B.Comp. Dissertation

Automated ECG Diagnosis using an Explainable AI Framework

By Tian Fang

Department of Computer Science

School of Computing

National University of Singapore

2022/2023

B.Comp. Dissertation

Automated ECG Diagnosis using an Explainable AI Framework

By Tian Fang

Department of Computer Science

School of Computing

National University of Singapore

2022/2023

Project No: H220330

Advisor: Asst Prof Brian Lim Youliang

Deliverables:

Report: 1 Volume

Abstract

Cardiovascular disease (CVD) is a primary cause of mortality globally, and the electrocardiogram (ECG) is a commonly used diagnostic tool for its detection. While Artificial Intelligence (AI) has shown an exceptional predictive ability for CVD, the lack of interpretability has deterred medical professionals from its use. To address this, we developed an explainable AI (XAI) framework that integrates ECG rules expressed in the form of first-order logic (FOL). The framework can uncover the underlying model's impressions of interpretable ECG features, which can be crucial for cardiologists to understand the diagnosis predictions generated by our system. Our experiments demonstrate the benefits of incorporating ECG rules into ECG AI such as improved performance and the ability to generate a diagnosis report that provides insights into how the model derived the predicted diagnoses. Overall, our XAI framework represents a great step forward in integrating domain knowledge into ECG AI models and enhancing their interpretability.

Subject Descriptors:

Machine Learning

Neural-Symbolic Learning

Keywords:

Explainable Artificial Intelligence, First-order Logic, Electrocardiogram

Implementation Software and Hardware:

Pytorch 2.0, Pytorch Lighting 1.9.4, NeuroKit2 0.2.3, Optuna 3.2, CSCHPC

Acknowledgment

I wish to express my sincere appreciation to my supervisor, Professor Brian Lim Youliang, for his invaluable guidance, mentorship, and support throughout the entire duration of this Final Year Project. His keen insights, attention to detail, and patient guidance have been critical in helping me navigate the complexities of this research.

I would also like to extend my heartfelt thanks to my family and friends, whose unwavering support and encouragement have been a constant source of motivation throughout this journey.

Table of Contents

1. Introduction

Cardiovascular disease (CVD) is one of the leading causes of death worldwide, particularly in developing countries (Timmis et al., 2020). In 2019, World Health Organization (WHO) reported that around 17.9 million people died from CVDs, accounting for 32% of deaths globally (WHO, n.d.). Electrocardiogram (ECG) is a widely used, low-cost, and non-invasive medical test for diagnosing various CVDs in clinical practice (Surawicz & Knilans, 2008). In the United States, ECGs are ordered in approximately 5% of office visits (Strodthoff et al., 2020), indicating their essential role in diagnosing CVDs.

Before the era of Artificial Intelligence (AI), particularly Machine Learning (ML), ECG diagnosis is mostly treated as a pattern recognition problem (Hegadi, 2014). This is also the case for cardiologists and the ECG's effectiveness heavily relies on the experts' interpretation and experience in detecting ECG patterns (Siontis et al., 2021). Compared to humans, AI is particularly good at exploiting hidden subtle patterns in ECG, and recent ECG AIs have shown exceptional performance in predicting CVD (Somani et al., 2021). Despite being a promising technology, the adoption of ECG AI in hospitals is still limited mainly due to its lack of explainability. To elaborate, in spite of the extraordinary performance, virtually all current ECG AI offers little explanation of why the ECG AI makes certain decisions (Somani et al., 2021). The black-box nature of these ECG AI makes doctors reluctant to bare the risk of wrong AI diagnosis due to liability concerns (Teodoridis, 2022).

To alleviate cardiologists' concerns when using AI-aided ECG systems, we can follow their thought process and go through a systematic process called differential diagnosis[1] (DDx). During the DDx process, doctors eliminate candidate diseases one by one, following a set of defined ECG rules while considering the patient's demographics, symptoms, and medical test results. However, ECG rules (Khan, 2008) are known to be complex, and are often challenging for cardiologists to grasp, let alone general practitioners or doctors in the emergency room who need to urgently interpret the ECG.

Therefore, we aim to create an explainable ECG AI that can efficiently incorporate a wide range of ECG rules to make accurate diagnosis predictions. To achieve this, we have created an

[1] The term "differential diagnosis" may also refer to the remaining diagnoses after the process of elimination.

explainable AI (XAI) framework to integrate ECG rules that can be expressed in first-order logic (FOL). For an input ECG, our ECG-XAI framework will process it and internally capture the ML model’s perceptions of the ECG’s explainable features, which are comprehensible to cardiologists. This enables the system to generate a diagnosis report that not only predicts the diagnosis but also explains how diagnoses were derived, increasing transparency and trustworthiness.

2. Basic concepts of ECG

To facilitate the discussion, this section will briefly introduce the basic concepts of ECG. To start with, the depolarization and repolarization of myocardial cells result in the contraction and relaxation of cardiac muscles, respectively (Surawicz & Knilans, 2008). Such processes will generate electrical impulses, which can be detected by ECG using electrodes placed on specific locations of a patient's skin (Meek & Morris, 2002). Each ECG is presented as a graph of voltage versus time, describing the electrical activities of cardiac cycles (Lilly, 2012). An ECG that deviates from the normal ECG pattern may indicate various CVDs described in the last part of this section.

2.1 Electrodes placement and leads

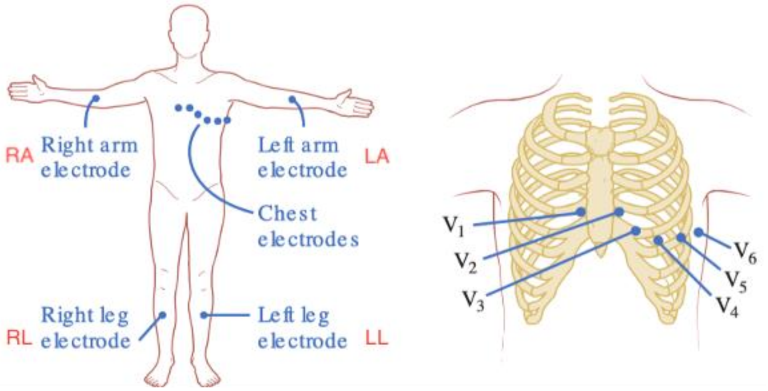


Figure 1 Placement of electrodes (Lilly, 2012)

During an ECG test, 10 adhesive pads called electrodes are attached to the skin: 4 placed on limbs (RA, LA, RL, LL[2]) and 6 placed on the chest (V1-V6) as shown in Figure 1. A lead is the electrical potential difference between a pair of electrodes. The 12 leads of a normal ECG can be broken down into 3 categories: 3 bipolar limb leads I, II, and III; 3 augmented limb leads aVF, aVL, and aVR; and 6 precordial/chest leads (V1-V6). Each of the 3 bipolar limb leads has actual limb electrodes as its negative and positive electrodes (e.g., I = LA - RA). Whereas the augmented limb leads and chest leads are unipolar leads, whose negative electrode is a virtual electrode (V_W) calculated by averaging the electrical potential of LA, RA, and LL (Lilly, 2012). Table 1 summarizes how different leads are derived.

[2] RA = Right Arm, LA = Left Arm, RL = Right Leg, LL = Left Leg

Lead	(-) Electrode	(+) Electrode
I	RA	LA
II	RA	LL
III	LA	LL
aVR	V_w	RA
aVF	V_w	LL
aVL	V_w	LA
V1	V_w	V1
V2	V_w	V2
V3	V_w	V3
V4	V_w	V4
V5	V_w	V5
V6	V_w	V6

Table 1 Summary of the 12 leads

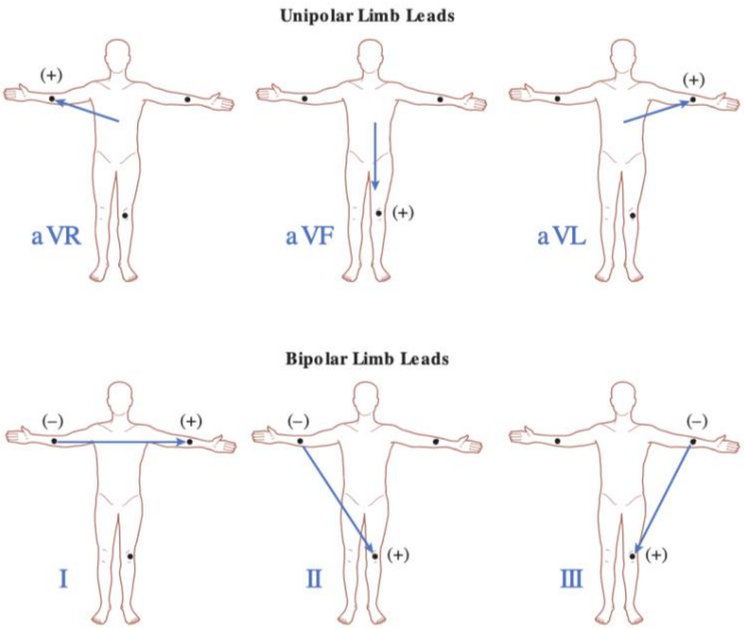


Figure 2 Illustration of limb leads (Lilly, 2012)

Moreover, the deflection of each lead has the following implications: the depolarization of the heart toward (resp. away) the positive electrode yields a positive (resp. negative) deflection; the repolarization of the heart toward (resp. away) the positive electrode yields a negative (resp. positive) deflection (Schrepel et al., 2021).

