

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ - VIỄN THÔNG**

**BÁO CÁO**  
**TIẾN ĐỘ PBL4**

**ĐỀ TÀI:**

**IMLE-NET: AN INTERPRETABLE MULTI-  
LEVEL MULTI-CHANNEL MODEL FOR ECG  
CLASSIFICATION**

Sinh viên thực hiện: **TRẦN DUY NGUYÊN**

Mã số sinh viên: **106220264**

Lớp: **22KTMT2**

**Đà Nẵng, 01/2026**

## I. GIỚI THIỆU

Bệnh tim mạch là nguyên nhân gây tử vong hàng đầu trên toàn cầu. Theo Tổ chức Y tế Thế giới (WHO), ước tính có khoảng 17,9 triệu người tử vong vì bệnh tim mạch vào năm 2016, chiếm khoảng 31% tổng số ca tử vong toàn cầu. Phát hiện sớm bệnh tim là rất quan trọng để điều trị hiệu quả và giảm tỷ lệ tử vong.

Điện tâm đồ (ECG) là một thủ thuật đơn giản, không xâm lấn và chi phí thấp giúp chúng ta hiểu được hoạt động của tim, từ đó hỗ trợ chẩn đoán các bệnh tim. Một bản ghi ECG chứa thông tin về hoạt động điện của tim, dựa vào đó bác sĩ tim mạch có thể xác định các bất thường trong hoạt động của tim để chẩn đoán nhiều tình trạng bệnh lý khác nhau.

Tuy nhiên, quá trình phân tích bản ghi ECG tốn thời gian và đòi hỏi sự chú ý của chuyên gia được đào tạo. Hơn nữa, nó dễ xảy ra lỗi do con người. Chẩn đoán sai các dấu hiệu sớm của bệnh tim từ bản ghi ECG vẫn là một mối quan tâm lớn. Do đó, nhiều nghiên cứu đã được thực hiện để tự động hóa việc phân loại tín hiệu ECG nhằm hỗ trợ chẩn đoán.

Hầu hết các phương pháp hiện có sử dụng học máy truyền thống hoặc học sâu để phát hiện bệnh tim từ bản ghi ECG của bệnh nhân. Tuy nhiên, phần lớn các nghiên cứu tập trung vào mô hình xử lý tín hiệu ECG đơn kênh. Trong khi đó, từ góc độ lâm sàng, bác sĩ tim mạch đưa ra chẩn đoán dựa trên bản ghi ECG 12 kênh tiêu chuẩn.

So với bản ghi đơn kênh, bản ghi 12 kênh tái tạo chính xác hơn các đặc trưng của ECG như phức bộ QRS, đoạn ST và sóng T. Ví dụ, sự nâng cao đoạn ST - một đặc điểm quan trọng để chẩn đoán nhồi máu cơ tim (MI) - được xác định tốt nhất bằng bản ghi 12 kênh và việc xác định vị trí tổn thương cơ tim không thể thực hiện nếu không có bản ghi 12 kênh.

Để tài tập trung vào việc nghiên cứu và triển khai mô hình IMLE-Net (Interpretable Multi-level Multi-channel Model) để phân loại tín hiệu ECG 12 kênh. Mô hình này có những ưu điểm nổi bật:

- Tận dụng thông tin đa kênh từ bản ghi ECG 12 kênh tiêu chuẩn.
- Học các mẫu ở ba cấp độ: cấp độ nhịp (beat level), cấp độ nhịp điệu (rhythm level) và cấp độ kênh (channel level).
- Sử dụng cơ chế chú ý (attention mechanism) đa cấp để tăng tính giải thích (interpretability) của mô hình.
- Đạt hiệu suất vượt trội so với các mô hình hiện có trên bộ dữ liệu PTB-XL.

