Bilevel Optimization for SEEG based Seizure Detection

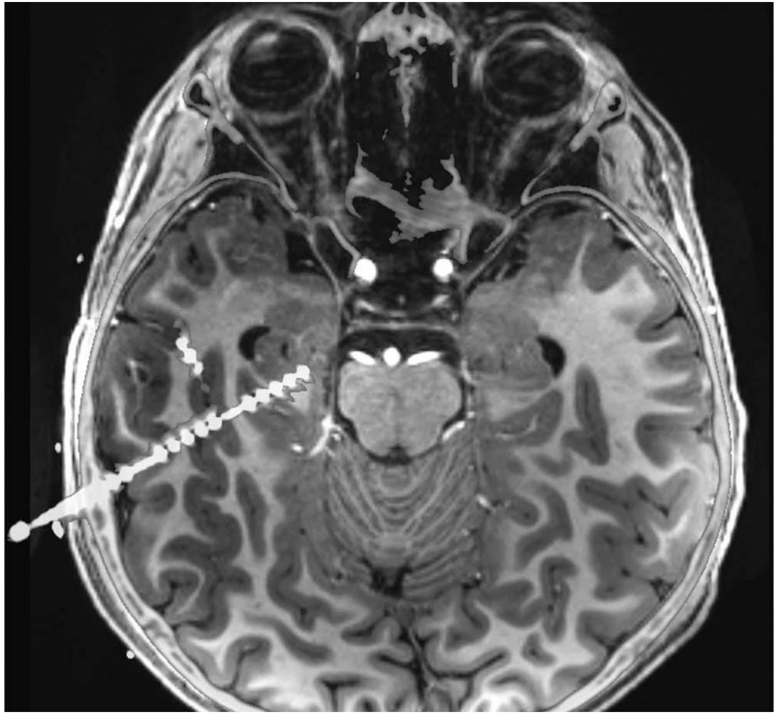
A F M Saif, Sombuddha Chatterjee
ECSE, RPI

Topics Overview

- Motivation and Problem
- Data Description
- Seizure detection task
- Overview of reference methodology
- Problem formulation
- Algorithm
- Model Architecture
- Experimental Results
- Ablation Study
- Conclusion

Motivation & Problem

- Epilepsy affects over 50 million people globally, with many having drug-resistant epilepsy.
- SEEG is used to localize the Seizure Onset Zone (SOZ) for surgical intervention.
- Existing seizure detection models are patient-specific and not generalizable.
- Challenge: Developing a patient-independent seizure detection model that handles variability across patients.

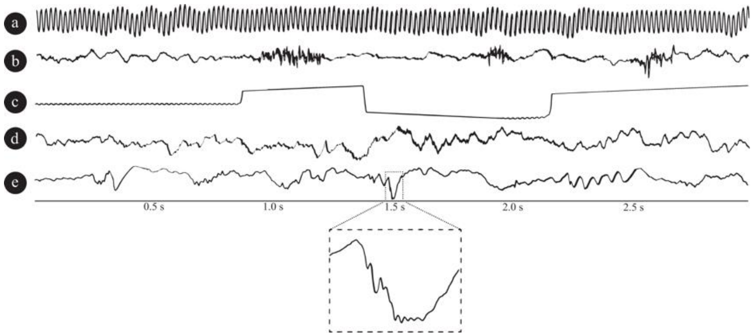


An MRI of the brain showing Stereo EEG implant

Data Description – Public dataset from Mayo Clinic

- 3-second clips categorized into four distinct events:
  - **Physiological SEEG**: During wakefulness and sleep.
  - **Pathophysiological activity**: Includes epileptiform features like spikes or high-frequency oscillations.
  - **Artifacts**: Muscle, movement, and machine-induced.
  - **Powerline noise** (50/60 Hz, dependent on clinic's powerline frequency).
- Data collected from 23 different patients.
- Each 3 second clip is sampled at 5 kHz and labelled independently by 3 different reviewers.

