9. Conference X - Temporal Graph Network for Adaptive Motor Imagery Classification in ALS

Target Conference:

- EMBC
- NER
- SMC
- Neural Networks

Inductive Graph Classification Task:
Can predict on unseen nodes.

Brief Overview:

Volitional modulations in EEG due to motor imagery can be used as control signals for various end-user needs. However, traditional decoding methods struggle with the non-stationarity of EEG signals, which can arise from factors like ALS disease progression, leading to changes in cortical and motor neuron activity.

Our past work has focused on modeling EEG signals as a graph at any point in time, giving us a new way to combine spatial and temporal features with additional measures like phase. This approach allows us to search for more robust patterns across various representations of the data.

Currently, we aim to explore temporal adaptive learning using **Temporal Graph Learning (TGN)**. TGNs can capture long-term dependencies for each node in a graph. When a new node is introduced, its memory is initialized as a zero vector and updated after every event involving the node—even after the model is fully trained.

Hypothesis:
Introducing temporal module to understand graph adaptation can enhance the model's ability to capture long-term dependencies in changing EEG data. If this hypothesis holds, we should observe improving classification accuracies over time as the model adapts to signal changes.

Relevant Papers:

- Temporal Graph Networks (TGN):
  TGN Paper
  TGN on Papers with Code

Project Tasks:
The selected candidate will focus on developing Graph Neural Network architectures, specifically a TGN. While we have an existing pipeline that produces graphs from our datasets, the student has the flexibility to adapt or rewrite the process as needed.

Primary Tasks:

1. Integrate a suitable **TGN architecture** into the existing pipeline and evaluate its performance.

Deliverables:

1. TGN model with results (classification accuracies).
2. *Aiming* for an improvement in Classification Accuracies from our adaptive CSP work as a causal system, and improvement from our GAT method evaluated with Cross Validation.

Timeline

1. Week 1: Literature : Overview and Foundations
2. Week 2: Literature : Graph Learning
3. Week 3-5: Implementation

My Github

https://github.com/rishannp/Motor-Imagery---Graph-Attention-Network/tree/main

Notes

Links

https://arxiv.org/pdf/2006.10637
https://github.com/pyg-team/pytorch_geometric/blob/master/examples/tgn.py#L130
https://github.com/twitter-research/tgn/blob/master/modules/memory.py
https://pytorch-

geometric.readthedocs.io/en/2.6.1/_modules/torch_geometric/nn/models/tgn.html
https://pytorch-geometric.readthedocs.io/en/2.5.2/modules/loader.html#torch_geometric.loader.TemporalDataLoader
https://pytorch-geometric.readthedocs.io/en/2.5.2/_modules/torch_geometric/loader/temporal_dataloader.html#TemporalDataLoader
https://pytorch-geometric.readthedocs.io/en/2.5.2/generated/torch_geometric.data.TemporalData.html#torch_geometric.data.TemporalData

https://pytorch-geometric.readthedocs.io/en/2.5.2/generated/torch_geometric.data.TemporalData.html#torch_geometric.data.TemporalData

import TemporalData from torch_geometric.data

TemporalData(src,dst,dst,t,msg)
src = list of source nodes
dst = destination nodes
t = timestsamp
msg = messages feature matrix with shape [events,msg_features]