## II. PHƯƠNG PHÁP THỰC HIỆN

### 1. Tìm hiểu bộ dữ liệu PTB-XL

Bộ dữ liệu PTB-XL là cơ sở dữ liệu điện tâm đồ lâm sàng quy mô lớn, công khai và cung cấp nền tảng vững chắc cho việc huấn luyện mô hình IMLE-Net.

#### 1.1. Quy mô và cấu trúc dữ liệu

- Quy mô: Bao gồm 21.837 bản ghi ECG từ 18.885 bệnh nhân.

- Cân bằng giới tính: 52% nam và 48% nữ.
- Độ tuổi: Rộng từ 0 đến 95 tuổi.
- Cấu trúc tín hiệu: Mỗi bản ghi kéo dài 10 giây, thu thập từ 12 đạo trình tiêu chuẩn (I, II, III, aVL, aVR, aVF, V1-V6).
- Tần số lấy mẫu: Ghi ở 400 Hz, được hạ xuống 100 Hz để tối ưu tính toán.

## 1.2. Phân loại chẩn đoán

Các bản ghi được phân loại theo 5 nhóm chính (superclasses):

- NORM: Điện tâm đồ bình thường.
- CD: Rối loạn dẫn truyền (Conduction Disturbance).
- MI: Nhồi máu cơ tim (Myocardial Infarction).
- HYP: Phì đại (Hypertrophy).
- STTC: Thay đổi đoạn ST/T (ST/T Changes).

Ngoài ra, dữ liệu còn hỗ trợ phân tích các nhãn con (subclasses) như Anteroseptal MI (ASMI) và Inferior MI (IMI) để đánh giá khả năng chẩn đoán chi tiết.

## 2. Tiên xử lý dữ liệu

### 2.1. Hạ tần số lấy mẫu (Downsampling)

Tín hiệu gốc từ bộ dữ liệu PTB-XL được ghi lại ở tần số 400 Hz. Để tối ưu hóa hiệu năng tính toán mà vẫn bảo toàn các đặc trưng hình thái (morphology), dữ liệu được hạ xuống còn 100 Hz. Sau khi hạ tần số, mỗi bản ghi 10 giây tương đương với 1000 điểm dữ liệu thời gian trên mỗi đạo trình.

### 2.2. Phân đoạn nhịp tim (Sliding Window Segmentation)

Thay vì dựa vào phát hiện đỉnh R (R-peak detection) vốn dễ sai sót khi gấp tín hiệu nhiều, mô hình sử dụng phương pháp cửa sổ trượt không chồng lấp (sliding window with no overlap):

- Để có được phân đoạn nhịp thứ k, cửa sổ sẽ trải dài từ  $(k-1) \times W$  đến  $k \times W$  trên toàn bộ tín hiệu.
- Với tổng chiều dài  $T = 1000$  và kích thước cửa sổ  $W = 50$ , mỗi đạo trình được chia thành  $N = T/W = 20$  phân đoạn nhịp tim cố định.

### 2.3. Chuẩn hóa và cấu trúc hóa đầu vào

Chuẩn hóa toàn cục: Thay vì chuẩn hóa từng kênh đơn lẻ, mô hình thực hiện chuẩn hóa dựa trên giá trị trung bình ( $\mu$ ) và độ lệch chuẩn ( $\sigma$ ) tính toán trên toàn bộ dữ liệu huấn luyện:

$$x_{norm} = (x - \mu) / \sigma$$

Tensor đầu vào được cấu trúc lại dưới dạng tensor 3 chiều kích thước  $12 \times 1000 \times 1$ , tương ứng với 12 đạo trình, mỗi đạo trình chứa 1000 điểm tín hiệu.

## 3. Kiến trúc mô hình IML-E-Net

Mô hình IMLE-Net được xây dựng dựa trên kiến trúc học sâu phân cấp, kết hợp giữa mạng tích chập (CNN), mạng bộ nhớ dài ngắn hai chiều (Bi-LSTM) và cơ chế chú ý đa tầng (Multi-level Attention).

