	0.680↑ <b>1.7%</b>	<b>0.897</b>	<b>0.914</b>
	Brighten	<b>0.458</b>	<b>0.575</b>	<b>0.301</b>	<b>0.227</b>	<b>0.753</b>	<b>0.685</b>	<b>0.822</b>
	Darken	0.439	0.405	0.436	0.206	<b>0.744</b>	0.272	<b>0.647</b>
Spatial distortions	Mean Shift	0.112	0.144	0.315	0.122	<b>0.591</b>	<b>0.348</b>	0.335
	Jitter	0.629	0.672	0.441	0.719	0.391	<b>0.778</b>	<b>0.899</b>
	Pixelate	0.196	0.648	0.577	0.587	<b>0.702</b>	0.700	<b>0.814</b>
	Quantization	<b>0.781</b>	0.714	0.571	0.259	0.681	0.735	<b>0.791</b>
	Color block	-0.020	0.067	0.003	0.094	<b>0.388</b>	0.160	<b>0.440</b>
Sharpness and contrast	Non-eccentricity patch	0.083	0.191	0.218	0.121	<b>0.461</b>	0.348	<b>0.433</b>
	High sharpen	-0.015	0.361	<b>0.681</b>	0.114	0.230	0.558	<b>0.932</b>
	Contrast change	0.062	0.105	0.072	0.125	<b>0.452</b>	0.421	<b>0.513</b>

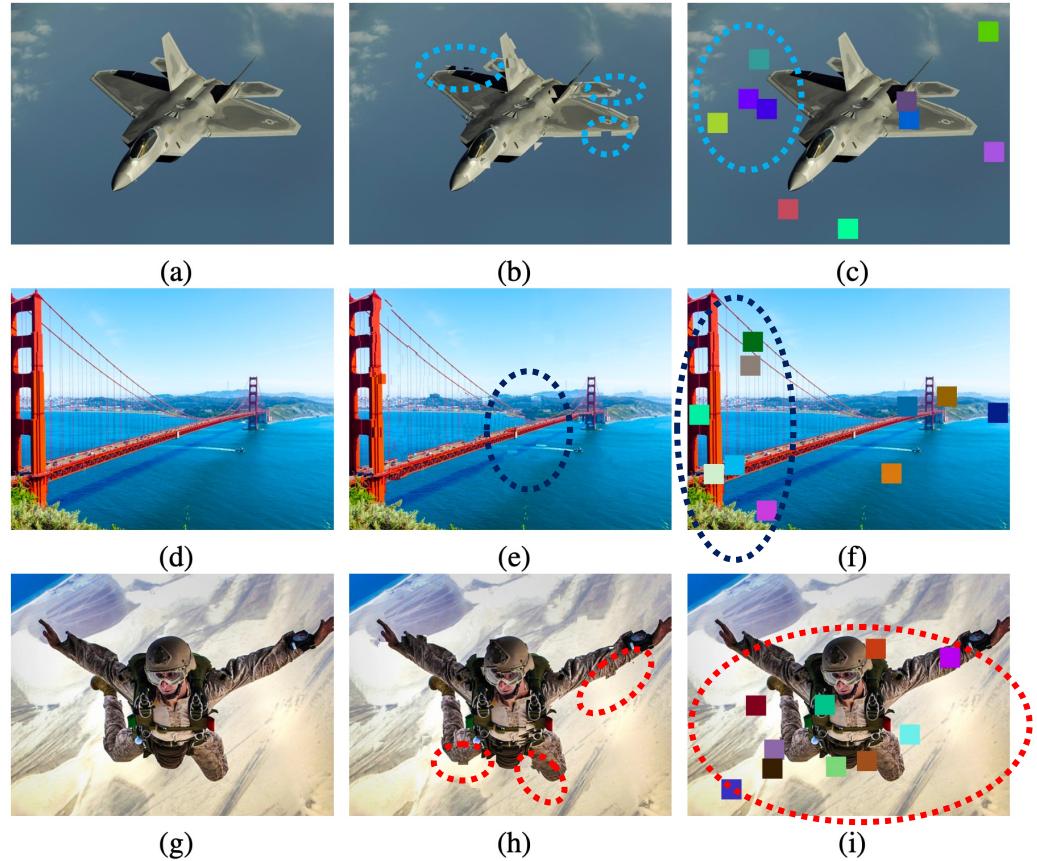


Figure 4.11: Demonstrations of the local distortions (b/e/h: non-eccentricity patch and c/f/i: color block). Figure (a), Figure (d), and Figure (g) are reference images from the KADID-10k database.

# Takeaways and Future Work

- ✓ [Non-local & Local Modeling]

- (1) The Non-local Modeling is complementary to traditional local methods.
- (2) CNN's Local Modeling features are effective and robust.

- ✓ [Global & Local Distortions]

- (1) Handle a wide variety of Global Distortions: globally and uniformly distributed with non-local recurrences.
- (2) Maintain sensitivity to Local Distortions: local nonuniform-distributed distortions in a local region.
- (3) Better assess the quality of Noisy and Compressed Images.

- ✓ [Generalization Capability] Cross-dataset Setting → High Generalization Capability

- ✓ [Future Work] Non-local Statistics [1, 2]

Credit:

[1] Zontak *et al.*, Internal Statistics of a Single Natural Image, In CVPR 2011

[2] Buades *et al.*, A Non-local Algorithm for Image Denoising, In CVPR 2005



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

Yimin Hou<sup>1</sup>, Shuyue Jia<sup>1, 2</sup>, Xiangmin Lun<sup>1</sup>, Ziqian Hao<sup>3</sup>, Yan Shi<sup>1</sup>,  
Yang Li<sup>4</sup>, Rui Zeng<sup>5</sup>, and Jinglei Lv<sup>5\*</sup>

<sup>1</sup> School of Automation Engineering, Northeast Electric Power University

<sup>2</sup> Department of Computer Science, City University of Hong Kong

<sup>3</sup> Jinan Vocational College

<sup>4</sup> School of Electrical Engineering, Northeast Electric Power University

<sup>5</sup> 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 (*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
- ▶ **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 (Signal-to-noise Ratio)
- Different brain electrical conductivity ← different anatomical structures of brain
- Electrodes' position error

Feature Extraction

EEG Electrodes'  
Structure Modeling

## ► 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)

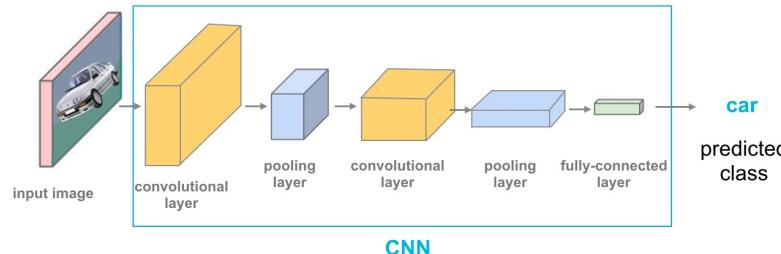
Time-resolved or Window-based  
Signal Sampling

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

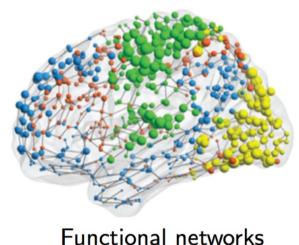
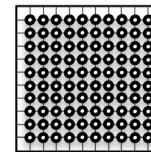
- [most literature] Subject-level prediction (Our Work)
- Group-level prediction (Our Work)

# Motivation

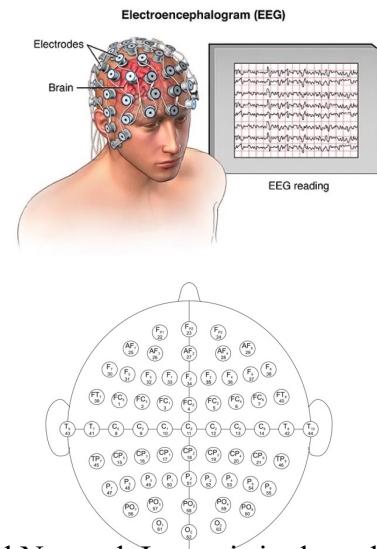
Convolutional Neural Networks:



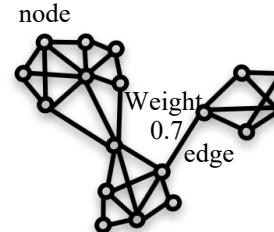
- **Module:** Convolution → Pooling → Fully-connected
- **Pros:** Translation Equivalence, Translation Invariance, Weight Sharing
- **Modeling:** Euclidean-Structured Data (e.g. Image, Speech, Natural Language)
- Neuroscience research has increasingly emphasized **Brain Network Dynamics**
  - Model **Functional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



measure  
→  
← interpret



model  
→  
← reflect



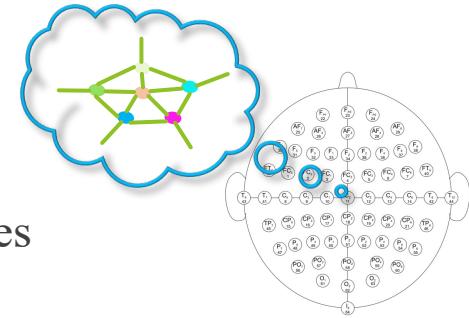
## Our Question

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

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

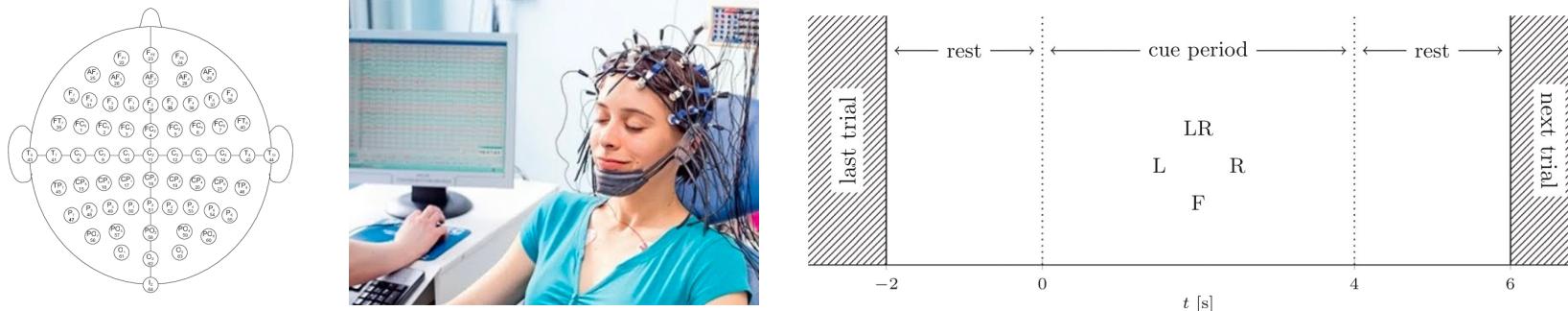
# Can we directly apply convolutions on graphs?

- ▶ Traditional CNN **cannot** directly process graph signals
  - ▶ **Graphs are irregular** (*i.e.*, unordered and vary in size)
  - ▶ Convolutions **cannot keep translation invariance** on 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

- ▶ 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)

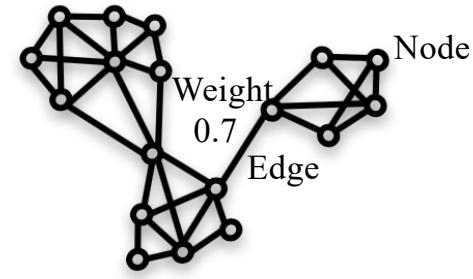


- ▶ **109 subjects** (the largest number of participants in the field of EEG Motor Imagery)
- ▶ Task: **4-class EEG Motor Imagery Classification**
  - ✓ Imagining (Task 1) left fist, (Task 2) right fist, (Task 3) both fists, (Task 4) both feet
- ▶ For each subject, **3 runs, 7 trials, 4 classes** → 84 trials in total
- ▶ For each trial, **4 seconds** experimental duration, **160 Hz** Sampling Rate → **640 Time Points**
- ▶ We applied **Time-resolved Method** for real-time Applications:
  - ✓ 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 nodes  $\mathbf{x}$  and  $\mathbf{y}$
- $\mu$  is the mean,  $\sigma$  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  $\mathbf{L}$
- $\Lambda$ : Fourier modes (frequencies) → the diagonal is the **ordered** and **real nonnegative** eigenvalues of  $\mathbf{L}$

### *Graph Fourier Transform*

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

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

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

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

# Preliminary: Graph Convolution via Graph Fourier Transform

Convolution in the spatial domain equals

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

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

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

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

Hamada Product

Element-wise Multiplication

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

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

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

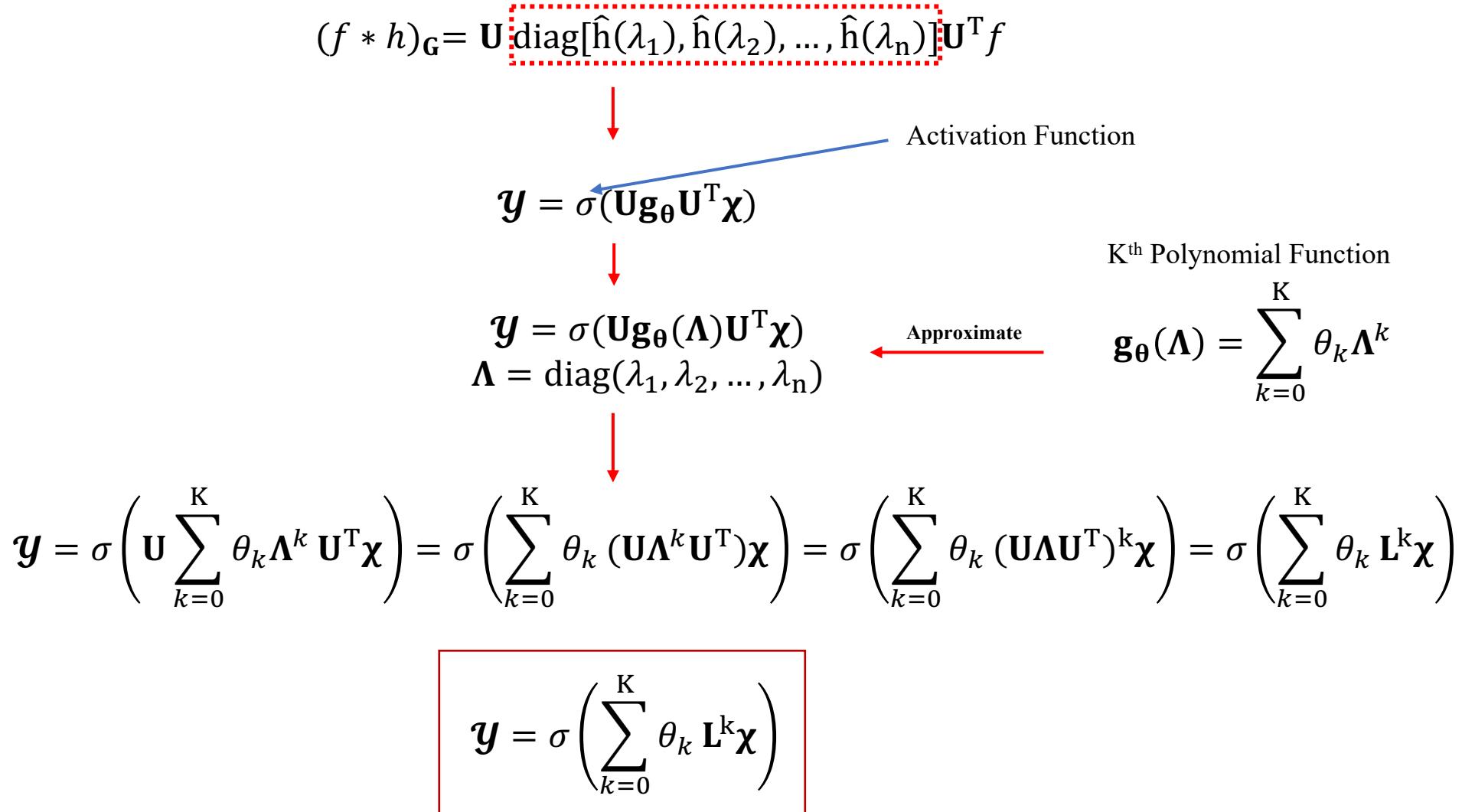
[n × n]

[n × n]

[n × n]

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

# Graph Convolution



# Graph Convolution

“Node Aggregation”  
*K* is Filter Size

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

Convolution:  
Weighted Sum

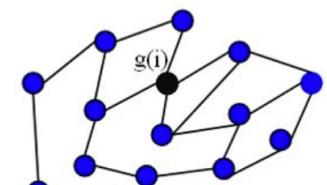
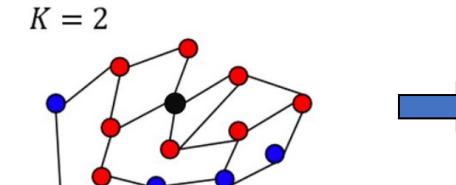
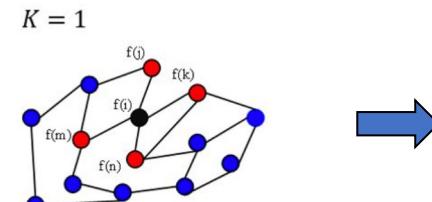
Weight Sharing → Translation Invariance

No need for Fourier Transform

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

“Laplace Operator”  
Local connectivity

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



Pros:

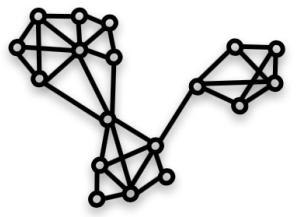
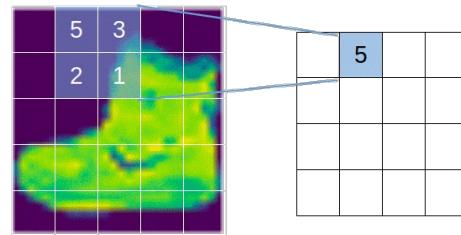
1. No need for Spectral Decomposition
2. Less number of parameters (decrease model complexity) →  $K \ll n$

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

Image Credit: in the public domain.