Classification category	Mayo Clinic
Physiological Activity	56730
Pathological Activity	15227
Artifacts	41303
Power line noise (50Hz/60Hz)	41922
Total	**155182**



- **Data Representation:**
  - SEEG recording is a multivariate time series: $\mathbf{T} \in \mathbb{R}^{N \times C}$
  - N : Length of series, C : Number of channels.
  - Single Channel: $\mathbf{x}_\mathrm{c} = (x_1, x_2, \dots, x_N)$
- **Data Segmentation:**
  - Contiguous data is divided into segments:

$$\mathcal{S}c = \{s_{c,0}, s_{c,1} + \cdots, s_{c,K-1}\}$$

  - where $\boldsymbol{s}_{c,k} = \{x_{l \times k+1}, \dots, x_{l \times (k+1)}\}$ is the k-th segment data on channel c from $\mathbf{T}$ (l is the length of each segment, $K = \lfloor N/l \rfloor$ is the total number of segments on channel c)
- **Labels:**

$$\mathbb{Y}_c = \{y_{c,0}, y_{c,1}, \dots, y_{c,K-1}\},$$

where $y_{c,k} \in \{0,1\}$ is the label of $s_{c,k}$, which indicates whether the segment contains a seizure event $(y_{c,k} = 1)$ or not $(y_{c,k} = 0)$.

Overview of the Reference Methodology (Yuan et al, NIPS 2024)

- Challenges:
  - Variability in seizure patterns across patients
  - Aligning seizure and non-seizure data distributions across patients.
  - Learning representations robust to domain shifts.

- Self-Supervised Learning (SSL):
  - Channel Discrimination: $\boldsymbol{h}_{cd}^{m_1} = \text{abs}\left(\boldsymbol{h}_{c_1,k_1}^{m_1} - \boldsymbol{h}_{c_2,k_2}^{m_1}\right)$
  - $\boldsymbol{h}_{c_1,k_1}^{m_1}, \boldsymbol{h}_{c_2,k_2}^{m_1}$ are the feature vector encodings of two sequences $(\boldsymbol{u}_{c_1,k_1}^{m_1}, \boldsymbol{u}_{c_2,k_2}^{m_1})$ sampled from different channels with equal probability. Binary CE loss applied: $\mathcal{L}_{cd}$
  - A decoder is used to reconstruct the original sequences $(\hat{\boldsymbol{u}}_{c_1,k_1}^{m_1}, \hat{\boldsymbol{u}}_{c_2,k_2}^{m_1})$
  - Reconstruction loss measured as: $\mathcal{L}_{rec} = \sum_{m_1=1}^{M_1}\left(\left\|\boldsymbol{u}_{c_1,k_1}^{m_1} - \hat{\boldsymbol{u}}_{c_1,k_1}^{m_1}\right\|^2 + \left\|\boldsymbol{u}_{c_2,k_2}^{m_1} - \hat{\boldsymbol{u}}_{c_2,k_2}^{m_1}\right\|^2\right)$.
  - Context Swapping: Context of the selected sequence was swapped with variable probability and a binary CE loss applied as $\mathcal{L}_{cs}$.

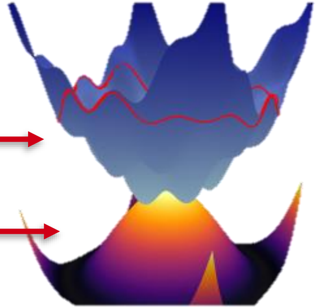
- SSL Loss: $\mathcal{L}_{ssl} = \mathcal{L}_{rec} + \mathcal{L}_{cd} + \mathcal{L}_{cs}$

Problem Formulation

Bilevel optimization can be presented as

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{\text{val}}(\theta, \phi^*(\theta); \mathcal{D}_{\text{val}}) + \lambda \mathcal{L}_{\text{pretrain}}(\theta; \mathcal{D}_{\text{pretrain}})$$

s.t. $$\phi^*(\theta) = \arg\min_{\phi} \mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}}).$$



Here,

- Backbone parameters: θ
- Classification head parameters: ϕ

Supervised loss

Supervised loss used here is the cross-entropy loss:

$$\mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \hat{y}_{i,c}$$

Here,

- Number of samples: N
- Number of classes: C
- True label: $y_{i,c}$
- Predicted label: $\hat{y}_{i,c}$

Channel discrimination loss:

$$\mathcal{L}_{\mathrm{cd}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{distance}(z_{i,c_1}, z_{i,c_2})$$

Here,

- Embedding of different channel: $z_{\{i,c1\}}, z_{\{i,c2\}}$

Context-swapping loss:

$$\mathcal{L}_{\mathrm{cs}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

Combined pretraining loss:

$$\mathcal{L}_{\mathrm{pretrain}}(\theta) = \mathcal{L}_{\mathrm{cd}}(\theta) + \mathcal{L}_{\mathrm{cs}}(\theta)$$