Điểm đặc biệt của IMLE-Net là xử lý đa kênh song song: mỗi trong 12 kênh ECG (I, II, III, aVL, aVR, aVF, V1-V6) được xử lý độc lập qua cùng một bộ CNN-Attention-BiLSTM (với trọng số chia sẻ), sau đó tổng hợp thông tin từ 12 kênh này qua Channel Level Attention để đưa ra dự đoán cuối cùng.

Kiến trúc có thể chia thành hai phần chính:

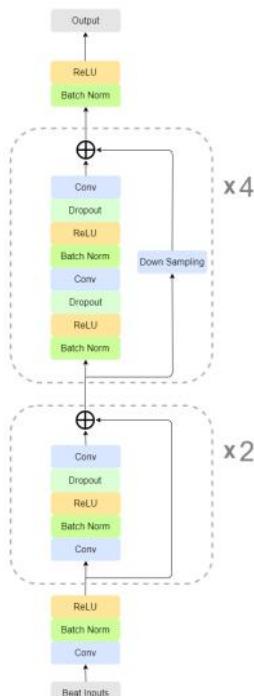
- Phần 1 (Xử lý từng kênh): Mỗi kênh ECG đi qua Beat Level Block → Rhythm Level Block để tạo ra Channel Encoding (vector  $R^c$ ). Phần này được lặp lại cho cả 12 kênh (Channel 1, Channel 2, ..., Channel M với M=12).
- Phần 2 (Tổng hợp đa kênh): Channel Level Attention Block nhận 12 channel encodings ( $R^1, R^2, \dots, R^{12}$ ), áp dụng attention để gán trọng số cho từng kênh, rồi tổng hợp thành một context vector C duy nhất, cuối cùng đưa qua Dense layer để phân loại.

### 3.1. Khối cấp độ nhịp (Beat Level Block)

Khối này xử lý từng phân đoạn nhịp tim đơn lẻ, bao gồm hai thành phần chính:

#### a) Mạng tích chập (CNN) với Residual Blocks

Lấy cảm hứng từ kiến trúc ResNet, CNN sử dụng tổng cộng 6 residual blocks với cấu trúc như sau:



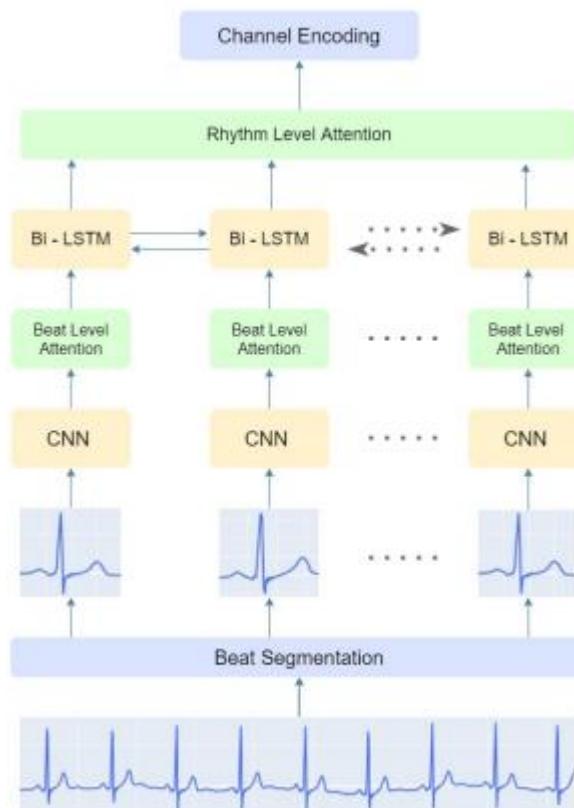
Hình 1: Cấu trúc chi tiết CNN với Residual Blocks