# Pooling on Graphs (Graph Coarsening)

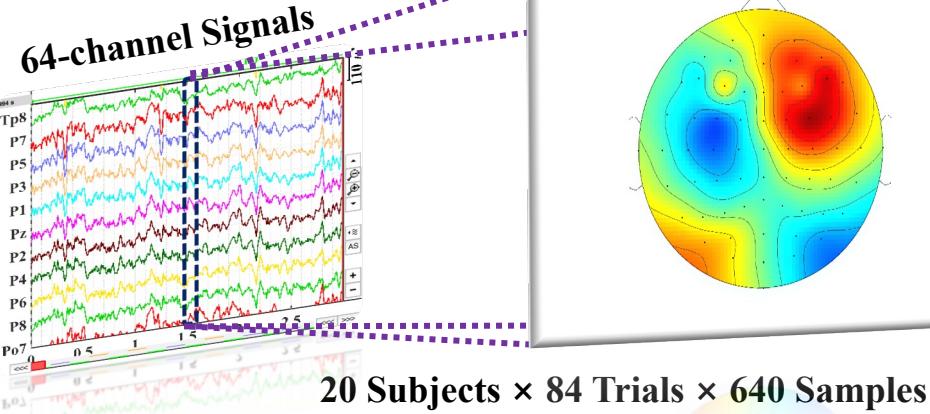
- Traditional CNN doesn't need to consider **neighbors** after convolutions
  - **[Euclidean Structure]** The output feature maps are regular
  - The neighbors are “meaningful”
- GCNs need to consider neighbors after convolutions
  - **[Non-Euclidean Structure]** The output graphs' nodes are not arranged 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)



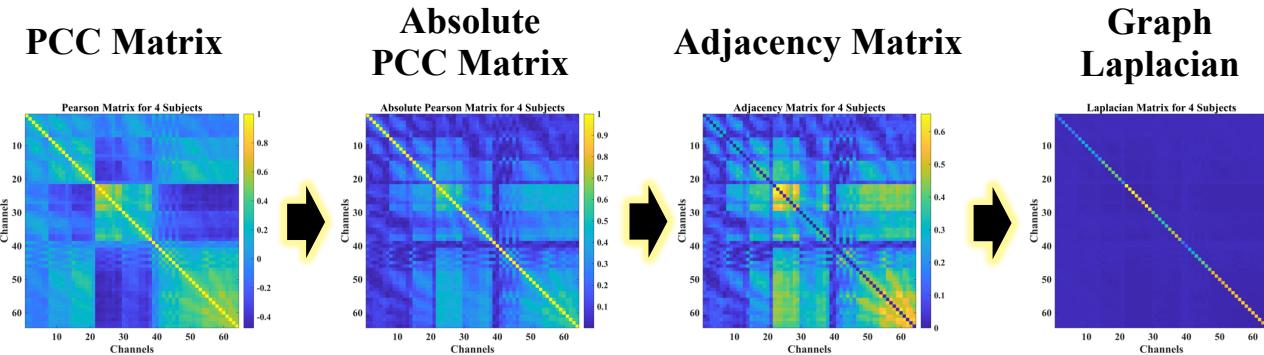
$$-W_{ij}\left(\frac{1}{d_i} + \frac{1}{d_j}\right)$$

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

### (i) EEG Data Acquisition

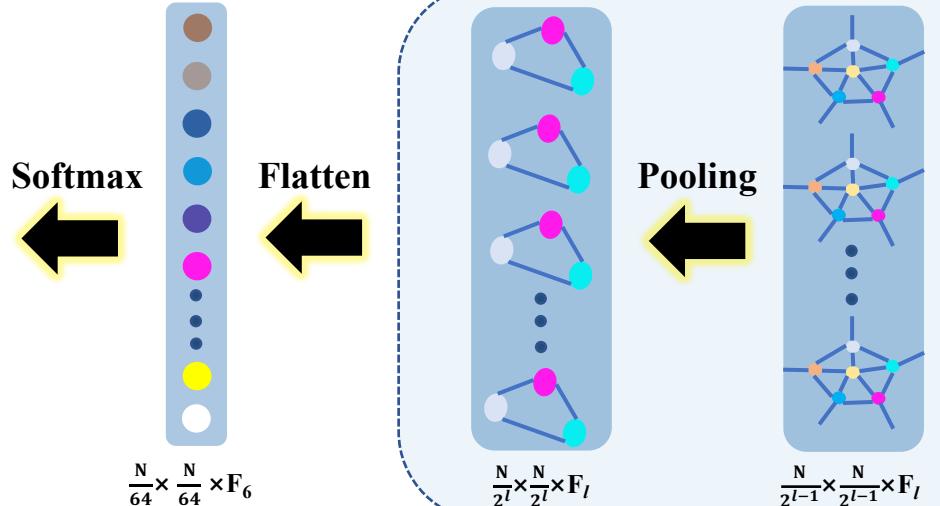


### (ii) Correlations between EEG Electrodes

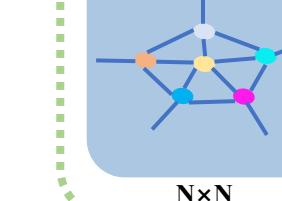
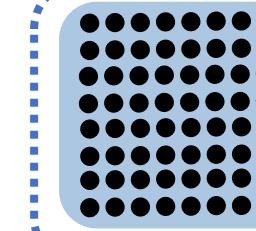


