Multi-Scale GCN-LSTM for EEG Seizure Detection

Zimu Zhao
School of Engineering
EPFL
Lausanne, Switzerland
zimu.zhao@epfl.ch

John Taylor
School of Engineering
EPFL
Lausanne, Switzerland
john.taylor@epfl.ch

Daniele Belfiore
School of Engineering
EPFL
Lausanne, Switzerland
daniele.belfiore@epfl.ch

Ahmed Abdelmalek
School of Engineering
EPFL
Lausanne, Switzerland
ahmed.abdelmalek@epfl.ch

***Abstract*—Seizure detection from EEG is a clinically critical but technically complex task due to the high-dimensional, non-stationary, and noisy nature of brain signals. While conventional deep learning models like LSTMs and Transformers learn temporal patterns effectively, they often ignore the brain's spatial structure. This work investigates graph-based models for EEG seizure detection using a subset of the TUSZ dataset. We propose a multi-scale GCN-LSTM architecture that processes EEG features at three temporal resolutions (0.5s, 1s, 2s), each via separate spatial-temporal branches with shared attention-based fusion. Dynamic graphs constructed from Pearson correlation coefficients model functional connectivity per window. Our model significantly outperforms baseline architectures, confirming the utility of spatio-temporal and multi-scale modeling in clinical EEG analysis.**

***Index Terms*—EEG, Seizure Detection, Graph Neural Networks, GCN-LSTM, Multi-Scale Representation**

I. Introduction

Epilepsy affects over 50 million people globally [3], often requiring reliable and timely seizure detection to inform treatment. EEG remains the gold standard for non-invasive brain monitoring [7], capturing time-series voltage signals from multiple scalp electrodes. Manual EEG inspection, however, is labor-intensive and prone to subjectivity, particularly in long-term monitoring, prompting a push toward automated detection.

Traditional deep learning models such as RNNs and CNNs have shown promise in modeling the temporal and local spatial patterns of EEG [6], yet they often overlook the spatial structure of electrode relationships. Since the brain exhibits rich spatial connectivity patterns, modeling EEG as a graph, with electrodes as nodes and connectivity as edges, enables more structured learning. Graph Neural Networks (GNNs), and in particular Graph Convolutional Networks (GCNs), have been applied to model such structures, capturing spatial dependencies that standard sequence models cannot.

In this work, we propose a hybrid, multi-scale GCN+LSTM architecture tailored for seizure detection using EEG. Our model processes feature sequences extracted at three temporal resolutions (0.5s, 1s, and 2s) through parallel GCN+LSTM branches. Each branch operates on dynamic, window-wise Pearson correlation graphs to capture functional brain connectivity. The outputs are fused via cross-scale attention, integrating information across temporal resolutions. We evaluate this architecture against non-graph baselines and basic GNNs on a curated version of the TUH EEG Seizure Corpus, demonstrating its superiority in capturing the complex spatio-temporal structure of seizure events.

II. Related Work

A. Sequential and Spatio-Temporal Models

Deep learning for EEG seizure detection has evolved from simple temporal models to architectures that capture both spatial and temporal dependencies. Early works employed RNNs, especially LSTMs and GRUs, to process the EEG as multivariate time series [4]. CNNs extended this by learning spatial filters across channels [6], with hybrid CNN-RNN models combining local spatial and global temporal modeling.

B. Graph Neural Networks for EEG

GNNs have gained attention for EEG modeling by treating electrodes as graph nodes and defining edges through spatial or functional metrics [5]. Graph Convolutional Networks (GCNs), GraphSAGE, and Graph Attention Networks (GATs) propagate information through these connections. GCNs paired with RNNs or LSTMs can model both spatial and temporal dynamics [1], an approach that aligns well with seizure activity patterns that manifest across both domains.

C. Multi-Scale Temporal Modeling

EEG signals are composed of waveforms with varying durations and frequencies, from brief spikes to longer discharges, necessitating multi-resolution analysis. Recent works [6] explored multi-frequency and multi-resolution approaches, such as downsampling and filter banks, to capture this variability. Our approach adopts a feature-level multi-scale strategy: extracting EEG features over 0.5s, 1s, and 2s windows, each yielding distinct input dimensions. Each timescale is processed by a dedicated GCN+LSTM branch, and the outputs are fused using cross-scale attention to capture transient and sustained dynamics.