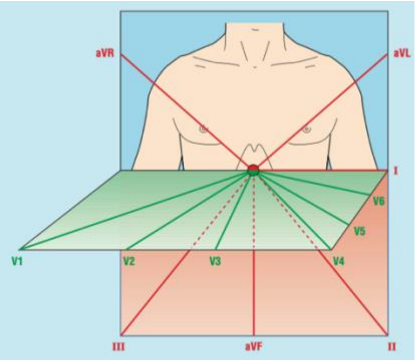


Figure 3 Leads give both vertical and horizontal views of the heart (Meek & Morris, 2002)

Furthermore, the 12-leads system gives a three-dimensional view of the heart which is critical for examinations of the heart's structural abnormalities. Specifically, the 6 limb leads describe the vertical plane, and the 6 chest leads describe the horizontal plane as shown in Figure 3(Meek & Morris, 2002).

2.2 Components of an ECG

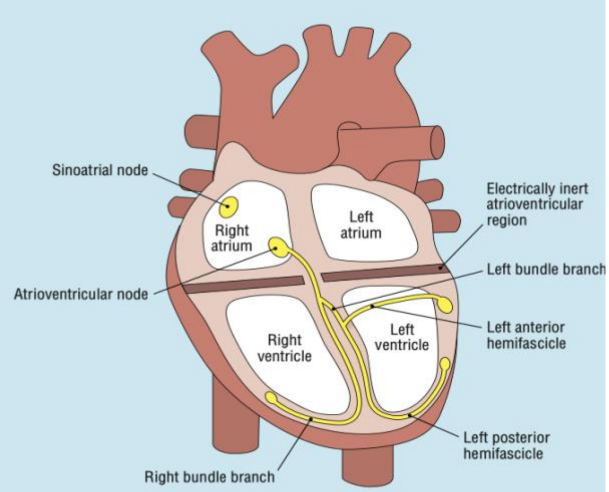


Figure 4 Electrical conduction system of the heart (Meek & Morris, 2002)

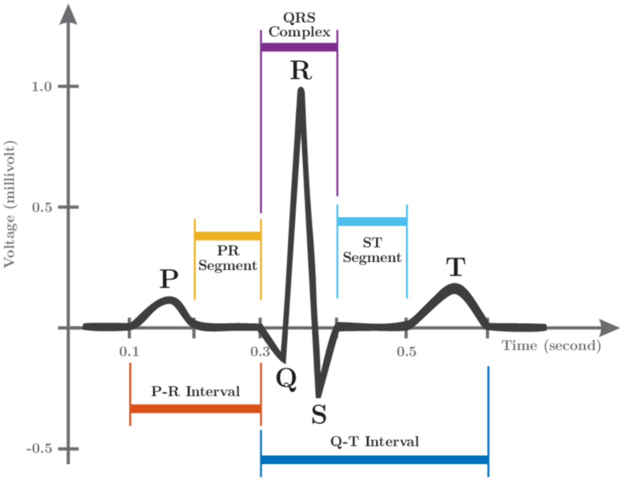


Figure 5 ECG components (Parsi, 2021)

To understand each component of an ECG, one should first learn the electrophysiology related to cardiac movements. Each cardiac cycle begins with atrial depolarization[3], initiated by the sinoatrial (SA) node located at the right atrium as shown in Figure 4. The electrical impulses generated by atrial depolarization then spread throughout the atria, travel through the atrioventricular (AV) node, and finally reach the ventricles via the downstream fascicles of the conduction system (Meek & Morris, 2002). Hence, there is a delay between atrial depolarization and ventricular depolarization. After depolarization, the atria and ventricles go through the process of repolarization[4], which similarly begins at the atria and ends at the ventricles. Therefore, each cardiac cycle comprises two phases, one for heart contraction (systole phase), and another one for heart relaxation (diastole phase) (Lilly, 2012).

With the knowledge introduced above, one may understand why an ECG is decomposed in the following way (Parsi, 2021): the P wave, which represents atrial depolarization; the QRS complex, which represents ventricular depolarization; and the T wave, which represents ventricular repolarization. The QRS can be further decomposed into Q wave, R wave, and S wave, though the Q wave and the S wave may not be present (even in normal ECG). The atrial

[3] Atrial depolarization results in the contraction of the atria
[4] Repolarization results in heart relaxation

repolarization happens during the QRS complex, although the electrical changes incurred by the atrial repolarization are not pronounced compared to the ventricular depolarization (Surawicz & Knilans, 2008). Other important ECG segments and intervals are shown in Figure 5.

2.3 Rate and rhythm

The first question to ask while interpreting an ECG is whether the heart rate and rhythm are normal. The heart rate is defined to be the rate at which the SA node depolarizes as it marks the beginning of a cardiac cycle. The normal heart rate for an adult is 60-100 beats per minute (bpm). A heart rate falling below 60 bpm is termed bradycardia, while a heart rate greater than 100 bpm is termed tachycardia (Meek & Morris, 2002).

The cardiac rhythm in a normal resting heart is called normal sinus rhythm (NSR), which leads to the typical P-QRS-T pattern on the ECG as shown in Figure 5. A rhythm that deviates from NSR is called an arrhythmia (Surawicz & Knilans, 2008).

2.4 Cardiac axis

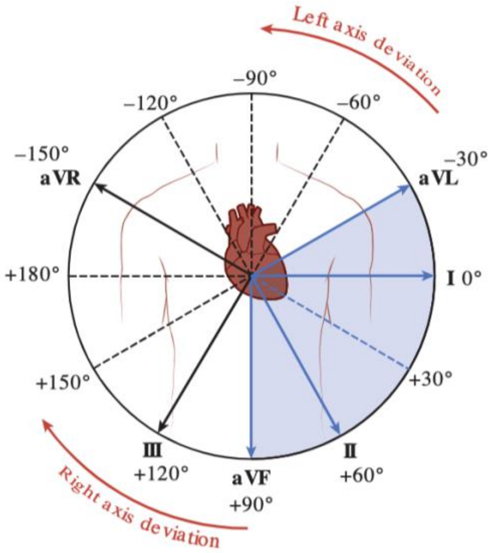


Figure 6 Hexaxial diagram for cardiac axis (Lilly, 2012)

The cardiac axis or QRS axis is the mean direction of the ventricular depolarization wave in the frontal plane (Meek & Morris, 2002). The direction of the lead I is the zero reference point of the hexaxial reference system as shown in Figure 6. The normal value for the cardiac axis is -30 degrees to +90 degrees, which can be determined by whether the QRS complex is mostly positive in lead I and II (Meek & Morris, 2002). A cardiac axis that is smaller than -30 degrees indicates left axis deviation, while a cardiac axis that is bigger than +90 degrees indicates left axis deviation (Lilly, 2012). A deviated cardiac axis may indicate enlargement of heart chambers (Hypertrophy) or impairment of the heart’s conduction system (Ashley & Niebauer, 2004).

2.5 ECG diagnosis

Despite its low cost, ECG can provide insights into a considerable number of CVDs. The CVDs related to ECG can be broadly categorized into the following categories: Arrhythmias (ARR), Ischemia (ISC) and Myocardial Infarction (MI), Conduction Disturbance (CD), and Hypertrophy (HYP). Table 2 describes these categories in detail.

CVD	**Description**
Arrhythmias	Heart rhythm that deviates from the normal sinus rhythm
Ischemia and Myocardial Infarction	*Myocardial infarction* is caused by tissue death (infarction) of the heart muscle (myocardium) because of prolonged *ischemia*, which is the lack of oxygen delivery to the myocardium
Conduction Disturbance	Damage or obstruction (block) in the heart’s electrical conduction system
Hypertrophy	Enlargement of heart chambers

Table 2 Description of major categories of ECG diagnosis

3. Literature Review

This section will provide an overview of the current state of ECG AI including ML algorithms applied to ECG and XAI techniques used in ECG AI. Additionally, it will highlight previous efforts to integrate logical rules into neural networks. At the end of this section, we will explore publicly available ECG datasets and determine the one that will be used for this project.