Real-time 64-channel Raw EEG Signals

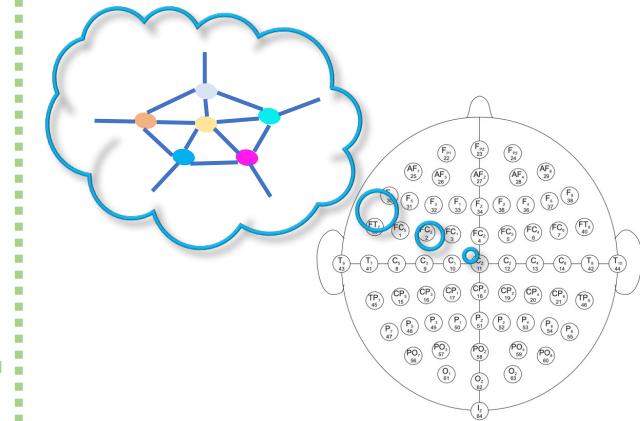
### (iv) The GCNs-Net



GCN



### (iii) Graph Representation



# Correlation among EEG electrodes

## Two Subjects: Subject 10 and 5

### Problem: Individual Variability

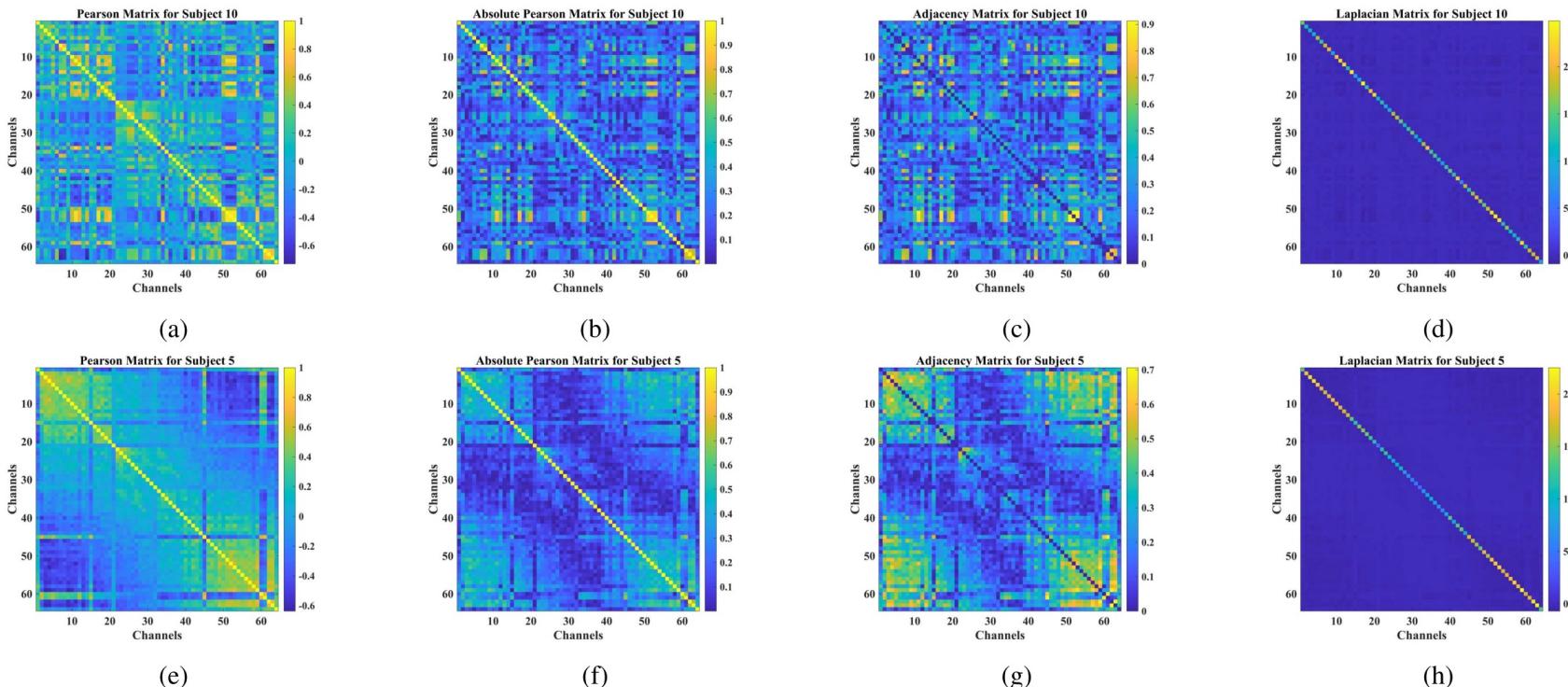


Fig. 6. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for Subjects 10 and 5 from the PhysioNet dataset. (a) PCC matrix for Subject 10. (b) Absolute PCC matrix for Subject 10. (c) Adjacency matrix for Subject 10. (d) Graph Laplacian for Subject 10. (e) PCC matrix for Subject 5.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

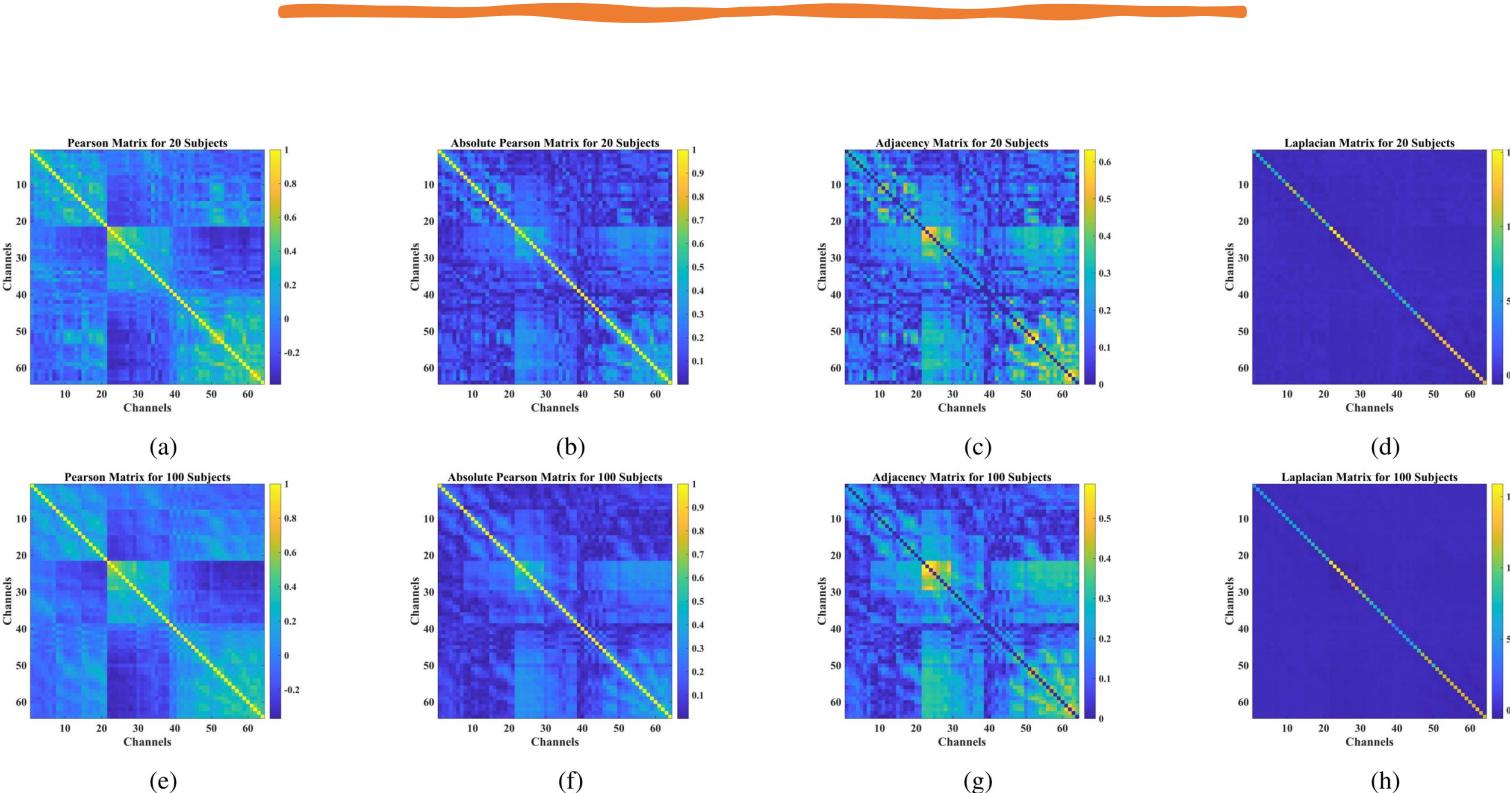


Fig. 2. PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for 20 and 100 subjects, respectively, from the PhysioNet dataset. (a) PCC matrix for 20 subjects. (b) Absolute PCC matrix for 20 subjects. (c) Adjacency matrix for 20 subjects. (d) Graph Laplacian for 20 subjects. (e) PCC 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 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:  $\mathbf{y}$  is labels,  $\hat{\mathbf{y}}$  is the final output tasks

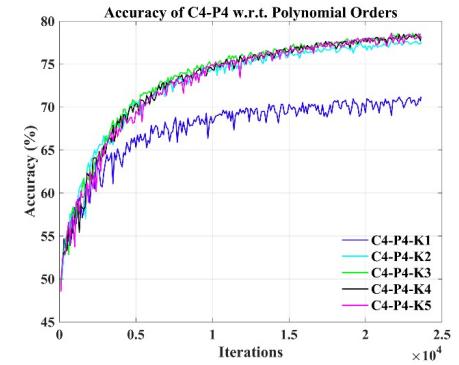
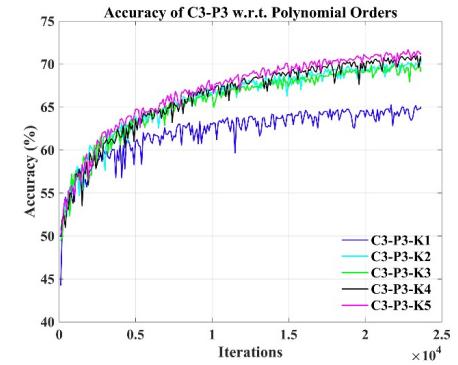
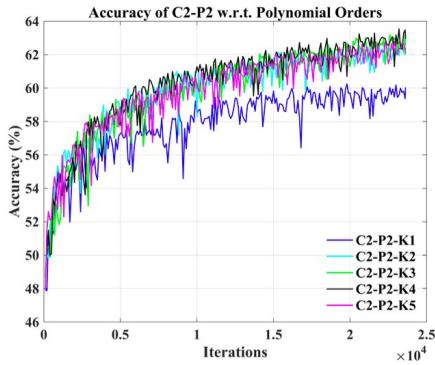
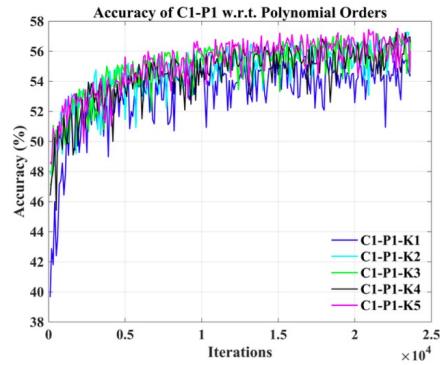
$$\hat{y}_i = \operatorname{argmax}\left(\frac{e^{y_i}}{\sum_{i=1}^4 e^{y_i}}\right)$$

- **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.