D. Real-Time and Practical Considerations

As emphasized by Lee et al. [6], real-world seizure detection systems must support low latency, robust performance, and high interpretability. Dynamic functional graphs—such as Pearson correlation computed per window—provide a compromise between biological relevance and computational feasibility. Our architecture is designed with these constraints in mind, supporting both accuracy and efficient inference.

III. Methodology

A. Dataset and Pre-processing

Corpus. Experiments are conducted on the official TUH EEG Seizure Corpus (TUSZ) subset released for the EE-452 Kaggle competition. It contains 50 training and 25 held-out test patients, each recorded with the international 10–20 montage ($C = 19$ channels, 250 Hz). Continuous recordings are partitioned into non-overlapping 12-second epochs and labelled *ictal* (1) or *background* (0).

Windowing. To capture seizure dynamics at different temporal scales, every 12-second epoch is further subdivided into overlapping windows at three resolutions. Let $X \in \mathbb{R}^{C\times T}$ be the raw EEG epoch, where $T = 3000$ and C is the number of channels. We extract multi-scale segments using sliding windows of lengths

$$w \in \{0.5\text{s}, 1\text{s}, 2\text{s}\}, \qquad \text{stride}(w) = \begin{cases} w/2, & \text{if } w \neq 1\text{s} \\ w, & \text{if } w = 1\text{s} \end{cases}.$$

For each scale w we obtain a tensor

$$\phi_w(X) \in \mathbb{R}^{C\times T_w \times d_w},$$

where T_w is the number of windows and d_w the feature dimension ($d_{0.5} = 145$, $d_1 = 208$, $d_2 = 210$).

Feature Extraction. Per–window features are summarised in Table II (Appendix) and fall into three groups:

1) *Time domain* (9 stats): mean, MAD, σ, peak-to-peak, zero-crossings, skewness, kurtosis, RMS, line-length.
2) *Frequency domain*: Power Spectral Density (first $\min(128, \lfloor \frac{\texttt{win\_len}}{2} \rfloor + 1)$ bins) and band-powers $\{\delta, \theta, \alpha, \beta, \gamma\}$.
3) *Wavelet domain*: 63 Morlet-scale energies and their 5 band-averaged summaries.

For PSD the number of bins follows

$$n_{\text{freq}} = \left\lfloor \frac{\texttt{win\_len}}{2} \right\rfloor + 1.$$

Normalisation. Channel-wise Z-score is applied using training means μ_c and standard deviations σ_c:

$$\hat{x}_{c,t} = \frac{x_{c,t} - \mu_c}{\sigma_c}.$$

Implementation note. The multi-scale pipeline is illustrated in Figure 2.
All features are stored on disk as `.npy` tensors to avoid on-the-fly computation overhead during model training.

B. Graph Construction

Scalp EEG affords an explicit spatial structure: each of the $C = 19$ electrodes is located at a fixed 3-D coordinate on the 10–20 cap. We exploit this by representing every time-window as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, \ldots, v_{19}\}$ and edge set $\mathcal{E}$ defined by one of three schemes below.

1) Structural K-NN graph (static, used in *all* models). Let $\mathbf{p}_i \in \mathbb{R}^3$ be the electrode coordinate for node v_i. For each node we connect to its k nearest neighbours ($k = 4$) under the Euclidean distance $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$. The undirected adjacency is

$$A_{ij} = A_{ji} = \mathbb{1}\big[v_j \in \text{KNN}(v_i) \ \vee \ v_i \in \text{KNN}(v_j)\big].$$

We use the standard symmetric normalisation $\tilde{A} = D^{-1/2}(A + I)D^{-1/2}$ prior to every GNN layer. This fixed graph is pre-computed once and stored as `edge_index_knn.pt`.