## Cấu trúc của một Residual Block:

1. Lớp Conv1D đầu tiên → Dropout → ReLU → Batch Normalization
2. Lớp Conv1D thứ hai → Dropout → ReLU → Batch Normalization
3. Skip Connection ( $\oplus$ ): Đầu vào gốc được cộng trực tiếp với đầu ra (identity mapping hoặc qua Conv nếu cần thay đổi kích thước)

## Cấu hình 6 residual blocks:

- 2 blocks đầu tiên ( $\times 2$ ): Không downsampling, giữ nguyên kích thước, 32 filters
- 4 blocks tiếp theo ( $\times 4$ ): Downsampling với hệ số 2 sau mỗi 2 blocks, số filters tăng gấp đôi ( $32 \rightarrow 64 \rightarrow 128$ )
- Tất cả lớp Conv1D có kernel size = 8
- Dropout rate = 0.5 để chống overfitting
- Trọng số CNN được chia sẻ (shared weights) cho tất cả 20 beat segments và cả 12 channels
- 32 bộ lọc ban đầu; số lượng bộ lọc tăng gấp đôi sau mỗi downsampling.



Hình 2: Sơ đồ chi tiết kiến trúc CNN ở cấp độ nhịp

Phép toán tích chập 1D cho một kênh đầu ra  $j$  tại vị trí  $i$  được tính bằng công thức:

$$y_{i,j} = \sigma \left( \sum (W_{k,j} \cdot X_{i+k-1}) + b_j \right)$$

Trong đó:  $x$  là tín hiệu đầu vào,  $w$  là trọng số của bộ lọc (kernel),  $b$  là độ chệch (bias),  $\sigma$  là hàm kích hoạt ReLU.

### b) Cơ chế chú ý cấp độ nhịp (Beat Attention)

Beat Attention là một mạng nơ-ron 2 lớp giúp xác định các vùng quan trọng trong một phân đoạn nhịp tim.

### Cơ chế hoạt động:

1. Đầu vào: Output từ CNN là  $b_k^c$  với kích thước  $[W \times d]$  ( $W$  là sequence length sau CNN,  $d$  là số filters)
2. Lớp 1: Linear transformation  $W_b \cdot b_k^c + b_b$ , sau đó qua hàm tanh để tạo non-linearity
3. Lớp 2: Linear transformation  $V_b$  để chiếu về không gian attention scores
4. Softmax: Chuẩn hóa scores thành phân phối xác suất  $\alpha_k^c$  (tổng = 1)
5. Weighted sum: Tổng có trọng số  $B_k^c = \sum(\alpha_{k,i}^c \cdot b_{k,i}^c)$  tạo ra beat context vector

Không phải tất cả các vùng trong một phân đoạn nhịp đều quan trọng như nhau. Cơ chế chú ý giúp xác định các vùng quan trọng hơn bằng cách gán điểm chú ý cao hơn cho:

- Phức bộ QRS: Phản ánh hoạt động khứ cực của tâm thất
- Sóng P: Phản ánh hoạt động khứ cực của tâm nhĩ
- Sóng T: Phản ánh hoạt động tái cực của tâm thất
- Đoạn ST: Quan trọng để phát hiện thiếu máu cục bộ cơ tim

$$\begin{aligned}\alpha_k^c &= \text{softmax}(V_b \cdot \tanh(W_b \cdot b_k^c + b_b)) \\ B_k^c &= \sum(\alpha_{k,i}^c \cdot b_{k,i}^c)\end{aligned}$$

Trong đó:  $V_b$ ,  $W_b$ ,  $b_b$  là các tham số attention được học;  $\alpha_k^c$  là điểm chú ý;  $B_k^c$  là vector ngữ cảnh nhịp (beat context vector).

## 3.2. Khối cấp độ nhịp điệu (Rhythm Level Block)

Khối này xử lý chuỗi nhịp điệu ECG được tạo từ nhiều phân đoạn nhịp tim, bao gồm:

### a) Bi-directional LSTM

Bi-LSTM được sử dụng với 128 đơn vị đầu ra để nắm bắt các phụ thuộc xa (long-term dependencies) trong tín hiệu nhịp điệu. Lớp này cho phép mạng phân tích thông tin từ cả hai chiều (quá khứ và tương lai) của chuỗi tín hiệu ECG.