3.1 Machine Learning and ECG

Numerous studies have explored the potential of using ML in ECG diagnosis. Being a data-driven modeling technique that excels at learning subtle patterns automatically from the dataset, ML shows superb performances in predicting heart diseases using ECG (Siontis et al., 2021). Jahmunah et al. (2021) give a summary of the ML models employed in the previous works about ECG AI and the most common ML models are Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). RNN models are by design suitable for sequential data such as ECG time series (Tealab, 2018). The most commonly used RNN variant for ECG data is LSTM (Jahmunah et al., 2021). As for CNN models, most of them used one-dimensional (1D) convolution that exploits temporal relationships for the time series of each ECG lead. However, some studies use two-dimensional (2D) convolution on stacked time series of 12 ECG leads, with the intent to exploit both temporal and spatial relationships of ECG recordings (Siontis et al., 2021). Nevertheless, this approach is questionable as it may yield erroneous relationships when convolving two leads that are not spatially adjacent (e.g., leads V1 and III).

Some papers have developed more advanced ML models on ECG data, usually by extending the existing architectures. A great example is combining CNN and RNN, where CNN is used to extract sequential features to be fed into RNN (Lih et al., 2020). Another example is the residual neural network (ResNet), which adds skip connections to resolve traditional CNN’s vanishing gradients problem (Wang et al., 2017). He et al. (2018) extend ResNet further and get xResNet, the best-performing model on the PTB-XL dataset with a macro-averaged AUC (Area Under Curve) of 0.925 when predicting all types of labels (Strodthoff et al., 2020).

It is worth noting that despite numerous efforts made to improve prediction accuracy, few studies have offered explanations for ECG AI’s predictions (Somani et al., 2021). Even among those that did attempt to interpret the AI model, many only used post hoc XAI techniques that provide little useful information from a cardiologist's perspective, which will be discussed in the following section.

3.2 Explainable AI techniques for ECG AI

As shown in the Introduction, one major obstacle to ECG AI's adoption in hospitals is doctors' reluctance to trust predictions from a black-box AI. There have been several attempts to improve ECG AI's explainability using XAI techniques such as the Local Interpretable Model-Agnostic Explanations (LIME) (Hughes et al., 2021), Gradient-weighted Class Activation Mapping (Grad-CAM) (Hicks et al., 2021; Raza et al., 2022; Taniguchi et al., 2021), and Shapley Values Additive Explanations (SHAP) (Anand et al., 2022; Zhang et al., 2021). These studies used XAI techniques to achieve similar goals of highlighting the parts of the ECG that contribute most to the model's prediction.

However, these XAI techniques have three limitations. Firstly, although these techniques are domain-agnostic and can be applied to any ML model, they may not generate explanations that are domain-specific and intuitive for trained cardiologists. For instance, simply highlighting abnormal ST segments is not sufficient for diagnosing MI because cardiologists should also check whether the ST segment is elevated or depressed as well as other components (namely Q wave and T wave) of the ECG. Secondly, ECG patterns that require examining multiple cardiac cycles are not well represented through highlighting. For example, the hallmark of atrial fibrillation (A-fib)[5] is abnormal rhythm with inconsistent patterns across cardiac cycles. In this case, the above XAI techniques might highlight most parts of the ECG, which again is not insightful. Last but not least, these XAI techniques are post hoc analyses. Therefore, one may conclude fallacious relationships between ECG patterns and CVD diagnosis, which might not correspond to the electrophysiology behind an ECG test.

The model that most align with this project's objectives is proposed by Jo et al. (2021). The model aims to predict whether the ECG shows A-fib (binary classification). To improve interpretability, the predictor makes use of results from two submodules that detect heart rhythm irregularity and the presence of a P wave. Although these two submodules indeed yield intermediate ECG characteristics comprehensible to cardiologists, this ECG AI still has areas to be improved. To begin with, this model has limited practical value as it only focused on A-fib,

[5] A form of arrhythmia

while ECG can generate predictions for myriads of CVDs. Additionally, this model's submodules straightforwardly apply ResNets on the whole ECG time series to obtain the intermediate characteristics. However, it would make more sense to first delineate the ECG into segments (either through traditional pattern recognition methods or through ML methods) and focus on only the P wave part.

3.3 Neural Networks and Logic

Our attempts to integrate ECG rules, particularly those that can be expressed in FOL, into neural networks can be seen as an instance of Neural-Symbolic Learning (Besold et al., 2017).

One of the widely adopted approaches for combining logic with ML models is through modification of the loss functions. This effectively regularizes the network to enforce constraints or minimize inconsistencies among predictions (Du et al., 2019; Minervini & Riedel, 2018). Taking this a step further, Xu et al. (2018) introduced a novel approach that incorporates general logical constraints about the output vectors into the loss functions. There are also some other methods that introduce additional structures to facilitate logical reasoning such as the teacher-student framework proposed by Hu et al. (2020), which can transfer FOL rules into the neural networks through iterative distillation.

Although the abovementioned methods all leverage domain knowledge in some way to produce predictions that are more aligned with the logical rules, they often do not prioritize explainability. As a result, even when the model's predictions are more logical and sensible, it may still be challenging to understand how the model arrived at those predictions.

One method that can indeed improve explainability is proposed by Li and Srikumar (2019), which involves modifying the pre-activated value (MPAV) before the activation function. However, it also has some limitations such as not being able to incorporate the comparison operators, which are crucial in ECG rules as they involve numerous thresholds and comparisons. With that being said, MPAV's implementation of implication remains valuable, and we will adapt it into our framework.

To address MPAV's limitation, we developed soft thresholds, which integrate comparison operators into our framework in a way that is not binary or brittle. The results from our comparison operators are the basic building blocks for "impressions", a concept we introduced to capture our model’s perceptions of explainable features that are comprehensible to cardiologists. These impressions can be further combined using the probabilistic soft logic (Beltagy et al., 2014) to enhance the explainability. We will provide further details about soft thresholds and soft logic in the “Methodology” section.

3.4 Publicly available ECG datasets

Below are publicly available 12-lead ECG datasets that may assist this project’s model development and validation. We eventually chose the PTB-XL as our dataset considering its large record count and comprehensive labeling.

3.4.1 Shaoxing People's Hospital dataset

Shaoxing People's Hospital dataset (Zheng et al., 2020) is comprised of 10-second multi-labeled ECG records of 10,646 patients, featuring 11 arrhythmia labels and 67 additional labels for other disorders. The labels are manually annotated by professional cardiologists. Although this dataset did not include patients with other major CVDs diagnosable via ECG (i.e., ISC and MI, CD, HYP), this dataset has the most diverse arrhythmia labels, which may help the development of this project’s arrhythmia model.

3.4.2 Shandong Provincial Hospital dataset

Shandong Provincial Hospital dataset (Liu et al., 2022) is a new dataset that includes 25770 multi-labeled ECG records (10~60 seconds) with corresponding demographics from 24666 patients. There are 44 labels covering major categories of ECG diagnosis. Although few patients with MI are included in this dataset, the myriad labels of this dataset make it a great validation dataset to test the ML model’s generalizability.

3.4.3 Lobachevsky University Electrocardiography Database

Lobachevsky University Electrocardiography Database (LUDB) (Kalyakulina et al., 2020) consisted of 200 multi-labeled 10-second ECG signal records. It has extensive labels for various aspects of ECG such as heart rate and rhythm, cardiac axis, ISC and MI, CD, and HYP. More importantly, ECG records in LUDB have been manually delineated/segmented into P wave, QRS complex, and T wave by cardiologists. Therefore, despite having few records, LUDB can be used to verify the accuracies of ECG delineation tools.

3.4.4 PTB-XL

PTB-XL (Wagner et al., 2020) contains 21837 multi-labeled 10-second ECG records gathered from 18885 patients. In total, PTB-XL has 71 labels consisting of diagnostic labels describing specific CVDs associated with the ECG, form labels describing ECG's morphology, and cardiac axis labels. Table 3 summarizes the number of records with diagnostic labels that fall under each major category, from which one can see that there are sufficient training data for each major category. Hence, PTB-XL is chosen as the dataset for this project.

# Records	Major Categories	Description
9517	NORM	Normal ECG
4132	ARR	Arrhythmia
8118	ISC and MI	Ischemia and Myocardial Infarction
4901	CD	Conduction Disturbance
2649	HYP	Hypertrophy

Table 3 Number of records for each major category of ECG diagnosis

Moreover, Strodthoff et al. (2020) have provided a framework for testing models' performances on PTB-XL. In addition to the framework, they have implemented various state-of-the-art (SOTA) ML algorithms on PTB-XL, which may be compared with this project's model.