2) Grid graph (baseline only). For comparison we also construct a coarse 2-D grid where each electrode is connected to its four virtual lattice neighbours. Empirically this topology under-represents true cortical proximity and yields lower F1 (Table **??**).

3) Dynamic functional graph (PCC, used in GCN+LSTM branches). Within every feature window w_t we compute the Pearson correlation between channel signals,

$$\rho_{ij}^{(t)} = \frac{\text{cov}(x_i, x_j)}{\sigma_i \sigma_j},$$

forming a dense functional matrix $\mathbf{R}^{(t)} \in \mathbb{R}^{C\times C}$. We retain edges with $|\rho_{ij}^{(t)}| > \tau$ (top 30 % absolute correlations) and normalise as above. The resulting time-varying graph $\mathcal{G}_t$ enables the hybrid GCN+LSTM to track evolving connectivity during seizure evolution.

Observation. Static K-NN graphs give the most stable performance across plain GCN / SAGE / GAT runs, whereas dynamic PCC graphs provide an extra performance boost (around +4 pp macro-F1) when coupled with temporal models, justifying their use in the multi-scale GCN+LSTM architecture.

C. Non-Graph Baselines

To quantify the benefit of spatial modelling, we train two architectures that operate on *flattened* feature sequences (no electrode topology is used).

Input representation. For every 12-s epoch we keep the 2-s feature windows (50% overlap) employed in prior work [6]. This gives $T = 11$ windows. Each window yields a channel-wise feature matrix $\Phi_t = [\phi_{1,t}, \ldots, \phi_{C,t}] \in \mathbb{R}^{C\times d}$. We flatten along the channel axis,

$$\mathbf{x}_t = \text{vec}(\Phi_t) \in \mathbb{R}^{C\,d},$$

producing a sequence $\{\mathbf{x}_t\}_{t=1}^{T}$ passed to the models below.

Transformer Encoder. We also implement a lightweight Transformer with $L = 2$ encoder blocks and $H = 4$ heads. Token embeddings are $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{p}_t$ (learnable positional encodings). Each block performs multi-head attention (MHA) followed by a position-wise feed-forward network (FFN):

$$\mathbf{Z} = \text{MHA}(\tilde{\mathbf{X}}), \quad \mathbf{U} = \text{LayerNorm}(\tilde{\mathbf{X}} + \mathbf{Z}),$$
$$\tilde{\mathbf{X}} \leftarrow \text{LayerNorm}(\mathbf{U} + \text{FFN}(\mathbf{U})).$$

Global average pooling over the final tokens produces the segment embedding fed to a sigmoid classifier.
Non-graph baselines achieve competitive macro-F1 (around 0.69). Their performance therefore provides the reference point against which graph-based models (Sec. III-D) are evaluated.

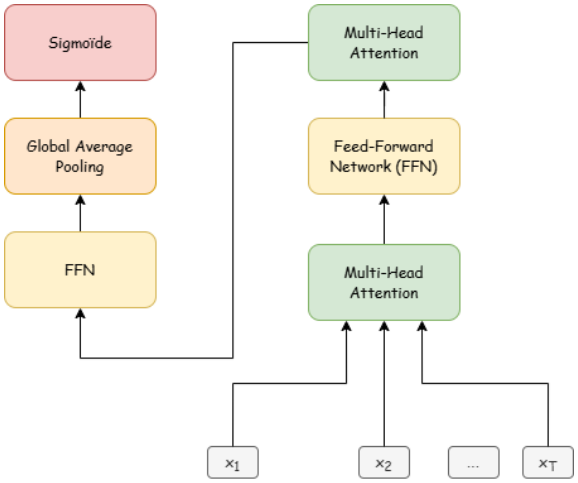

Fig. 1: High-level architecture of the Transformer baseline. Token embeddings $\mathbf{x}_t$ (flattened channel features with positional encodings) pass through $L = 2$ encoder blocks (multi-head self-attention + FFN). Outputs are globally averaged and passed through a sigmoid layer for binary seizure prediction.