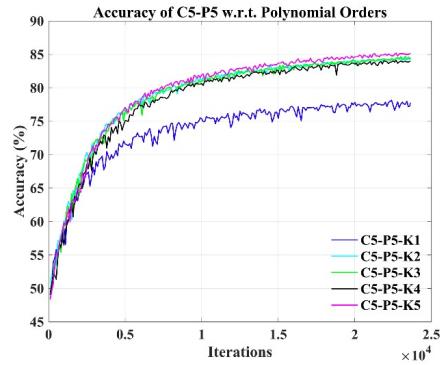
# Ablation Study



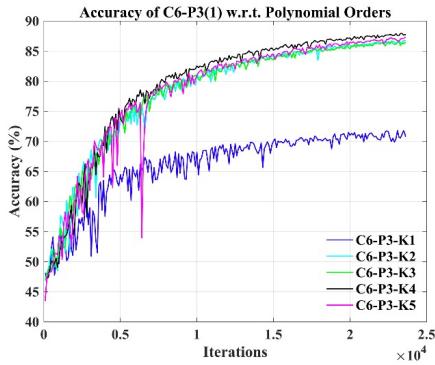
K1

Poor Performance

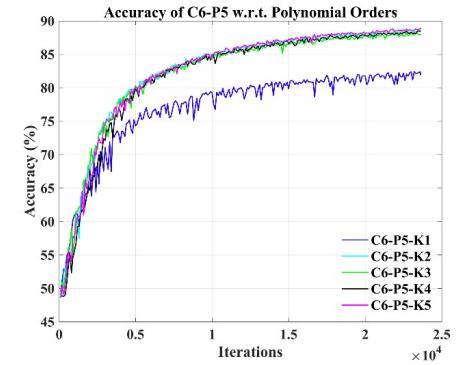
(a)



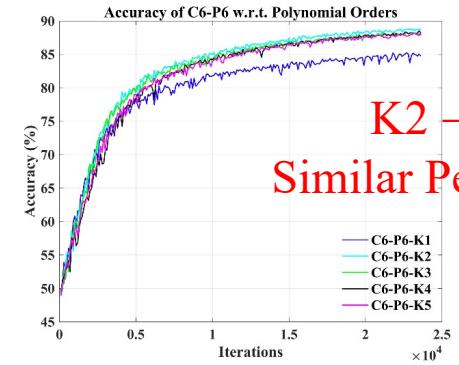
(e)



(f)



(g)



(h)

$K2 \rightarrow K5$

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
<b>Author</b>	<b>89.39%</b>	<b>88.57%</b>	—	Group		<b>20</b>
	<b>88.14%</b>	—	—	Group	<b>GCNs-Net</b>	<b>100</b>
	<b>98.72%</b>	<b>93.06%</b>	—	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 Hou<sup>1</sup>, Shuyue Jia<sup>1, 2 \*</sup>, Xiangmin Lun<sup>1</sup>, Shu Zhang<sup>3</sup>, Tao Chen<sup>1</sup>, Fang Wang<sup>1</sup>, and Jinglei Lv<sup>4</sup>

<sup>1</sup> School of Automation Engineering, Northeast Electric Power University

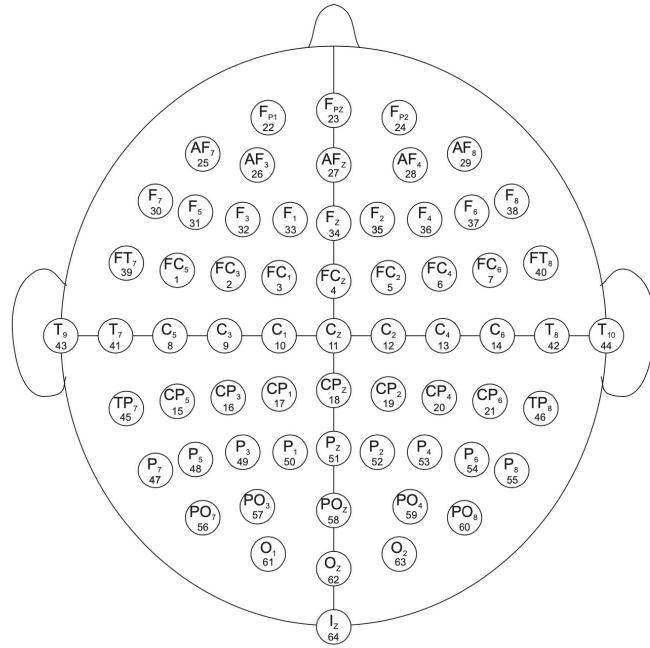
<sup>2</sup> Department of Computer Science, City University of Hong Kong

<sup>3</sup> School of Computer Science, Northwestern Polytechnical University

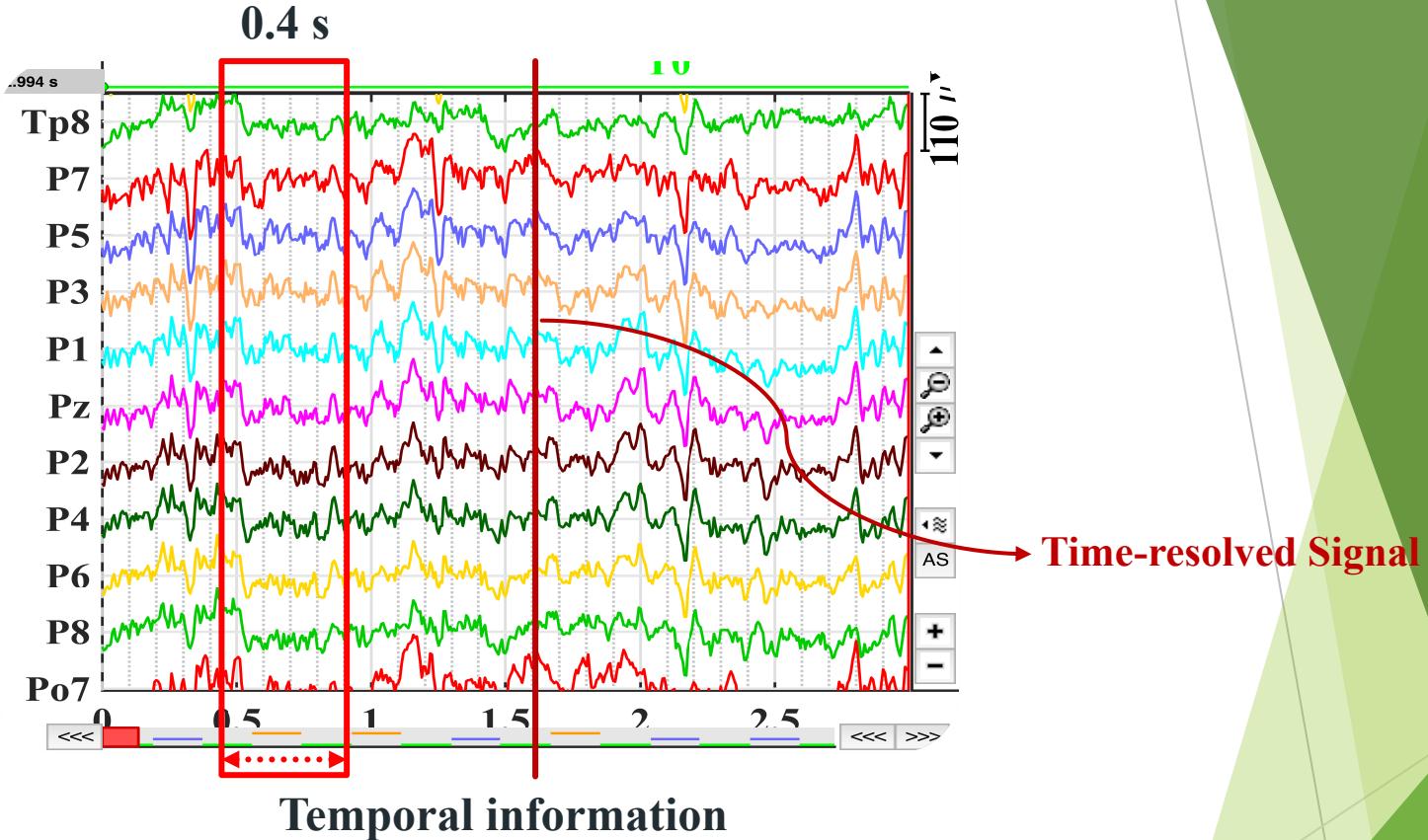
<sup>4</sup> School of Biomedical Engineering and Brain and Mind Center, The University of Sydney