4. Methodology

To generate an ECG report with an explanation comprehensible to cardiologists, the ECG-XAI system should carry out a DDx process similar to the one that the cardiologists use. The DDx process carried out by our system can be broken down into two steps. First, as described in the following “Preprocessing” section, the system will extract relevant ECG features such as the heart rate. Subsequently, the system will utilize the methods described in the “ML Model Augmentation using FOL” section to build architectures (described in the “Architectures” section) that incorporate FOL rules related to the ECG DDx process. Finally, the “ECG-XAI framework” section will briefly introduce our easy-to-use framework that captures the above functionalities in a scalable fashion.

4.1 Preprocessing

4.1.1 ECG Cleaning and Delineation

The first step of preprocessing is ECG cleaning and delineation. During delineation, an ECG record is segmented into P-QRS-T waves. This is a crucial step as downstream tasks and the explainability of the DDx results heavily rely on the accuracy of the delineation.

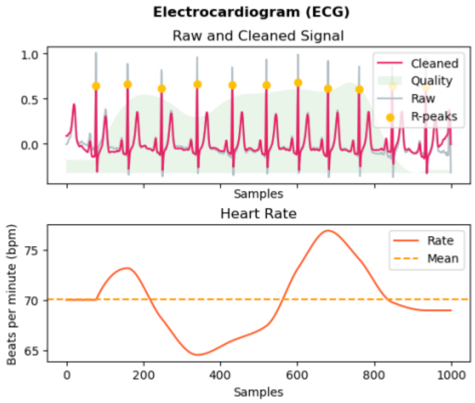


Figure 7 Lead II of a normal ECG

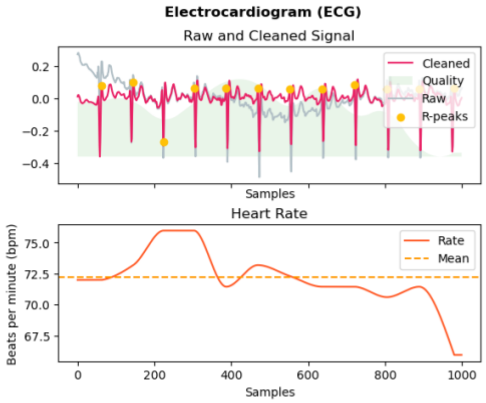


Figure 8 Lead III of a normal ECG with baseline drift

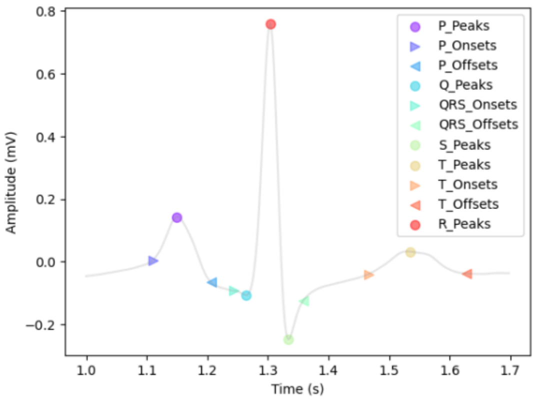


Figure 9 Delineation of Lead II of a normal ECG

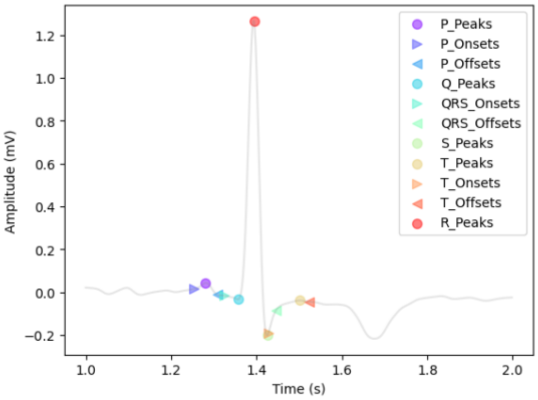


Figure 10 Incorrect delineation of Lead V5 of an ECG with inverted T waves

Makowski et al. (2021) have provided a package called NeuroKit2 for ECG cleaning and delineation. ECG records from PTB-XL are used to test the NeuroKit2 package. The first feature offered by this package is ECG signal cleaning via removing high-frequency noise, adjusting drifted baseline, etc. Figure 8 shows that the package has done a decent job of cleaning the ECG signal. Moreover, using wavelet transform, this package can delineate an ECG signal and obtain peaks and boundary points of different ECG waves. The algorithm provided by the package usually yields satisfactory results for normal ECGs as shown in Figure 9. However, it encounters problems when the ECG shows abnormal patterns such as inverted T waves as shown in Figure 10.

To correctly identify peaks and boundaries of inverted waves (namely inverted P or inverted T waves), we create a custom delineation function to first check whether the wave is inverted based on delineation results from the NeuroKit2. There are two criteria for detecting inverted waves. The first one is checking whether the voltage at the wave peak is negative and the second one is checking whether the voltage at the wave peak is lower than the voltages at the wave onsets and offsets. If an inverted wave is detected in a lead, our delineation function will feed the inverted ECG lead into NeuroKit2's delineation algorithm and use the delineation of the inverted lead to correct the inverted waves in the original ECG lead. As shown in Figure 11, our custom delineation function successfully corrects the segmentation mistakes made in Figure 10.

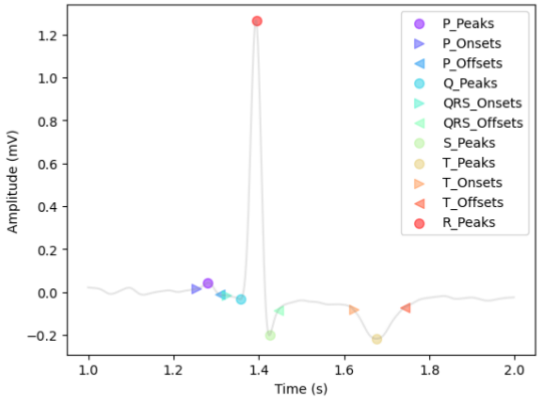


Figure 11 Correct delineation of Lead V5 of an ECG with inverted T waves

4.1.2 Extracting Objective Features

After cleaning and delineation of ECG, each lead of an ECG record can be decomposed into several cardiac cycles with their respective P-QRS-T waves. Then we may proceed to extract objective features for each ECG record. These objective features are either continuous features such as heart rate, or binary features such as whether the PR interval is prolonged. The "objective" here indicates that the features are computed directly using established ECG rules rather than the soft rules introduced in the next section. The reason why we are extracting these objective features is that we will compare and align the model's impressions with these objective features to ensure that the model's impressions make sense and do not deviate too much from the normal ECG rules. Additionally, cardiac cycles may output different objective features due to factors such as noise or variability between cardiac cycles. For instance, the PR interval in one cardiac cycle may be greater than the threshold 200ms, while the next cycle's PR interval does not exceed 200ms. Therefore, the ECG record's objective features are computed as the averages of the objective features across cardiac cycles. If a feature is continuous such as the PR interval, then its aggregated feature is the average estimate of the continuous feature. If the feature is binary such as "whether the PR is prolonged", then the aggregated feature reflects the percentage of cardiac cycles where the binary feature is true. All objective features used in this project are summarized in Appendix A2.

4.2 ML Model Augmentation using FOL

As will be shown in the "FOL in ECG DDx Process" section, a considerable number of rules in ECG diagnosis can be described using FOL. Hence, it may be beneficial to develop a method to inject FOL rules into an ML model to make it more aware of the constraints in the ECG DDx process. The following sections will first introduce basic FOL concepts used in this project, followed by how to incorporate simple comparison operators and implications into ML models, and finally how to incorporate more general FOL rules.

4.2.1 Basic FOL Concepts

To formally formulate rules in the ECG diagnosis, we can make use of the concept of formula in FOL. In FOL, a formula is a well-formed expression made up of symbols from the FOL alphabet/language. Below describes the subset[6] of FOL symbols and formulas that will be used in this project.

FOL has two types of symbols. The first type of symbol is the logical symbol, and the main type of logical symbol used in this project is logical connectives. The logical connectives employed in our ECG-XAI framework include the following: $\wedge$ for conjunction, $\vee$ for disjunction, $\rightarrow$ for implication, ~ for negation. The second type of symbol is the non-logical symbol, and the main type of non-logical symbol included in this project is the predicate. A predicate typically describes the relationship between the input variables. In the case of our system, comparison operators (a type of predicate) such as greater than (>) and less than (<) are commonly used.