D. Graph-Based Models

All graph networks use the electrode graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ described in Sec. III-B. Node features are the per-window vectors $\mathbf{X}_t \in \mathbb{R}^{C\times F}$ extracted in Sec. III-A. We evaluate three levels of spatial-temporal coupling.

(1) Pure Spatial GNNs. GCN, GraphSAGE and GAT are applied to a *static* K-NN graph per 2-s window. A L-layer network produces node embeddings $\mathbf{H}^{(L)}$; global mean pooling yields a segment vector $\mathbf{z}$ passed to an MLP classifier. These models quantify the benefit of spatial propagation alone (Table **??**).

(2) Single-Scale GCN + LSTM. To add temporal context we deploy a two-stage pipeline:

$$\underbrace{\mathbf{z}_t = \mathrm{GCN}(\mathbf{X}_t, \mathcal{G}_t)}_{\text{spatial}}, \qquad \hat{y} = \mathrm{MLP}\big(\mathrm{LSTM}(\mathbf{z}_{1:T})\big),$$

where $T = 11$ for the 2-s windows. Static K-NN and dynamic PCC graphs were both tested; the latter improves macro-F1 by around 4 pp thanks to window-specific connectivity.

(3) Multi-Scale GCN + LSTM (ours). Seizure signatures exist at disparate time-scales. We therefore instantiate three parallel branches $s \in \{0.5, 1, 2\}$ s:

$$\mathbf{h}^{(s)} = \mathrm{LSTM}_s\Big(\big\{\mathrm{GCN}_s(\mathbf{X}_t^{(s)}, \mathcal{G}_t^{(s)})\big\}_{t=1}^{T_s}\Big),$$

each using dynamic PCC graphs at its own resolution. The branch embeddings are fused by a cross-scale attention module $\mathbf{h} = \sum_s \alpha^{(s)}\mathbf{h}^{(s)}$, $\alpha^{(s)} = \mathrm{softmax}(W\mathbf{h}^{(s)})$ and classified with a final MLP. This architecture captures both transient spikes (0.5 s) and slower rhythmic discharges (2 s), delivering our best Kaggle score.

Unlike non-graph models that treat channels as independent or flattened vectors, GNNs exploit the structured inter-channel dependencies of the EEG montage. The proposed hybrid and multi-scale architectures effectively unify spatial and temporal reasoning, yielding substantial improvements in F1 and robustness.

Summary. Compared with channel-flattened baselines, GNNs exploit electrode topology; adding the LSTM captures temporal evolution; multi-scale fusion further boosts robustness, yielding a +10 pp macro-F1 gain over non-graph models.

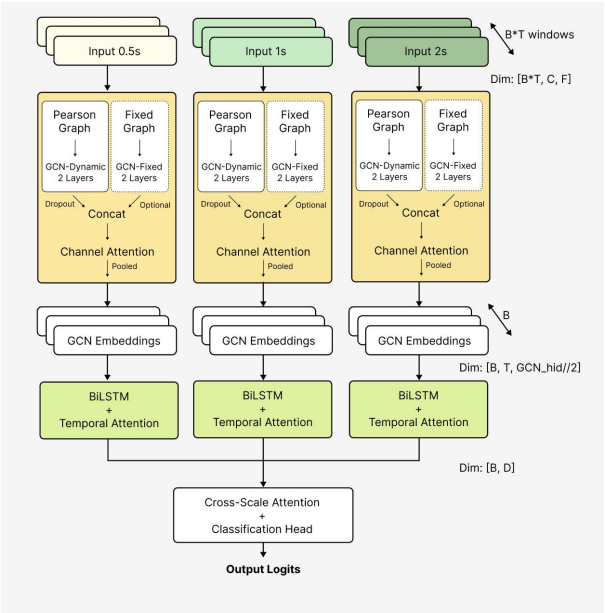

Fig. 2: Schematic of the proposed multi-scale GCN+LSTM: three scales, dynamic PCC graphs, cross-scale attention.