- Đầu vào: Chuỗi các beat context vectors  $B_k^c$  với  $N = 20$  phân đoạn nhịp.
- Đầu ra: Trạng thái ẩn forward và backward được nối (concatenate) để tạo  $r_k^c$ .

### b) Cơ chế chú ý cấp độ nhịp điệu (Rhythm Attention)

Rhythm Attention hoạt động tương tự Beat Attention nhưng ở cấp độ cao hơn - xác định các phân đoạn nhịp quan trọng trong toàn bộ chuỗi nhịp điệu 10 giây.

### Quy trình xử lý:

1. Đầu vào: Chuỗi 20 beat context vectors  $(B_1^c, B_2^c, \dots, B_{20}^c)$  từ Beat Level
2. Bi-LSTM xử lý chuỗi theo cả 2 chiều (forward và backward) để nắm bắt context
3. Output từ Bi-LSTM: Chuỗi  $r^c = (r_1^c, r_2^c, \dots, r_{20}^c)$  với mỗi  $r_i^c$  là concatenation của hidden states từ 2 chiều

4. Rhythm Attention: Tính  $\beta^c = \text{softmax}(V_r \cdot \tanh(W_r \cdot r^c + b_r))$  để xác định beat nào quan trọng
5. Channel Encoding:  $R^c = \sum(\beta_i^c \cdot r_i^c)$  - tổng có trọng số tạo ra vector đại diện cho toàn bộ kênh c

Ý nghĩa: Không phải tất cả 20 beat segments đều mang thông tin chẩn đoán như nhau. Một số beat có thể chứa bất thường rõ ràng (ví dụ: nhịp ngoại tâm thu, block dẫn truyền) trong khi các beat khác hoàn toàn bình thường. Rhythm Attention giúp mô hình 'chú ý' vào những beat bất thường này.

$$\beta^c = \text{softmax}(V_r \cdot \tanh(W_r \cdot r^c + b_r))$$

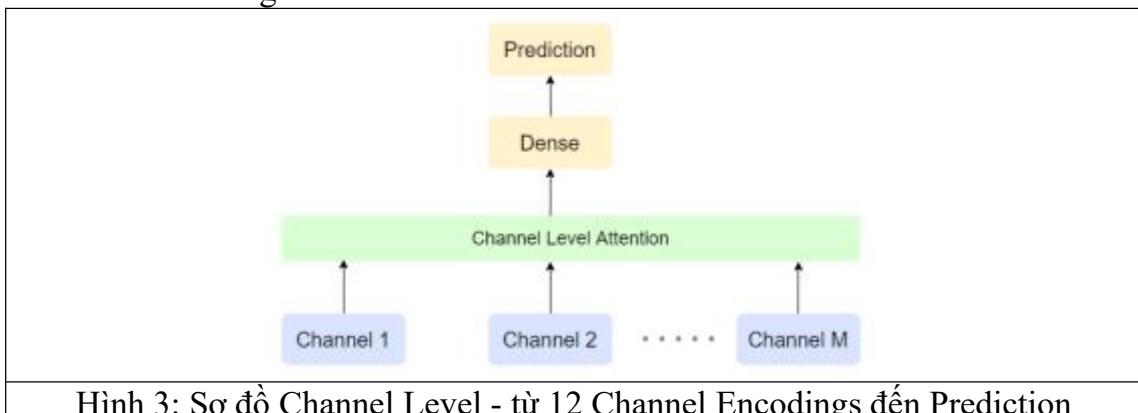
$$R^c = \sum(\beta_i^c \cdot r_i^c)$$

Trong đó:  $V_r, W_r, b_r$  là tham số rhythm attention;  $\beta^c$  là điểm chú ý;  $R^c$  là rhythm context vector (mã hóa kênh - Channel Encoding).