With the help of symbols, we can then construct formulas inductively using:

- Propositional variable: a variable itself can be a formula if it is either true or false.
- Predicate: predicates applied to a set of variables/terms is a formula

[6] Other FOL concepts such as Function, Equality, Quantifiers are omitted for simplicity as the current subset of FOL is sufficient to describe rules used in ECG diagnosis

- Logical Connectives: if φ and ψ are formulas, then the expressions formed by connecting them using logical connectives are also formulas. For instance, ~φ, φ → ψ, and φ ∧ ψ are also formulas.

Moreover, a literal is defined to be either a propositional variable, a predicate applied to variables, or a negation of them. Furthermore, we can ground a formula by replacing variables in the formula with actual values corresponding to those variables. The resulting formula is called the ground formula.

A great number of ECG constraints can be formed using the above building blocks, and the next section will discuss how to incorporate formulas into ML models to make use of our prior knowledge.

4.2.2 Incorporate Simple FOL Concepts

We will first focus on utilizing comparison operators and logical connectives in this section and will then use them to build more complex formulas in the "More General FOL Rules" section.

4.2.2.1 Formula with Comparison Operators

As discussed in the "preprocessing" section, the extracted objective feature involving threshold is usually a binary feature (either 0 or 1). Take tachycardia (TACH) as an example, it is a binary feature defined by "TACH = HR > 100", where HR is the heart rate. However, the implication of an HR of 102 is wildly different from an HR of 200. To model this and have richer information retained in the TACH feature, we come up with soft thresholds, whose definition is given below.

Suppose we have a binary feature B = A > Thresh_{A}, where A is a variable and Thresh_{A} is the threshold of A. Then the soft version or impression of B using soft threshold can be defined as

$$B_{imp} = \sigma\left(w\left(A - \text{Thresh}_{\text{A}}(1 + \delta)\right)\right) \qquad (1)$$

where σ is the sigmoid function, δ is a real-valued factor to slightly modify the threshold, and w is a non-negative value to strengthen or weaken the impression. The resulting impression of B is a real number between 0 and 1, which can be further combined using logical connectives in probabilistic soft logic as will be shown in the next section.

The intuition behind modifying the threshold is that in many cases, thresholds used in practice may vary across hospitals or even doctors. A great example is the thresholds for detecting LVH, for which multiple standards coexist. In Khan's book alone (2008), there are four standards introduced, each with its own strength and weakness. Allowing the threshold to fluctuate around the given threshold may overcome the weakness of the given threshold to some extent. That being said, the soft threshold should not deviate too much from the given threshold as that will render the soft threshold meaningless. Hence, we will add a loss $\text{Loss}_\delta = \delta^2$ to the total loss to regulate the δ.

Moreover, we have another parameter w that acts as an amplifier for our impression of feature B. However, to avoid cases where w becomes too large or 0, we will add another loss $\text{Loss}_{\text{feat}}$ which is the cross-entropy between the objective feature B and the feature impression B_{imp} to the total loss to regulate w.

It is worth mentioning that the amount of regulation can be modified with coefficients (presented in the “Experiments” section). And we may not want to overly emphasize on making δ close to 0 or making feature impressions similar to the objective features, as that will make the ML model equivalent to a rule-based system where hard rules are employed.

4.2.2.2 Formula with Logical Connectives

Since the truth values of the propositional variables or predicates in our system are real values between 0 and 1, the traditional definitions of logical connectives that focus on binary truth values will not apply here. Inspired by the probabilistic soft logic used in Li and Srikumar's work (2019), we define our logical connectives as follows:

$$\begin{gathered}\bigwedge_i z_i = \max\left(0, 1 - |z| + \sum_i z_i\right) \\ \bigvee_i z_i = \min\left(1, \sum_i z_i\right) \qquad (2) \\ \sim z_i = 1 - z_i\end{gathered}$$

Here, each z_i is the truth value of a ground formula, and |z| is the number of ground formulas connected using conjunctions. We can see for conjunction, the truth value of the conjunction is 0 even if only one of the z_i is false (i.e., 0). Moreover, if one of z_i in the disjunction is true (i.e., 1), the truth value of the disjunction is 1 regardless of the other z_i.

Now there's only one logical connective left to be modeled: implication. Suppose we have a simple implication: Z → Y. Then Z and Y are called the antecedent and consequent of the implication, respectively. The implication statement is of great significance to modeling ECG rules as many rules can be summarized as "If some evidence Z is observed, then it increases the likelihood of diagnosis Y", which can be concisely described using "Z → Y". It is worth emphasizing that "Z → Y" does not indicate that there's a causal relationship between Z and Y. Rather, the implication "→" here is an abuse of notation to express "… increases the likelihood of …".

We will implement and compare two methods Modify Pre-Activated Value (MPAV) and Hierarchical Lattice (HL) that are capable of incorporating implication statements into ML models.

The first method MPAV is inspired by the work of Li and Srikumar (2019). The basic idea of MPAV is to increase the pre-activated value PAV_Y (i.e., logit, or the raw/unnormalized prediction) of consequent Y by an amount proportional to the truth value z of the antecedent Z. In other words, the modified prediction $\hat{y}$ is

$$\hat{y} = \sigma(PAV_Y + \rho z) \tag{3}$$

where σ is the activation function sigmoid and $\rho \geq 0$ is a scaling factor that controls the strength of the modification.

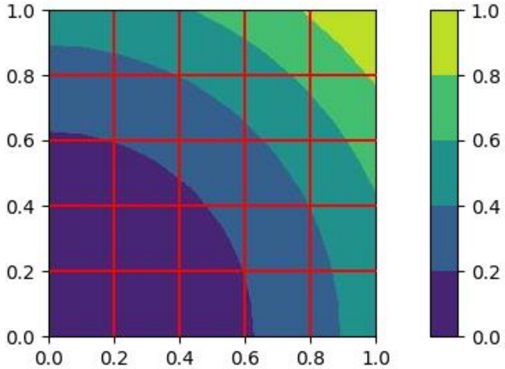


Figure 12 Lattice with $f(\boldsymbol{x}) = \frac{x_1^2 + x_2^2}{2}$

The second method HL (Yanagisawa et al., 2022) is based on the lattice method originally proposed by Google (Gupta et al., 2016). Suppose we want to train a neural network to learn a monotone target function $f(\mathbf{x}) = \frac{x_1^2 + x_2^2}{2}$ where x_1 and x_2 are inputs. If a standard neural network is used to approximate f(**x**), there is no guarantee that the predicted output $\widehat{f(\boldsymbol{x})}$ is monotonically increasing with respect to x_1 and x_2. However, such monotonicity is guaranteed using the lattice method. To achieve this, a lattice layer keeps a look-up table used for approximating the input-output relationships found in data by interpolation. It involves overlaying a standard grid onto the input space and acquiring values for predicted output $\widehat{f(\boldsymbol{x})}$ at the vertices/intersections of the grid. Whenever a test point, **x**, is evaluated, the predicted output $\widehat{f(\boldsymbol{x})}$ is determined through linear interpolation from the surrounding lattice values. Take the lattice shown in Figure 12 as an example, it has a 5x5 grid and 36 vertices. The lattice layer will keep a look-up table for estimated $\widehat{f(\boldsymbol{x})}$ values for those 36 vertices. Afterwards, when the lattice layer receives input (0.1, 0.7), it will locate the nearest four vertices (i.e., (0, 0.6), (0, 0.8), (0.2, 0.6), and (0.2, 0.8)), retrieve their $\widehat{f(\boldsymbol{x})}$ from the look-up table, and perform interpolation to get the $\widehat{f(\boldsymbol{x})}$ for the input (0.1, 0.7). It can be shown that as long as the $\widehat{f(\boldsymbol{x})}$ is monotonically increasing with respect to the vertices, $\widehat{f(\boldsymbol{x})}$ is monotonically increasing with respect to input features as linear interpolation is used for inputs that are not vertices (Yanagisawa et al., 2022). HL method (Yanagisawa et al., 2022) further improved the lattice method by reducing memory consumption and not requiring a projected gradient descent algorithm. In the context of diagnosing a medical condition using ECG, the implication statement "Z → Y" can be generally thought of as a monotone function, where an increase in the truth value of the antecedent Z leads to an increase in the probability of the diagnosis Y. In other words, when there is more evidence (represented by Z) to support a particular diagnosis Y, the likelihood of that diagnosis being correct also increases. Although this makes intuitive sense, it is worth mentioning that if there are some other unconsidered input features that heavily influence f(x), ensuring monotonicity will have a mild or even negative impact on the prediction of f(x). We will explore this further in the "Experiments" section.

4.2.3 More General FOL Rules

4.2.3.1 More complex antecedents

We may construct a more complex antecedent by connecting terms and formulas using logical connectives. Take "$(\sim A \lor B) \land (C \lor D) \rightarrow Y$" as an example, we can eventually reduce the complex antecedent to a truth value using the probabilistic soft logic introduced in the previous section.