E. Training and Evaluation

Loss and Class Imbalance. To address the heavily imbalanced seizure/background distribution, we minimise a *weighted* binary cross-entropy:

$$\mathcal{L}(p, y) = -\,w_y\big[y\log p + (1-y)\log(1-p)\big], \qquad w_y \propto \tfrac{1}{\mathrm{freq}(y)}.$$

In addition, we apply **minority oversampling** to rebalance training batches to a seizure-to-background ratio of approximately 1:2, effectively improving recall without increasing false positives.

Training Stability and Scheduling. To enhance training stability and convergence, we adopt several techniques: a linear **warm-up** phase over the first three epochs; a **ReduceLROnPlateau** scheduler that halves the learning rate when validation macro-F1 stagnates (down to a floor of 2e-5); and **gradient clipping** with a maximum norm of 5. All models are optimised using Adam or AdamW with L2 regularisation, depending on architecture. Training is early-stopped if no improvement is observed over 10 validation epochs.

Cross-validation. We use **5-fold stratified cross-validation** at the patient level to avoid inter-subject leakage. Each fold uses 80% of patients for training and 20% for validation, and average metrics are reported across folds.

Evaluation Metrics.

- **Macro F1 (primary)** – class-balanced harmonic mean of precision and recall.
- **Accuracy** – overall classification rate.

Threshold Selection. We sweep the decision threshold $t \in [0.10, 0.95]$ (step 0.01 or 0.05) on the validation set of each fold and select the threshold that maximises macro-F1. The resulting per-fold optimum is used during test-time inference.

Hyper-parameter sweep (logged via `W&B`).

- Graph type: static K-NN vs. dynamic PCC
- GNN layers $L \in \{2, 4, 6\}$
- Hidden dimension $d \in \{64, 128, 256\}$
- Dropout $p \in [0.2, 0.5]$
- Learning rate $\eta_0 \in \{5\text{e-}5, 1\text{e-}4\}$

Ensembling. Final prediction is an unweighted mean of the $K = 5$ fold probabilities,

$$p_{\text{ens}} = \tfrac{1}{K} \sum_{k=1}^{K} p^{(k)}.$$

Probability averaging outperformed weighted or majority voting; the ensemble yields a small increase in the accuracy.

IV. Results

We report validation performance for every architectural tier: static–graph GNNs, single-scale GCN+LSTM, and the proposed multi-scale variant. All metrics are macro-F1 unless noted otherwise.

A. Static Spatial GNNs

Table I summarises performance for GCN, GraphSAGE, and GAT using a single 2-s window (node feature dim 210).

TABLE I: Static GNNs on 210-dimensional node features

Model	Grid	K-NN
GCN	0.58	0.66
GraphSAGE	0.60	0.67
GAT	0.63	0.68

K-NN topology outperforms the naïve grid in every case. GAT benefits most from edge-level attention but brings higher compute overhead.

B. Single-Scale GCN+LSTM

Injecting temporal context via an LSTM boosts performance to around 0.78 F1 (2-s windows). Using dynamic PCC graphs per window adds a further +0.04 pp, indicating that time-varying functional connectivity is informative. A moderate drop (around 6–10 pp) on the private Kaggle leaderboard reveals distribution shift between training and hidden test patients.

C. Multi-Scale GCN+LSTM (Ours)

Processing three resolutions (0.5 s (145 dim), 1 s (208 dim), 2 s (210 dim)) in parallel and fusing with cross-scale attention yields the best results:

$$\text{F1}_{\text{val}} = \mathbf{0.82} \quad (\pm 0.5 \text{ pp over 5 CV folds}).$$

On the private leaderboard the multi-scale model ranks in the top.

D. Ablation: Feature Depth and Graph Type

- **Feature richness.** Switching from 9-dim statistics to full spectral–wavelet bags increases F1 by a bit on average across models.
- **Graph choice.** Static K-NN is the most stable; dynamic PCC is superior only when temporal modelling (LSTM) is present.

E. Engineering Challenges

Memory. Three-scale features expand a 12-s segment to $\sim$1.8 MB; we therefore store pre-normalised `.npy` tensors and batch 64 segments per GPU.

I/O. On-the-fly PCC computation was moved to `numba`-accelerated Cython and cached per batch, reducing DataLoader overhead by 47 %.