Lower-level Problem: Supervised-Training

Lower-level objective is to learn ϕ:

$$\phi^*(\theta) = \arg\min_{\phi} \mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}})$$

Here,

- Supervised training loss: L_{train}
- Fine-tuned parameters: $\phi^*(\theta)$

Upper-level objective is to learn θ:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{\mathrm{val}}(\theta, \phi^*(\theta); \mathcal{D}_{\mathrm{val}}) + \lambda \mathcal{L}_{\mathrm{pretrain}}(\theta; \mathcal{D}_{\mathrm{pretrain}})$$

Here,

- Validation loss: L_{val}
- Pre-training loss:$L_{pretrain}$

Penalty-based bilevel optimization algorithm is used:

$$\min_{\theta,\phi} f(\theta,\phi) = \min_{\theta,\phi} \Bigg[(1-\gamma)\Big(\mathcal{L}_{\text{val}}(\theta,\phi^*(\theta);\mathcal{D}_{\text{val}}) + \lambda\mathcal{L}_{\text{pretrain}}(\theta;\mathcal{D}_{\text{pretrain}})\Big) \underbrace{+\,\gamma\Big(\mathcal{L}_{\text{train}}(\theta,\phi;\mathcal{D}_{\text{train}}) - \min_{\phi^*}\mathcal{L}_{\text{train}}(\theta,\phi^*;\mathcal{D}_{\text{train}})\Big)\Bigg]}_{p(\theta,\phi)}$$

Here,

- Penalty constant: γ

Step 1: Computing $\nabla_\phi f(\theta, \phi)$:

$$\nabla_\phi f(\theta, \phi) = \nabla_\phi \gamma \mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}})$$

Parameter update:

$$\phi^{(k+1)} = \phi^{(k)} - \alpha_\phi \nabla_\phi f(\theta, \phi)$$

Here,

- Learning rate: α_ϕ

Step 2: Computing $\nabla_\theta f(\theta, \phi)$:

$$\nabla_\theta f(\theta,\phi) = \nabla_\theta \Bigg((1-\gamma)\Big(\mathcal{L}_{\text{val}}(\theta, \phi^*(\theta); \mathcal{D}_{\text{val}}) + \lambda \mathcal{L}_{\text{pretrain}}(\theta; \mathcal{D}_{\text{pretrain}})\Big)$$

$$+ \gamma\Big(\mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}}) + \mathcal{L}_{\text{train}}(\theta, \hat{\phi}; \mathcal{D}_{\text{train}})\Big)\Bigg)$$

Here:

$$\hat{\phi} \approx \phi^*(\theta) := \arg\min_{\phi} \mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}})$$

Parameter update:

$$\theta^{(k+1)} = \theta^{(k)} - \alpha_\theta \nabla_\theta f(\theta, \phi)$$

Full Algorithm

Algorithm 1 Penalty-Based Bilevel Optimization Algorithm

Initialize parameters θ and ϕ.
while not converged **do**
    Compute $\nabla_\phi f(\theta, \phi)$.
    Update ϕ using:

$$\phi^{(k+1)} = \phi^{(k)} - \alpha_\phi \nabla_\phi f(\theta, \phi).$$

    Compute $\nabla_\theta f(\theta, \phi)$.
    Update θ using:

$$\theta^{(k+1)} = \theta^{(k)} - \alpha_\theta \nabla_\theta f(\theta, \phi).$$

    Here,

$$\nabla_\theta \min_\phi f(\theta, \phi) \approx \nabla_\theta f(\theta, \hat{\phi})$$

    where,

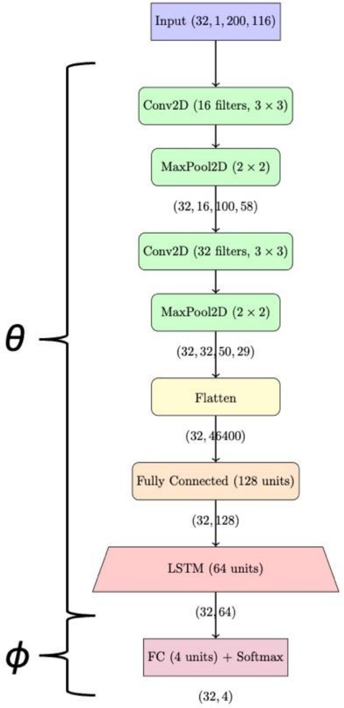
$$\hat{\phi} \approx \phi^*(\theta) := \arg\min_\phi \mathcal{L}_{\text{train}}(\theta, \phi; \mathcal{D}_{\text{train}})$$