4.2.3.2 More complex consequents

We can introduce negation and conjunction in the consequents. To begin with, consider the case where the system encounters an implication statement with a negated consequent such as "Z → ~Y". For the MPAV method, we only need to flip the sign of the modification and the resulting modified prediction will be $\hat{y} = \sigma(PAV_Y - \rho z)$. For the lattice method, we can deem the statement as ~Y is monotonically increasing with respect to Z, and use Y = ~(~Y) to get the predicted probability for diagnosis Y. Moreover, for an implication statement with conjunctive consequents such as "Z → Y ∧ X", we can decompose the statement into several simpler implications with only one consequent term (i.e., "Z → Y" and "Z → X").

4.2.3.3 Generalized Disjunction

As can be seen from the ECG interpretation flowcharts and associated FOL rules in Appendix A3, the occurrence of antecedents in the form of "at least k out of m formula are true" is not rare. For instance, in step 7, at least 2 out of the 5 RVH criteria should be true for the patient to be diagnosed with RVH. This can be seen as a generalized version of the disjunction/OR gate, and its soft logic can be formally defined as a predicate[7] that can take an arbitrary finite number of input terms:

$$GOR_k(z_1, z_2, \dots, z_m) = \min\left(1, \frac{\sum_i z_i}{k}\right) \tag{4}$$

We can see that for the generalized disjunction to be true, at least k out of the m z_i should be true.

4.3 Architectures

In this section, we will make use of the distilled objective features and the methods in the previous section to build architectures that are to be tested in the "Experiments" section. The Multi-Layer Perceptron (MLP) in this section refers to one or more ReLU-activated linear layers. The CNN in this section refers to one or more ReLU-activated 1D convolution layers, each followed by a max-pooling layer. For both MLP and CNN, Batch Normalization is used to regularize the model. Moreover, note that the prediction task is a multi-label task with 21 possible diagnoses, details of which can be found in Appendix A1. Therefore, averaged Binary Cross-Entropy (BCE) loss is used to compare the predicted diagnoses vector $\hat{\boldsymbol{y}}$ with the ground truth diagnoses $\boldsymbol{y}$. We will refer to this loss as the Loss_{dx}. Furthermore, the hyperparameters such

[7] GOR stands for Generalized OR

as the kernel size of the convolution layer are tuned thoroughly for each of the architectures below to make a fair comparison between architectures and ensure that the great performance of a particular model is not due to random chance. Details of the hyperparameter tuning can be found in Appendix A4.

4.3.1 Baseline CNN

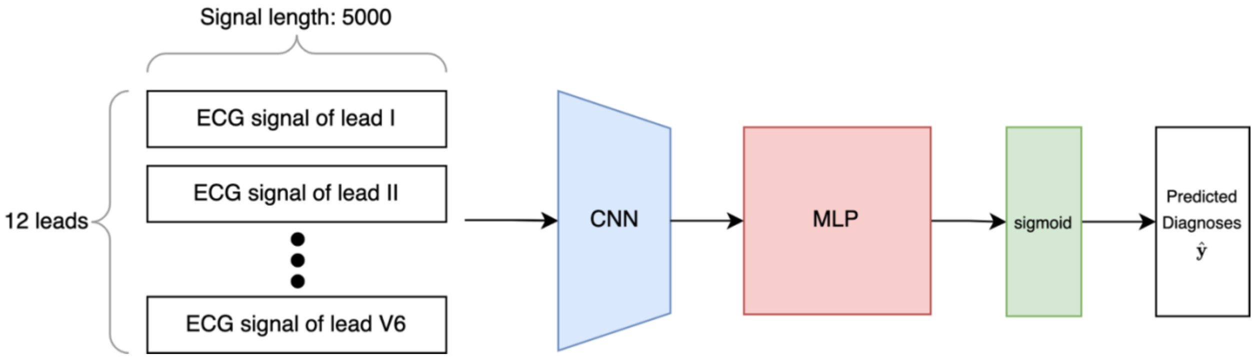


Figure 13 Baseline CNN

The baseline architecture that we are comparing against is a simple black-box CNN network that has not incorporated any medical domain knowledge. As depicted in Figure 13, the baseline network will treat the 12 leads of an ECG record as 12 channels of the input and feed the signal into a 1D CNN block, the output of which is then fed into an MLP followed by a sigmoid layer to generate the final prediction vector $\hat{\boldsymbol{y}}$.

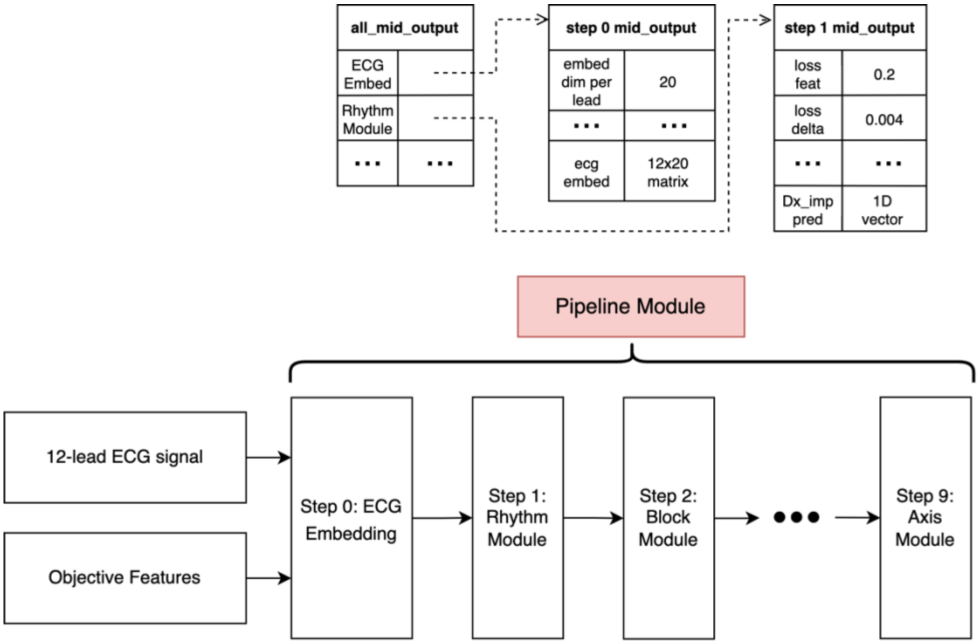
4.3.2 Hard rule system

Another architecture worth comparing to is a simple rule-based system that only uses the “hard version” of the FOL rules. In other words, the truth value is binary (i.e., either 0 or 1) in the hard rule system and we may apply the FOL with their traditional definitions. More specifically, in the hard rule system, the result of a comparison is binary instead of a real value between 0 and 1. Additionally, the consequent of an implication is only true (i.e., 1) if the antecedent is true. This architecture is included in the comparison with the intention to let it mimic the deterministic solution using fixed ECG rules before the era of ML. It is worth mentioning that while the hard rule system may seem completely inflexible, it still has some trainable parameters in the ensemble layers at the end. This is due to the fact that there are no established guidelines on how to combine a diagnosis’s primary criteria with its ancillary criteria (Khan, 2008). Therefore, the ensemble layers should be optimized through training to find an ideal combination of these criteria for accurate diagnosis.

4.3.3 Soft rule system

The soft rule system implemented in our framework will use the aforementioned methods to incorporate the ECG DDx process's FOL rules into the ML models. Our system follows the DDx steps defined in the "Rapid ECG Interpretation" book, whose detailed flowcharts and corresponding FOL rules can be seen in Appendix A3. The following sections introduce the overall architecture, followed by demonstrations on how to create individual modules for each ECG interpretation step.

4.3.3.1 Overarching Architecture



all_mid_output	
ECG Embed	
Rhythm Module	
...	...

step 0 mid_output	
embed dim per lead	20
...	...
ecg embed	12x20 matrix

step 1 mid_output	
loss feat	0.2
loss delta	0.004
...	...
Dx_imp pred	1D vector

Figure 14 Overarching architecture of the soft rule system

The overarching architecture is illustrated in Figure 14, where the cleaned 12-lead ECG signal of shape 12x5000 and the extracted objective features are fed into the Pipeline Module which comprises 10 modules called Step Modules. An important mechanism of the soft rule system is the "all_mid_output" dictionary/look-up table, which serves as a storage of intermediate values/outputs. During the initialization, the Pipeline Module will create an empty "all_mid_output" and pass the "all_mid_output" reference to each Step Module. Subsequently, each Step Module will create its own "mid_output" dictionary and add an entry to the "all_mid_output" using the Step Module's id as the key and the reference to its "mid_output" as the value. In this way, Step Modules can easily communicate with each other, and the pipeline can effortlessly aggregate results.