### 3.3. Khối cấp độ kênh (Channel Level Block)

Đây là giai đoạn cuối cùng, tổng hợp thông tin từ 12 kênh ECG độc lập.

- $M = 12$  (tổng số kênh trong ECG chuẩn)
- Channel 1 = Lead I, Channel 2 = Lead II, ..., Channel 12 = Lead V6
- Mỗi channel đã được xử lý độc lập qua Beat Level + Rhythm Level để tạo ra Channel Encoding  $R^c$



Thông tin đặc trưng cho từng bệnh lý thường chỉ xuất hiện rõ ràng trên một số đao trình nhất định:

- Inferior MI (nhồi máu vùng dưới): Dấu hiệu chủ yếu ở leads II, III, aVF
- Anteroseptal MI (nhồi máu vùng trước vách): Dấu hiệu rõ ở leads V1, V2, V3, V4
- Left Bundle Branch Block: Thường thấy rõ ở leads I, aVL, V5, V6

Channel Level Attention học cách tự động gán trọng số cao hơn cho các kênh chứa thông tin chẩn đoán quan trọng:

Điểm chú ý cấp độ kênh được tính toán:

$$\gamma = \text{softmax}(V_c - \tanh(W_c \cdot R^c + b_c))$$

$$C = \sum (\gamma_i \cdot R_i^c)$$

Trong đó:  $V_c, W_c, b_c$  là tham số channel attention;  $\gamma$  là điểm chú ý kênh; C là context vector mã hóa toàn bộ tín hiệu ECG từ tất cả các kênh.

### 3.4. Lớp phân loại (Classification Layer)

Context vector C được truyền qua một lớp mạng nơ-ron tuyến tính (Dense/Fully Connected) để đưa ra dự đoán cuối cùng cho 5 nhóm bệnh lý.

Hàm mất mát sử dụng Binary Cross-entropy để xử lý bài toán phân loại đa nhãn:

$$L = -\frac{1}{N} \sum [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

## 4. Cấu hình huấn luyện

Mô hình được huấn luyện với các thông số sau:

- Optimizer: Adam với momentum  $\beta_1 = 0.9$  và  $\beta_2 = 0.99$
- Learning rate: Ban đầu 0.001, giảm 10 lần sau mỗi 10 epochs nếu không cải thiện
- Batch size: 32
- Max epochs: 60 (với early stopping)
- Window size (W): 50 time-points
- Dropout rate: 0.5
- L2 regularization: Hệ số  $2 \times 10^{-5}$
- Tổng tham số mô hình: 748,517 tham số

Dữ liệu được chia theo 10 folds có sẵn trong PTB-XL: folds 1-8 cho training, fold 9 cho validation, fold 10 cho testing.

## III. KẾT QUẢ THỰC HIỆN

### 1. Tiến độ huấn luyện

Đến nay, mô hình đã được huấn luyện được 25 epochs trên bộ dữ liệu PTB-XL. Các chỉ số đạt được sau 25 epochs:

Metric	Training	Validation	Loss
Accuracy	0.7982	0.6865	Train: 0.1595
AUC	0.9737	0.9107	Val: 0.3432
ROC-AUC Score	-	0.9137	-
Accuracy Score	-	0.8796	-

Bảng 1: Kết quả huấn luyện sau 25 epochs

### 2. Giải thích các chỉ số đánh giá

#### a) Accuracy (Độ chính xác)

- Training Accuracy (0.7982): Tỷ lệ dự đoán đúng trên tập huấn luyện. Đạt 79.82% cho thấy mô hình đang học được patterns từ dữ liệu.

- Val\_accuracy (0.6865): Độ chính xác trên tập validation (fold 9). Thấp hơn training accuracy cho thấy có hiện tượng overfitting.