end while

Model Architecture and Hyperparameter set

- Input Data: EEG signals shaped as **(32, 1, 200, 116)** (batch size: 32, 1 channel, 200 time steps, 116 spatial channels).
- Feature Extraction (CNN Encoder):
  - **Layer 1**: 16 filters, kernel: 3x3, max-pooling: 2x2 → Output: (32, 16, 100, 58).
  - **Layer 2**: 32 filters, kernel: 3x3, max-pooling: 2x2 → Output: (32, 32, 50, 29).
  - **Fully Connected Layer**: Flattens to size **46,400**, reduces to **128-dimensional representation**.
- LSTM:
  - Single-layer LSTM with **64 hidden units** for temporal dependencies.
  - **Output Layer**: Fully connected, 4 classes (softmax activation).

Component	Details
Pretraining Tasks	Channel discrimination, Context swapping
Epochs	50
Upper-level learning rate	10^(-3)
Lower-level learning rate	10^(-4)
Penalty constant increase rate (gamma)	0.002
Batch Size	16
Loss Function	Cross-entropy loss
Optimizer	Adam optimizer



Model architecture

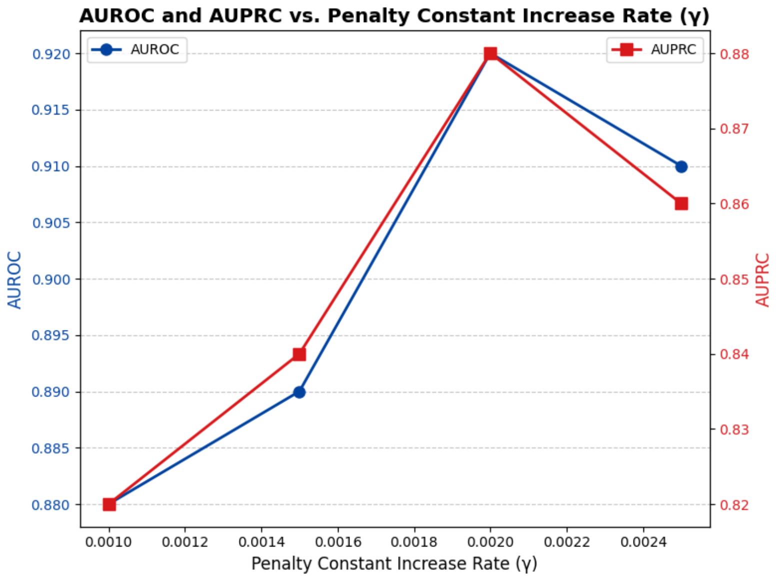
Experimental Results Summary

Method	AUROC	AUPRC
Base Model (Random Initialization)	0.84	0.79
PT+FT (Yuan et al. (2024))	0.89	0.85
Bilevel Optimization (Our Method)	0.92	0.88

- **AUROC (Area Under the ROC Curve)**: evaluates the model's ability to distinguish between classes by plotting the True Positive Rate (sensitivity) against the False Positive Rate (1-specificity) across thresholds.
  - Measures the model's ability to differentiate seizure vs. non-seizure events across all thresholds.
- **AUPRC (Area Under the Precision-Recall Curve)**: focuses on the positive (seizure) class by plotting Precision (positive predictive value) against Recall (sensitivity).
  - Focuses on identifying seizures in imbalanced datasets by balancing recall (sensitivity) and precision (positive predictive value).
- The result shows an improvement of **3.7%** in AUROC and **3.4%** in AUPRC.

Ablation Study

- Low γ: The performance is moderate.
- Moderate γ: The performance improved.
- High γ: The performance declined.



Conclusion

- Bilevel optimization enhances seizure detection performance.
- Effectively addresses the domain shift challenge across patients.
- Successfully tackles variations in brain region characteristics.
- Regularization improves the model’s robustness and reliability.

Thank You

RPI 200