Imbalance. Weighted BCE and minority oversampling kept seizure recall above 0.78 without inflating false positives.

Most experiments were tracked with `Weights&Biases`; We tried multiple seeds and patient splits to find the best outcome.

Overall, explicit spatial reasoning (GNN) + temporal aggregation (LSTM) + multi-scale features is essential: each component alone is insufficient, but together they push macro-F1 from 0.68 (baseline) to 0.82 (proposed model) on held-out patients.

V. Conclusion

This work evaluated graph-based neural models for automated EEG seizure detection and demonstrated their advantages over classical sequential baselines. In particular, the Multi-Scale GCN+LSTM architecture, leveraging rich temporal features and dynamic functional graphs, achieved the best performance by capturing both spatial and temporal dependencies in EEG data.

We found that model performance depends heavily on the quality of input features and graph topology. Multi-scale feature extraction and dynamic Pearson-based graphs proved especially effective, though at higher computational cost.

Despite these advances, challenges remain in graph definition, scalability, and model interpretability. Future directions include learning graphs end-to-end, refining attention mechanisms for neurophysiological signals, and developing clinically interpretable GNN frameworks.

Graph-based approaches thus offer a robust and extensible foundation for seizure detection and broader EEG analysis tasks.

REFERENCES

[1] Justin A Cover et al. "Seizure detection and prediction by graph neural networks". In: *Epilepsia* 62 (2021), S60–S73.

[2] Rosana Esteller et al. "Line length: An efficient feature for seizure onset detection". In: vol. 2. Feb. 2001, 1707–1710 vol.2. DOI: 10.1109/IEMBS.2001.1020545.

[3] Robert S Fisher et al. "Epileptic seizures and epilepsy: definitions proposed by the International League Against Epilepsy (ILAE) and the International Bureau for Epilepsy (IBE)". In: *Epilepsia* 46.4 (2005), pp. 470–472.

[4] Michael Hüsken and Peter Stagge. "Recurrent neural networks for time series classification". In: *Network: Computation in Neural Systems* 14.2 (2003), pp. 321–337.

[5] Xiaofei Jia et al. "EpilepsyGAN: a generative adversarial network for seizure detection in EEG". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 1514–1524.

[6] Kwanhyung Lee et al. "Real-Time Seizure Detection using EEG: A Comprehensive Comparison of Recent Approaches under a Realistic Setting". In: *Conference on Health, Inference, and Learning*. PMLR. 2022, pp. 116–133.

[7] Donald L Schomer and Fernando H Lopes da Silva. *Niedermeyer's electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2012.

APPENDIX

Our code and trained model weights are available at: https://github.com/JohnTomasTaylor/NetworkMLGroup14.git

TABLE II: Features Extracted From Each Window

Category	**Feature**	**Dim.**	**Description**
Time Domain	Mean (x_{avg})	1	Average amplitude
	Abs. Mean Dev. (x_{arv})	1	Mean of absolute deviation
	Std. Deviation (x_{std})	1	Variability of signal
	Peak-to-Peak (x_{p_p})	1	Maximum - Minimum
	Zero Crossing ($zeros$)	1	Number of sign changes
	Skewness (x_{skew})	1	Asymmetry of distribution
	Kurtosis (x_{kurt})	1	Tail heaviness of distribution
	RMS (x_{rms})	1	Root mean square
	Line Length (ll)	1	Sum of distances between successive points (see [2])
Frequency Domain	PSD	Variable	First min(128, win_len//2+1) bins of the Power Density Spectrum
	Delta (0.5-4 Hz)	1	Mean PSD in delta band
	Theta (4-8 Hz)	1	Mean PSD in theta band
	Alpha (8-13 Hz)	1	Mean PSD in alpha band
	Beta (13-30 Hz)	1	Mean PSD in beta band
	Gamma (30-45 Hz)	1	Mean PSD in gamma band
Wavelet Domain	CWT Energy	63	Energy of each CWT scale (Morlet Wavelets)
	Avg CWT Energy (δ-γ)	5	Band-averaged values computed from the 63-scale CWT energy