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

# One Problem of the GCNs-Net and Motivation



Spatial information



Temporal information

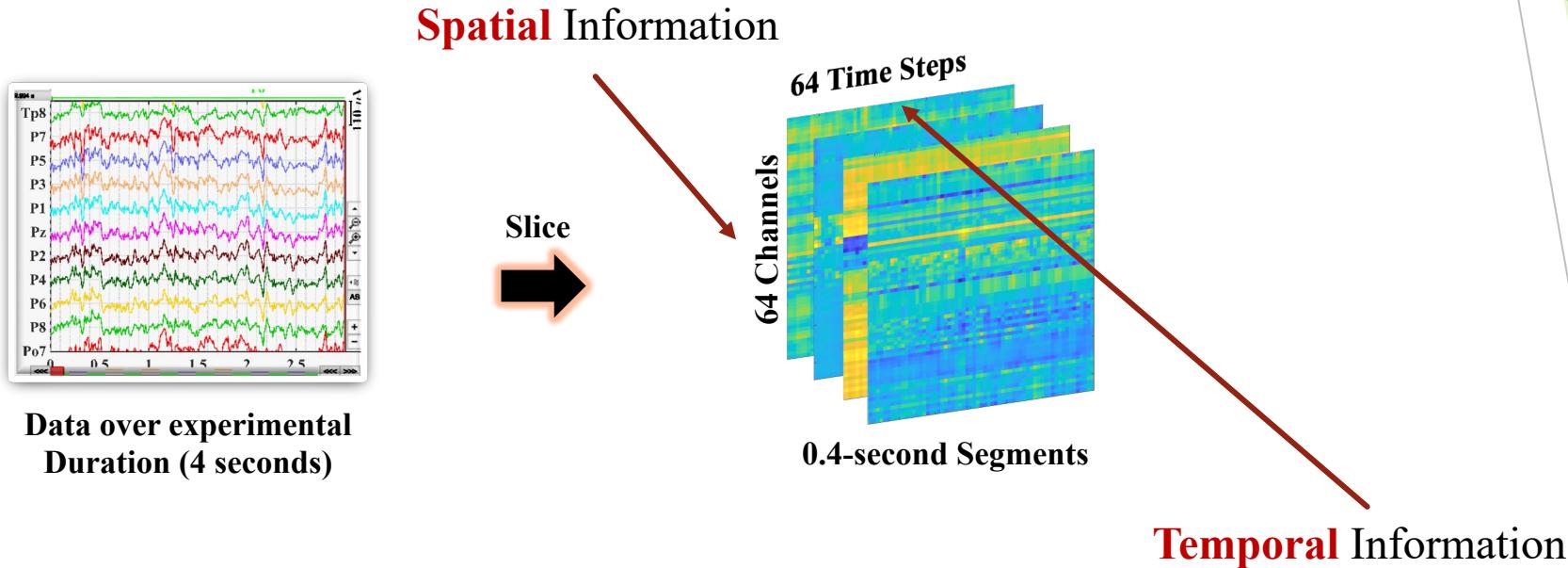
- ✓ GCNs-Net is based on **Time-Resolved Signal** → doesn't consider **Temporal Information**

## Motivation:

- ✓ **[Spatial-Temporal Analysis]** Consider **Temporal** and **Spatial** Information at the same time from EEG signals
- ✓ **[Responsive]** Maintain High Responding Time

Image Credit: The PhysioNet Dataset.

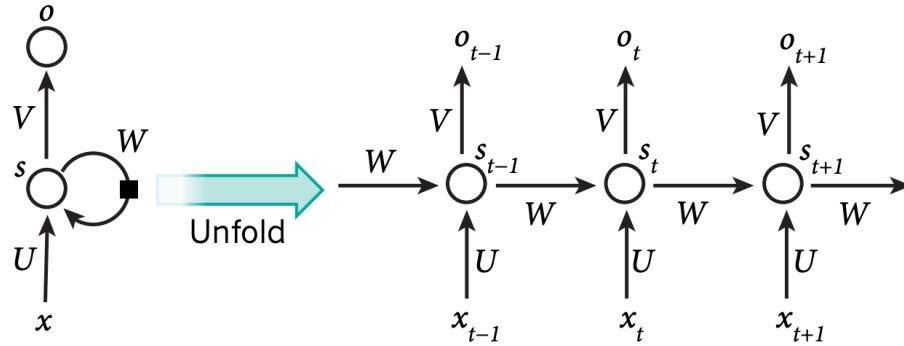
# 64-channel Raw EEG Signals Acquisition



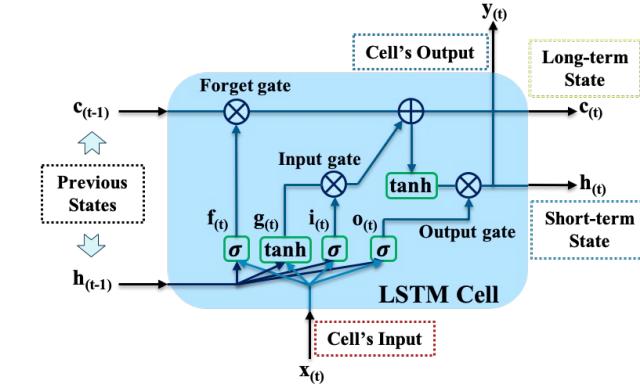
- ✓ 4-s (experimental duration) Signals → 0.4-s segments over time
- ✓ Each Segment: 64 channels × 64 time steps
- ✓ Pre-processed Data: **Temporal** Information + **Spatial** Information

# Temporal Information Extraction

unrolling the network through time



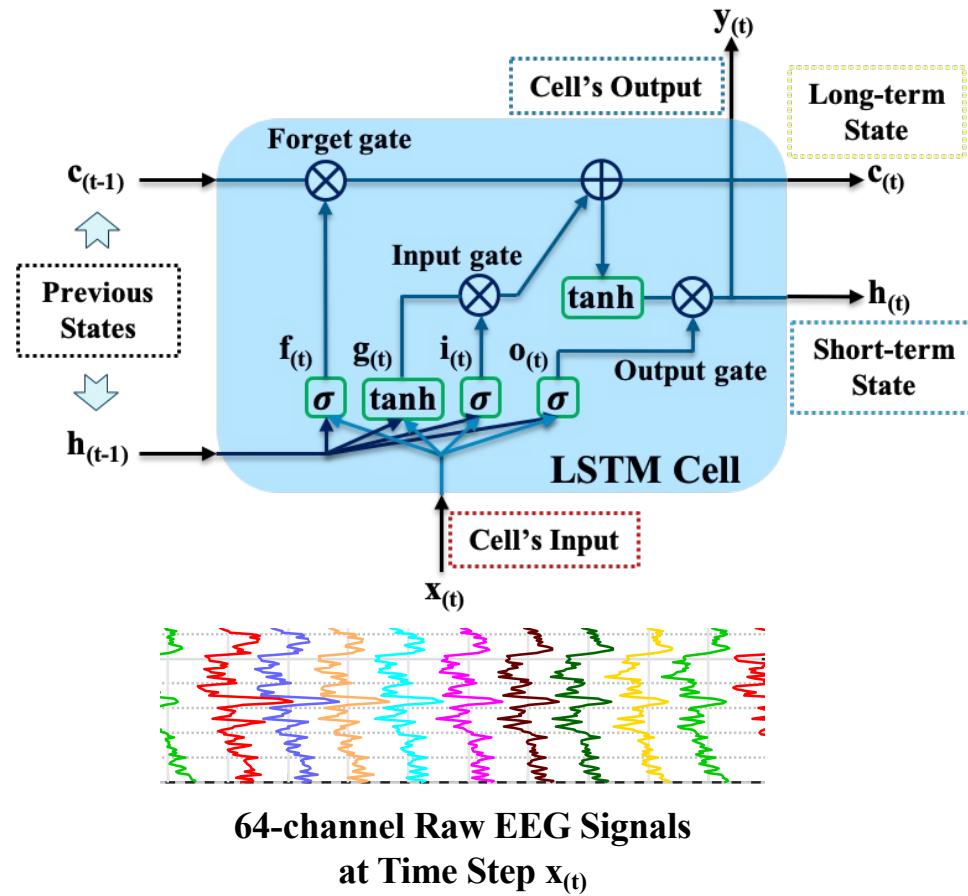
Recurrent Neural Network (RNN)



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  $\mathbf{x}_{t-1}$  → continually learn from time series
- ✓ **LSTM**: better capture long-range sequence dependencies
- ✓ **GRU**: lightweight architecture with comparable performance

# Long Short-term Memory (LSTM)



- ✓ 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  
→ 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**

