**1. Brief introduction**

This task involves segmenting regions of interest from ABP waveforms, where a label is assigned to each sample point. For the model, I used a 1D U-Net, which was originally designed for pixel-wise image segmentation and translates well to point-wise classification in time series data.

**2. Clarifications about labels**

The whole pipeline is designed to perform 4-class segmentation and use 10-fold cross-validation for model evaluation:

- 0: nothing

- 1: Diastolic chest compression

- 2: Systolic events (I combined original label 2 and 4 to label2, so this label currently includes systolic events from chest compression and normal heartbeat).

The reason I combined these two labels is that the morphology of this region in both compressed and normal ABP waveforms is very similar, and since our target is the diastolic region, I combined them.

- 3: Diastolic normal heartbeat

3**. Core scripts:**

1. main\_pytorch.py - Main execution script (PyTorch version)
2. unet\_model\_pytorch.py- U-Net architecture implementation

- 1D U-Net adapted for signal segmentation

- Weighted CrossEntropyLoss for class imbalance

- Training and evaluation methods

1. data\_preprocessing.py - Data preprocessing and cross-validation setup

- 4-class label mapping (combines labels 2 and 4)

- Signal normalization

- Creates 10-fold cross-validation splits

- Saves preprocessed data

1. training\_pipeline\_simple.py - Training pipeline

- 10-fold cross-validation training

- Model evaluation and metrics calculation

- Results saving and statistics

1. generate\_overall\_results.py - Results aggregation

- Combines results from output json files from all folds

- Calculates overall statistics

- Generates summary reports

**4. Current results from 10-fold cross validation:**

*Overall cross-validation results:*

Mean ACCURACY: 0.9758

Mean MACRO\_PRECISION: 0.8818

Mean MACRO\_RECALL: 0.9724

Mean MACRO\_F1: 0.9221

Mean MACRO\_DICE: 0.9221

Mean MACRO\_IOU: 0.8613

*Per-class results (Mean values across folds):*

Class 0 (nothing):

PRECISION: 0.9963

RECALL: 0.9760

F1: 0.9860

DICE: 0.9860

IOU: 0.9724

Class 1 (diastolic\_Compression):

PRECISION: 0.7385

RECALL: 0.9633

F1: 0.8357

DICE: 0.8357

IOU: 0.7181

Class 2 (systolic\_Compression\_normal):

PRECISION: 0.9512

RECALL: 0.9886

F1: 0.9695

DICE: 0.9695

IOU: 0.9409

Class 3 (diastolic\_normal):

PRECISION: 0.8411

RECALL: 0.9618

F1: 0.8972

DICE: 0.8972

IOU: 0.8138

**5. Some thoughts:**

1. Currently, I treat each segment as an independent unit. The issue with the current version is that segments from the same patient could appear in both the training and testing sets. To avoid this type of data leakage, if patient IDs are available, we can split the data by patient ID using the ‘GroupKFold’ function. This cross-validation splitter ensures that the same group is not represented in both the training and validation folds.
2. From what I’ve observed with the sample dataset, there are a couple of strategies that could help improve performance in the future.

First, instead of using a label length of only 7 points, I would try something longer (e.g., 20+ points). Seven points are too short—the waveform within such a narrow window can resemble other morphologies, many of which also appear in the “nothing” class, making the model prone to confusion. With a longer window and some post-processing, we could maintain localization accuracy while improving segmentation

Second, I found something interesting we could explore in the signal stage. For example, during compression, the diastolic peaks only appear on the rising edge of the ABP pulse, while in a normal pulse they usually appear on the descending edge or in other regions. Instead of working directly with the raw waveforms, if we take the first derivative to highlight the rising edge and restrict the peak search to that region, we can filter out many false positives. In addition, the derivative amplifies the morphological differences among classes, which should further improve segmentation. I demonstrated this in the following figure: the first column shows the ABP waveforms from the first 10 segments, the second column shows the corresponding labels for each segment, and the third column shows the waveforms after applying the first derivative. Before applying the derivative, the region of class 1 may appear similar to class 0 and class 3 due to the narrow span. However, after applying the derivative, the region of class 1 becomes clearly distinguishable from class 3 (can see figures of segment 2, 4, 6, 7, 10 VS segment 5, 8, 9).

A screenshot of a graph

AI-generated content may be incorrect.