#### b) AUC (Area Under Curve)

- Training AUC (0.9737): Diện tích dưới đường cong ROC trên tập training. Giá trị gần 1.0 rất tốt, cho thấy mô hình phân biệt tốt giữa các lớp.
- Val\_AUC (0.9107): AUC trên validation set. Cao (>0.9) cho thấy khả năng phân biệt tốt ngay cả trên dữ liệu chưa thấy.

#### c) Loss (Hàm mất mát)

- Training Loss (0.1595): Binary Cross-entropy loss trên training set. Càng thấp càng tốt.
- Val\_loss (0.3432): Loss trên validation. Cao hơn training loss gấp đôi - dấu hiệu rõ ràng của overfitting.

#### d) Val\_ROC\_AUC và Val\_accuracy\_score

- Val\_ROC\_AUC (0.9137): Macro-averaged ROC-AUC score tính trung bình từ 5 lớp bệnh. Đây là metric chính để so sánh với các mô hình khác.
- Val\_accuracy\_score (0.8796): Accuracy tính theo cách khác (có thể là micro-average hoặc weighted-average). Đạt 87.96% là kết quả tốt.

### 3. Phân tích kết quả

Kết quả ban đầu cho thấy:

- Độ chính xác trên tập training đạt 79.82%, cho thấy mô hình đang học được các đặc trưng từ dữ liệu.
- AUC trên tập training đạt 0.9737, thể hiện khả năng phân biệt tốt giữa các lớp.
- ROC-AUC score trên tập validation đạt 0.9137, cho thấy mô hình có khả năng tổng quát hóa khá tốt.
- Accuracy score trên validation đạt 0.8796 (87.96%), là kết quả khá tốt so với mục tiêu ban đầu.
- Sự chênh lệch giữa training accuracy (79.82%) và validation accuracy (68.65%) cho thấy vẫn còn hiện tượng overfitting cần được giải quyết trong các epochs tiếp theo.

Toàn bộ code huấn luyện và kết quả chi tiết có thể xem tại:

[https://colab.research.google.com/drive/1j1weVB4MwCqNejUHe\\_Ogk9jOSEYGp\\_zq](https://colab.research.google.com/drive/1j1weVB4MwCqNejUHe_Ogk9jOSEYGp_zq)

## **IV. KẾ HOẠCH TIẾP THEO**

### **1. Tuần 6 (03/02/2026 - 09/02/2026): Fine-tuning và đánh giá**

Mục tiêu chính của tuần 6 là tinh chỉnh mô hình để đạt được hiệu suất tối ưu:

- Tiếp tục huấn luyện mô hình cho đến khi đạt early stopping hoặc hết 60 epochs.
- Thử nghiệm các chiến lược điều chỉnh learning rate để cải thiện hội tụ.
- Điều chỉnh các hyperparameters (dropout rate, L2 regularization) để giảm overfitting.
- Đánh giá chi tiết hiệu suất trên tập test (fold 10) theo các metrics: Accuracy, ROC-AUC, F1-score.
- So sánh kết quả với các mô hình baseline (ResNet101, ECGNet, Rajpurkar et al.).
- Phân tích chi tiết kết quả theo từng lớp bệnh lý (NORM, CD, MI, HYP, STTC).

### **2. Tuần 7 (10/02/2026 - 16/02/2026): Tinh chỉnh và báo cáo cuối**

Tuần cuối cùng tập trung vào hoàn thiện dự án:

- Trực quan hóa (visualization) các attention weights ở 3 cấp độ (beat, rhythm, channel).
- Thực hiện ablation study để đánh giá vai trò của từng thành phần trong mô hình.
- So sánh hiệu suất khi sử dụng các tổ hợp kênh khác nhau (limb leads, precordial leads, 12 leads).
- Kiểm chứng tính giải thích (interpretability) của mô hình với các trường hợp ASMI và IMI.
- Hoàn thiện báo cáo cuối kỳ với đầy đủ kết quả, biểu đồ và phân tích.
- Chuẩn bị slide thuyết trình và demo mô hình.