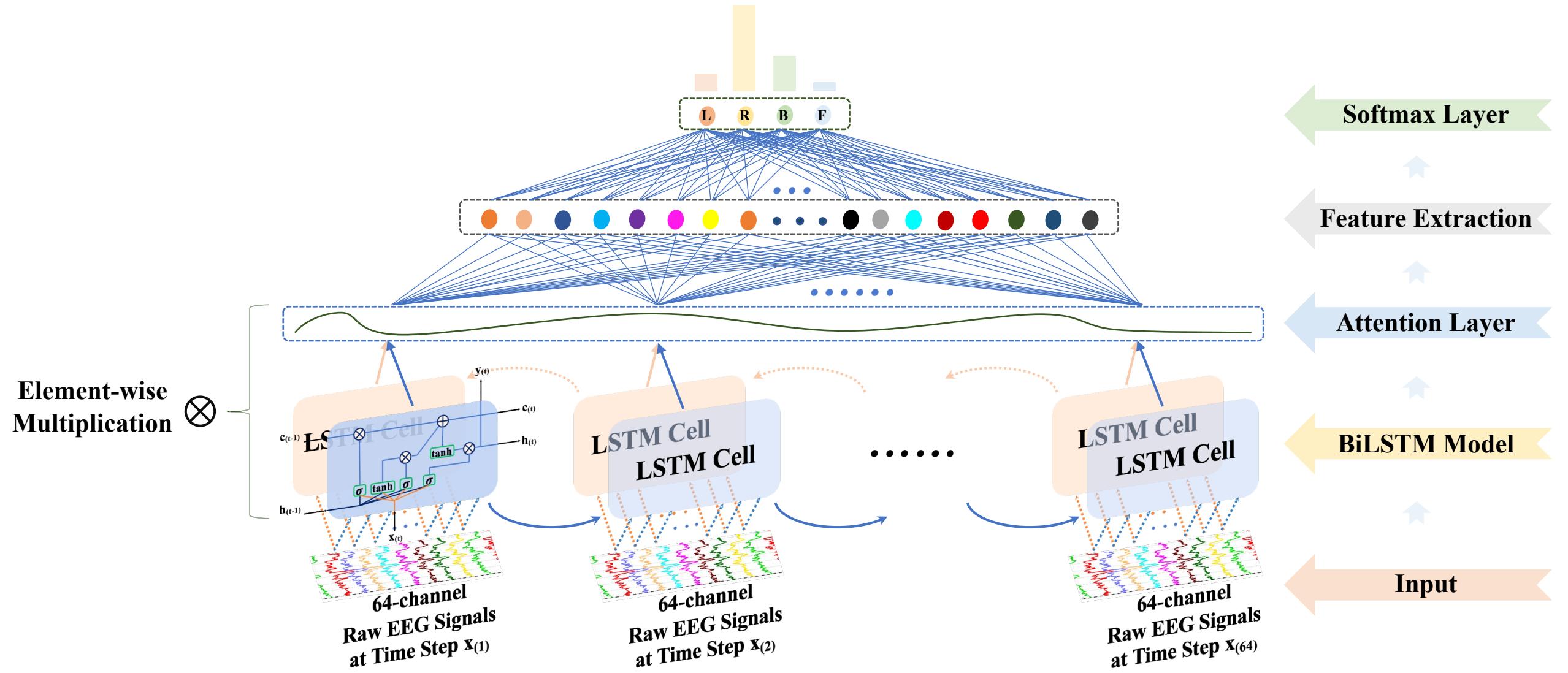
FC Layer  $\mathbf{U}_t = \tanh(\mathbf{W}_w \mathbf{y}_t + \mathbf{b}_w)$

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

Weighted Sum

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

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



# Model Design Ablation Study

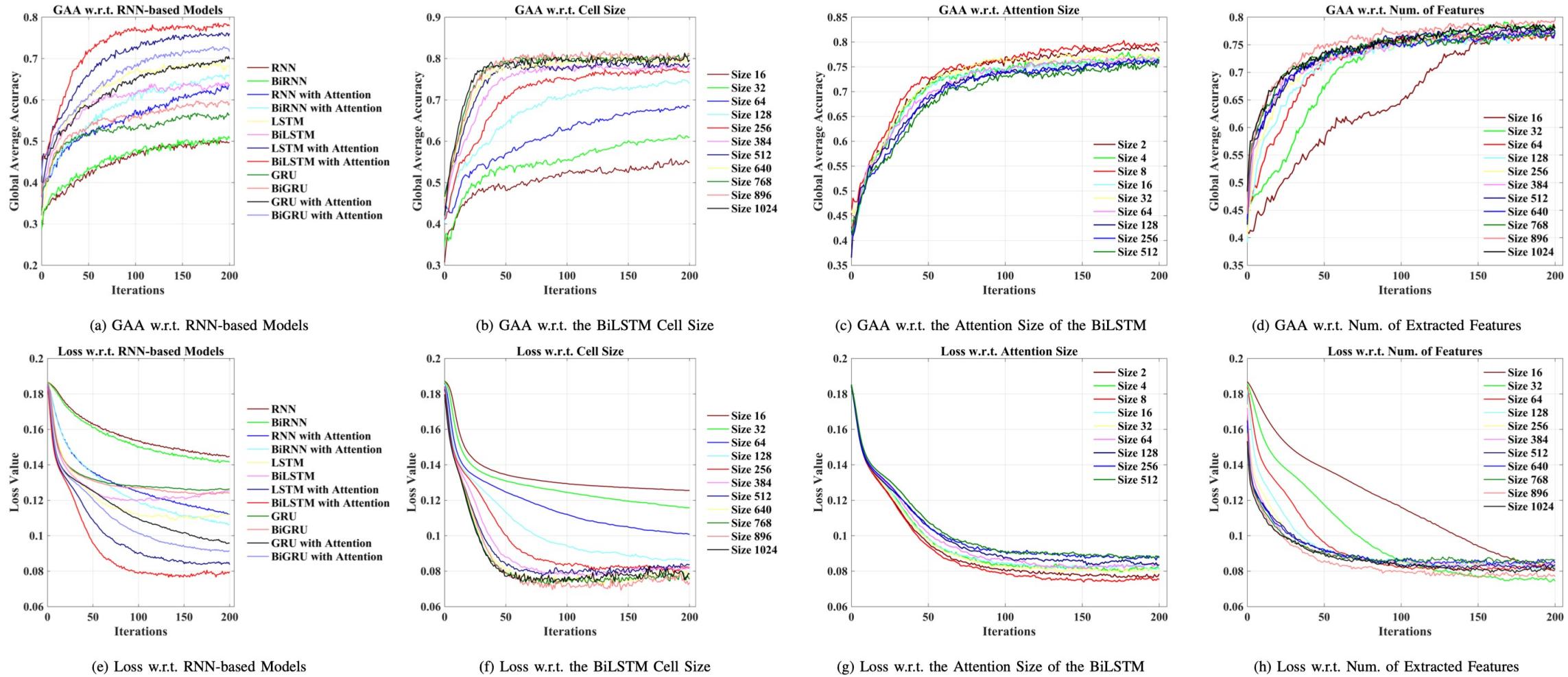
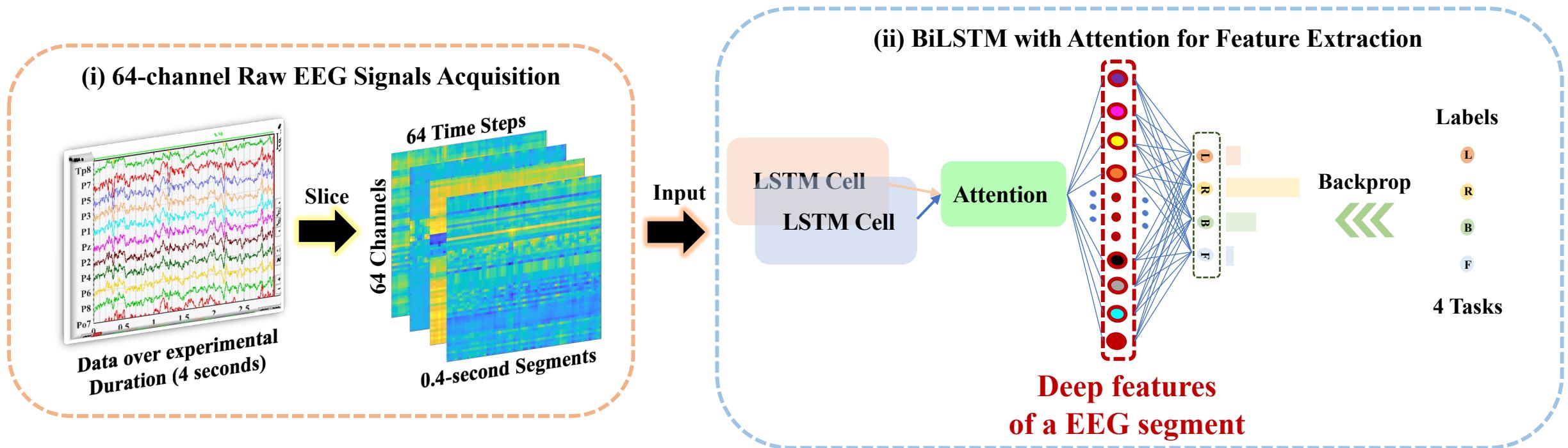


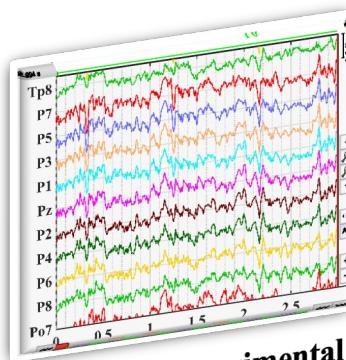
Fig. 3: Models and Hyperparameters Comparison w.r.t. the RNN-based Methods for Feature Extraction