We will use two examples to illustrate the benefits of having an "all_mid_output" dictionary. Firstly, some steps are dependent on the intermediate results from previous steps (e.g., Step 1-9 depends on the ECG embeddings extracted in Step 0; Step 3 relies on Step 2's prediction about Bundle Branch Block (BBB)). Instead of passing all the intermediate outputs step by step along the pipeline, saving them to the "all_mid_output" allows later steps to directly retrieve only relevant midway outputs created in the earlier steps.

Another example that shows the necessity of the "all_mid_output" dictionary is the Pipeline Module. The presence of the "all_mid_output" enables the Pipeline Module to have functionalities including but not limited to the ensemble of diagnosis predictions from different steps, logging or aggregating intermediate outputs that will be later used for generating diagnosis report, getting the total loss that is a weighted sum of different types of losses. Specifically, diagnoses such as MI may have supporting evidence from Steps 4, 5, and 8. Therefore, The Pipeline Module needs to ensemble diagnosis prediction from those steps. To maximize interpretability, a linear layer is used instead of an MLP to perform the ensemble. Then by looking at the weight of the linear layer, one can tell which step contributes the most to the prediction of that particular diagnosis. Moreover, the total loss at the end has three components: Loss_{dx} which is the BCE loss between the ensembled predictions $\hat{\boldsymbol{y}}$ and the ground truth labels $\boldsymbol{y}$, the sum of individual $\text{Loss}_{\text{feat}}$ from every Step Module, and the sum of Loss_{δ} from every Step Module. To put it more formally:

$$Loss_{total} = Loss_{dx} + \alpha \sum Loss_{feat} + \beta \sum Loss_{\delta} \tag{5}$$

where α and β are the weight constants that respectively control the relative emphasis on "making feature impressions similar to the objective features" and "making the soft threshold closer to the fixed threshold".

4.3.3.2 Step Module

Now that the overall architecture is introduced, we may now focus on building individual modules for each ECG interpretation step.

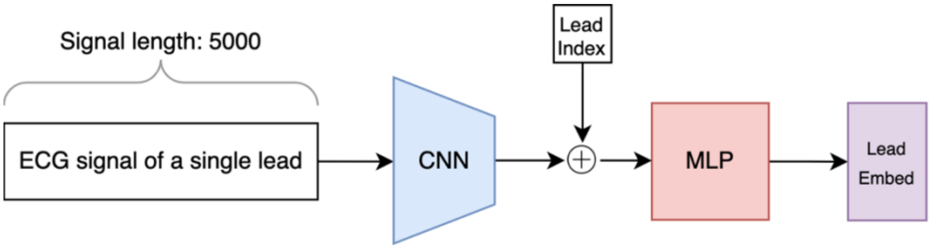


Figure 15 Module for Step 0 ECG Embedding

The "Step 0: ECG Embedding" Module, presented in Figure 15, is markedly different from other Step Modules as its sole responsibility is to extract a compressed representation of the input ECG record. Although it's quite similar to the baseline CNN, an important distinction here is that the input is a single lead of the ECG record. The lead signal will first go to a CNN block that accepts 1-channel inputs, the result of which is then concatenated with the lead index and further fed into an MLP to create the embedding for this lead signal. The motivation behind extracting embeddings for each lead individually is that later modules can get embeddings only for the leads that they should focus on. This is of great significance to the explainability as the doctors will only focus on certain leads at a specific step. If all later Step Modules use an overall

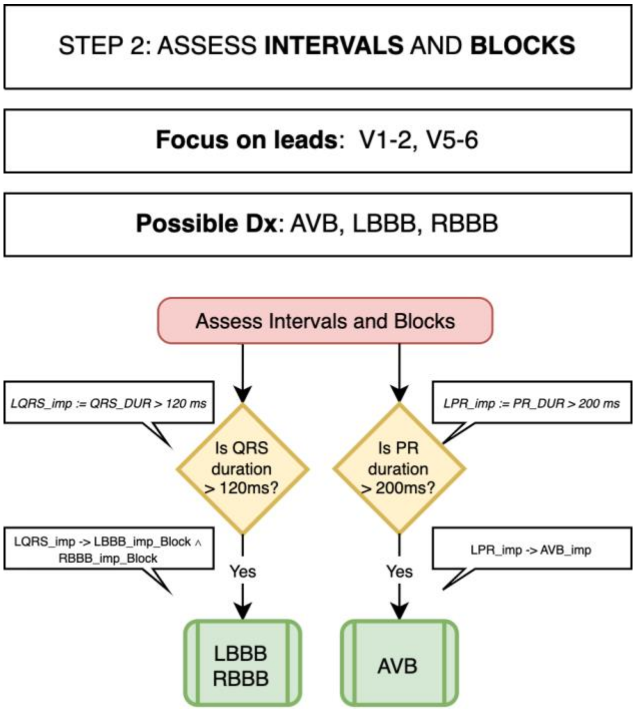

STEP 2: ASSESS **INTERVALS** AND **BLOCKS**
Focus on leads: V1-2, V5-6
Possible Dx: AVB, LBBB, RBBB
Assess Intervals and Blocks
LQRS_imp := QRS_DUR > 120 ms
Is QRS duration > 120ms?
LPR_imp := PR_DUR > 200 ms
Is PR duration > 200ms?
LQRS_imp -> LBBB_imp_Block ∧ RBBB_imp_Block
Yes
Yes
LPR_imp -> AVB_imp
LBBB RBBB
AVB

Figure 16 Step 2 of ECG Interpretation

embedding that captures information from all leads, then those Step Modules may cheat by looking at information from other leads and reach conclusions that human doctors cannot comprehend. Additionally, we apply the same embedding extracting network to different leads instead of having an individual network for each lead. The intuition here is that the morphology of the cardiac cycle generally does not change drastically during the 10-second ECG recording period. Therefore, given sufficient tuning and training, the network presented in Figure 15 should be able to capture the difference between leads. In this way, the number of trainable parameters is reduced by a factor of 12, which will reduce overfitting and speed up training.

The Step Modules other than Step 0 are generally similar in terms of how to construct them using soft rules introduced earlier. We will first go through the meaning of symbols/shapes in flowcharts for ECG interpretation, using which one can easily translate the FOL rules into sub-modules of Step Modules. As illustrated in Figure 16, each step of ECG interpretation identifies the leads that the step should focus on, possible diagnoses to be made in this step, and corresponding flowcharts describing rules on how to perform DDx to make those diagnoses. The associated FOL rules are written in the textbox beside the flowcharts. Along the direction of flow, the flowchart will go through a series of diamond-shaped decision nodes and eventually reach possible diagnoses (green rounded rectangles) in the current step. If a decision node is based on the comparison (e.g., PR duration > 200ms), the node will calculate the corresponding feature impression using the soft threshold introduced in the “Formula with Comparison operators” section. The feature impressions have subscript “_imp” to differentiate them from the objective features. The green diagnoses node at the end of each flowchart also has a subscript “_imp” in the associated FOL rules to indicate that it is an impression of the diagnosis at that particular step. This subscript is to differentiate them from the final prediction of the diagnoses which are ensembles of diagnosis impressions at certain steps.

After obtaining a basic understanding of the flowchart, one can effortlessly create a Step Module using sub-modules provided by our framework, Take Step 2 shown in Figure 16 as an example. Step 2 involves two comparisons and two implications. Hence, we will have two comparison operator sub-modules (each with their trainable w and δ) and two implication sub-modules.

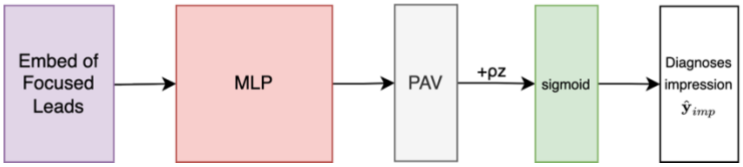


Figure 17 Implication sub-module using MPAV

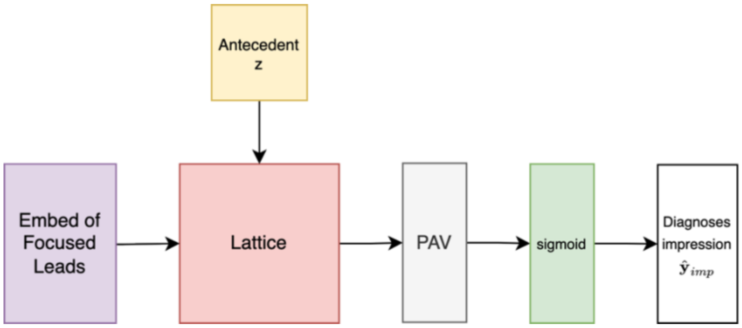


Figure 18 Implication sub-module using HL

Then, depending on the method used (either MPAV or HL), the implication sub-module will adopt one of the architectures shown in Figure 17 and Figure 18. Similarly, we can create other Step Modules according to the flowcharts in Appendix A3.

