Projects for Students June 2025

ALS Dataset:
https://rdr.ucl.ac.uk/articles/dataset/Longitudinal_ALS_EEG_Dataset_for_Motor_Imagery_Studies/28156016

Riemannian Manifold Learning on PLV (Offline/Online) **

Aim:
Test whether PLV-based covariance matrices can be used in a Riemannian classification framework, building on Corsi et al. 2021.

Tasks:

- Replicate Corsi's approach: build covariance matrices from filtered PLV data.
- Explore both offline and online (session-by-session or time-windowed) decoding.
- Evaluate on ALS dataset and potentially simulate real-time feedback.

divide the continuous EEG signal into small segments (or windows), like 1-second chunks, and calculate PLV values separately in each window.

Deliverables:

- Functional implementation of manifold learning with PLV features.
- Accuracy results and generalization analysis (esp. session transfer).
- Optional: exploratory online pipeline for live decoding.

Supervision:
Medium — paper is implementable, but adapting to PLV needs experimentation.

Why It's Useful:
If PLV + manifold learning proves effective, this provides a hybrid approach — geometric interpretability with neural synchrony.

Non-stationarity Autoencoder Testing on ALS Dataset **

Aim:
Replicate and evaluate the denoising autoencoder approach from Ofer et al. 2025 on your ALS dataset to mitigate non-stationarity.

Tasks:

- Implement or reproduce the DDAE model from the paper.
- Train on source session(s), test on unseen sessions (pseudo-online).
- Apply it as a pre-processing module before GAT/CNN classifier.
- Quantify improvements in session transfer accuracy and stability.

Deliverables:

- Model + Feature Stability visualizations.
- Evaluation results with and without DDAE on the same classifier.
- Clean code for integration into your existing pipelines.

Skills:

- PyTorch, Autoencoders, Good coding practice.

Supervision:
Medium — code may not be open, but paper is reproducible with effort.

Why It's Useful:
It's one of the few *deep* approaches to solving the non-stationarity problem and could give real performance boosts across longitudinal data.

时间的

TGN: Temporal Graph Networks for EEG (Offline/Online) - Difficult Project

Aim:
Implement and test Temporal Graph Networks (e.g., Rossi's TGN) for modeling time-varying PLV graphs across motor imagery trials or sessions.

Tasks:

- Preprocess EEG into dynamic graphs: PLV over time windows.
- Implement TGN using PyTorch Geometric Temporal or similar.
- Compare to static GAT in cross-session decoding.
- Optional: explore memory modules (GRU, attention) for adaptation.

Deliverables:

- Temporal Graph Network implementation for MI EEG.
- Accuracy vs. static GAT comparison.

- Performance across folds/sessions.

Skills:

- PyTorch, PyTorch Geometric (Temporal).
- Understanding of dynamic graphs and memory networks.

Supervision:
Medium-High (temporal GNNs are tricky to debug but impactful).

Why It's Useful:
This is your **cutting-edge direction** — few have done dynamic GNNs on EEG. A well-done TGN could be the centerpiece of a future paper.

Proof-of-Concept Online Dataset Collection & Testing

Aim:
Work with you to collect a clean multi-session dataset using the new EEG headsets and test a full online decoding + feedback loop.

Tasks:

- Assist with headset setup, calibration, and recording.
- Design controlled tasks (MI, rest, eyes-open/closed, etc.).
- Preprocess in real-time or near-real-time.
- Run model inference and test accuracy drift across sessions.

Deliverables:

- A small but well-documented BCI dataset.
- Annotated logs + metadata.
- First reports on live system accuracy, latency, and robustness.

Skills:

- Comfortable with EEG hardware, scripting, and data collection.

Grosse-Wentrup Dataset: Resting State ALS Graph Biomarkers

Aim:
Analyse functional connectivity in longitudinal resting-state EEG from ALS patients to discover early biomarkers of degeneration.

Tasks:

- Load and process Grosse-Wentrup's 2-year dataset.
- Compute graph metrics (entropy, strength, modularity) per session.
- Correlate features with ALSFRS or other clinical scores.
- Visualize evolving networks and trends.

Deliverables:

- Graph analysis codebase (based on `networkx`, `bctpy`, or `nilearn`).
- Figures showing metric evolution and patient progression.
- Exploratory insights or hypotheses for degeneration mechanisms.

Skills:

- Graph theory, EEG preprocessing.
- Data visualization, exploratory stats.

Why It's Useful:
Excellent publication material, helps with grant positioning, and bridges neuroscience with your ML pipeline.