# Topological Structure of Features

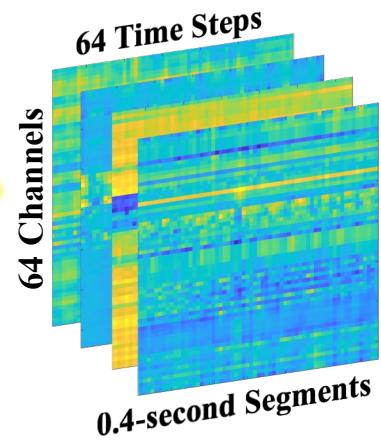


✓ **Deep feature mining** → Intra-feature relationship → Intra-feature Modeling

### (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

### (iii) Graph Convolutional Neural Network

Labels

L

R

B

F

Backprop

Softmax

Flatten

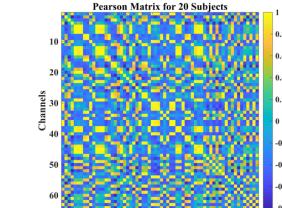
Max Pooling

GCN

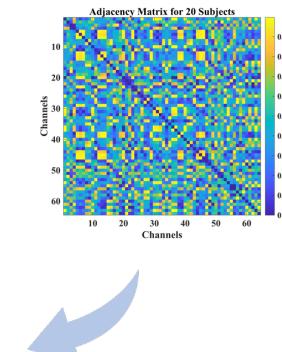
Features



Pearson Matrix

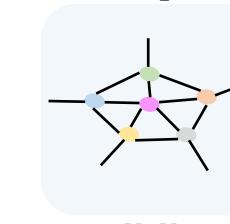


Adjacency Matrix

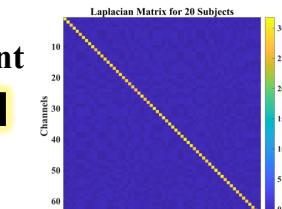


4 Tasks

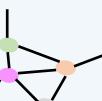
Graph



Laplacian Matrix



Present



# Topological Structure of Features

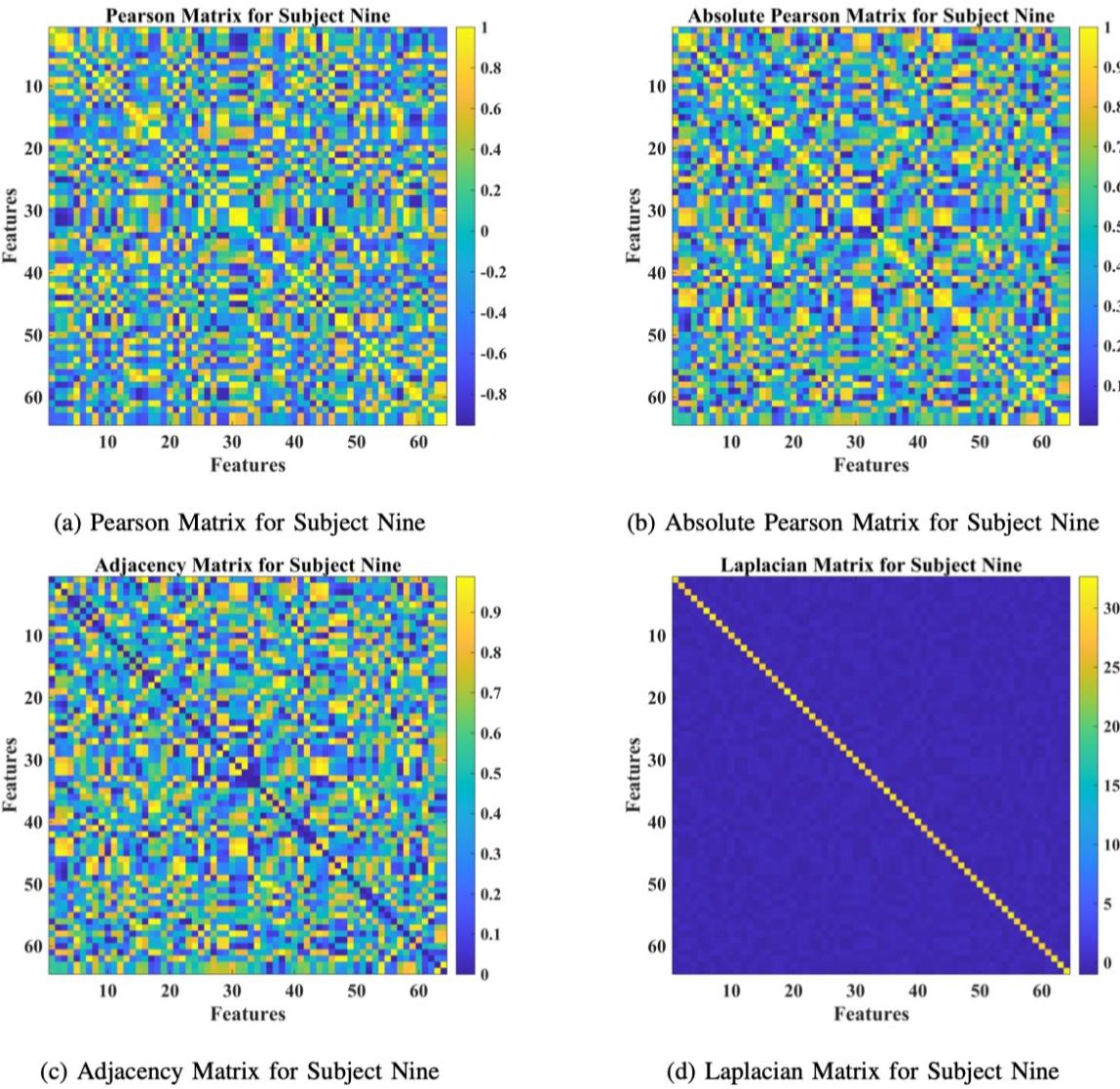
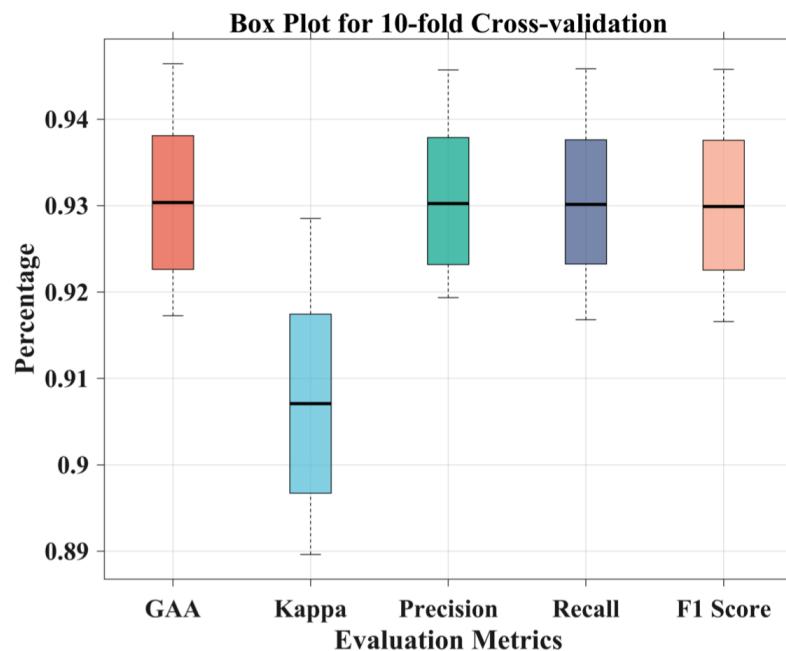
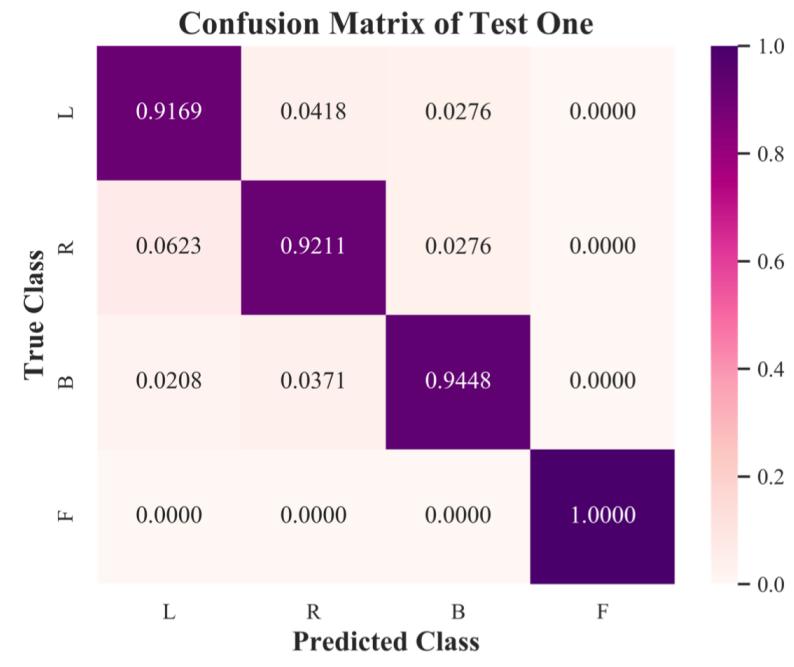


Fig. 4: The Pearson, Absolute Pearson, Adjacency, and Laplacian Matrices for Subject Nine.

# 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%
<b>Average</b>	<b>95.48%</b>	<b>93.94%</b>	<b>95.50%</b>	<b>95.61%</b>	<b>95.35%</b>

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
<b>This work</b>	<b>98.81%</b>	<b>Attention-based BiLSTM-GCN</b>	

# 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) Advance the **Clinical Translation** of the EEG MI-based BCI technology to meet diverse demands, such as those of paralyzed patients.
- (2) The **Highest Accuracy and Time-Resolved Prediction**.

## ✓ [Future Work]

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

# Thank you!

Any question?