4.4 ECG-XAI framework

We have invested a significant amount of effort to ensure that our framework is both user-friendly and scalable.

To begin with, our framework provides users with the flexibility to customize targeted diagnoses, objective features, and modules for each step to suit their specific requirements. This enables them to easily add more diagnoses, objective features, or rules as needed.

Additionally, the Pipeline Module that encapsulates all Step Modules is designed with many functionalities targeting explainability. For example, users can specify which intermediate outputs to aggregate using the Pipeline Module, and those outputs will be saved as a CSV file at the end of the training process. The intermediate outputs that can be aggregated are not limited to feature impressions and implication statements' antecedents and consequents. Other terms, such as each Step Module's $\text{Loss}_{\text{feat}}$ and Loss_{δ}, as well as comparison operators' w and δ, can also be logged. In addition to aggregating intermediate output, users can require the Pipeline Module to plot one intermediate output against another when the training ends. Another crucial explainability functionality of the Pipeline Module is its ability to generate diagnosis reports. When given an input ECG record, the Pipeline Module processes it and generates a Markdown document containing the corresponding differential diagnoses list, along with explanations of how it was derived. With these functionalities of the Pipeline Module, users can easily verify whether the rules incorporated are working as expected during training. Once the training is complete, users may examine the aggregated intermediate outputs and the diagnosis report to ensure that the feature impressions, diagnosis impressions, and explanations are meaningful and make sense.

Last but not least, our framework is designed to be scalable from the ground up, starting with the creation of classes that can be applied to a variety of medical waveform signals. For instance, the 'Ecg' class inherits from the more general 'Signal' class, which encapsulates functionalities applicable to not just ECG, but also other types of signals. This approach allows us to effortlessly extend the framework to include other medical waveform signals whose rules can be represented in our system, without the need to repeat functionalities like caching preprocessed signals for each signal type. Another example of scalability is the Rule class, which serves as the root class for all rule classes, including FOL rules. While our framework currently supports rules expressible in FOL, we can expand it to include other types of rules, such as complex shape constraints.

5. Experiments

In this section, we will compare the different architectures listed in the “Methodology” section and explore the effect of MPAV’s ρ. Subsequently, the best model will be tested on the test set to inspect its generalizability. Moreover, at the end of this section, we will examine the diagnoses report created for an ECG record in the test set to verify that the generated feature impressions and explanation along the DDx process make sense to cardiologists.

As aforementioned, the dataset used is PTB-XL. The train, validation, and test sets were derived from stratified samples of the PTB-XL dataset, with an approximate ratio of 6:2:2 (for the number of ECG records in each set). The prediction task performed was a multi-label task and a total of 21 diagnoses were considered, whose details can be found in Appendix A1. The optimizer used was Adam with an exponential learning rate schedular. Each model had its hyperparameter tuned by the Optuna framework (Akiba et al., 2019), the details of which can be found in Appendix A4. Then the models with the best-performing hyperparameter configurations were tested on the validation set and compared with each other. The evaluation metrics used were accuracy (ACC) and macro-averaged Area Under the Receiver Operating Curve (AUROC). Moreover, it should be pointed out that the weight constants α and β for the loss of the soft rule system were not automatically tuned as the tuning framework might set them to zero or close to zero to maximize the performance, in which case the explainability would be heavily impaired. Instead, the α and β were grid searched, each in the set {0.01, 0.1, 1, 10, 100}. A good configuration that strikes a balance between performance and explainability was found to be α = 0.1 and β = 10, which was adopted in the soft rule systems for the following experiments.

5.1 Compare Four Architectures

Architecture	ACC	AUROC
Baseline CNN	0.9163	0.7926
Soft Rule System with MPAV	**0.9202**	**0.8360**
Soft Rule System with HL	0.9132	0.7173
Hard Rule System	0.8471	0.6829

Table 4 Performances of the four architectures

We started by comparing four architectures: baseline CNN, soft rule system with MPAV, soft rule system with HL, and hard rule system. The results of this experiment are collected in Table 4.

Overall, the soft rule system that incorporated MPAV demonstrated the highest level of performance. It had significant improvements over the hard rule system, and it outperformed the baseline CNN. This observation highlights that the integration of FOL not only enhances the interpretability of the system's predictions but also contributes to the system's overall performance.

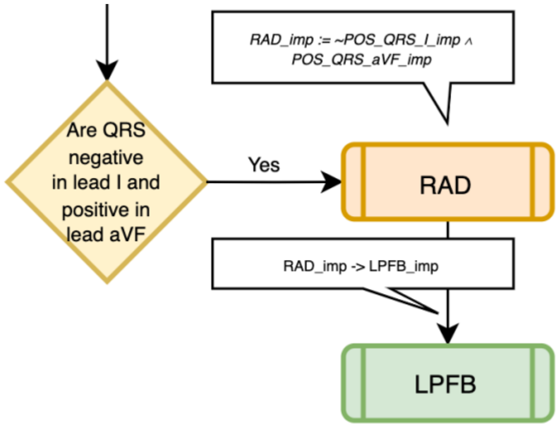

RAD_imp := ~POS_QRS_I_imp ∧ POS_QRS_aVF_imp
Are QRS negative in lead I and positive in lead aVF
Yes
RAD
RAD_imp -> LPFB_imp
LPFB

Figure 19 Part of Step 9's (Axis Module) flowchart

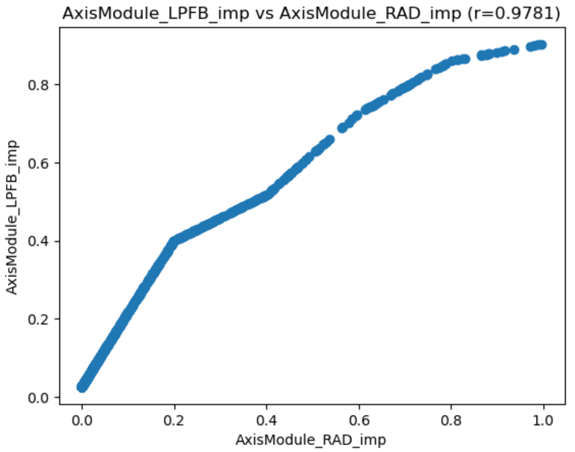


Figure 20 LPFB$_{imp}$ vs RAD$_{imp}$ (HL-version system)

We can interpret their performances in terms of the model's flexibility. On one hand, the hard rule system has very few trainable parameters and is nearly inflexible, which may hinder its ability to effectively fit the training set. On the other hand, the baseline CNN may be too flexible as it lacks domain knowledge and has not been regularized using ECG rules. As a result, the search space for the baseline model is effectively larger than that of the soft rule system, making it more challenging for the baseline model to identify optimal model weights without guidance from the ECG rules.

In addition, it was observed that the HL-version soft rule system's performance was even worse than the baseline. To verify that the lattice layer was functioning as expected, the model with HL was tested on the validation set, and the system's impression of antecedents and consequents of implication statements were aggregated and recorded. Take the implication "RAD$_{imp}$ → LPFB$_{imp}$" in step 9 (Axis Module) as an example, whose corresponding portion in the flowchart is extracted from Appendix A3 and presented in Figure 19 for ease of reference. By plotting the

HL-version system's impression of the consequent (i.e., LPFB_{imp}) against the impression of the antecedent (i.e., RAD_{imp}) in Figure 20, we can see that the consequent impression is indeed monotonically increasing with respect to the antecedent impression. However, if we plot the validation set's ground truth labeling of LPFB against the objective feature RAD calculated using fixed thresholds, we will notice that the correlation is weak, and it does not strictly follow a monotone relationship as shown in Figure 21. A reason for this is that the criteria involving cardiac axis deviation such as RAD are only ancillary, and the main criteria for LPFB cannot be easily encoded using FOL as it involves complex shape constraints on the QRS complex. Consequently, although RAD is suggestive of LPFB, the presence of RAD alone is not sufficient for a definitive LPFB diagnosis. Therefore, the relationship between RAD and LPFB may not be monotone since patients with a high likelihood of RAD may not meet the main shape criteria for LPFB. This may also explain the poor performance of the hard rule system as its implications are a special case of monotone function where the consequent is equal to the antecedent (e.g., $\text{LPFB}_{\text{imp}} = \text{RAD}_{\text{imp}}$ in the hard rule system)

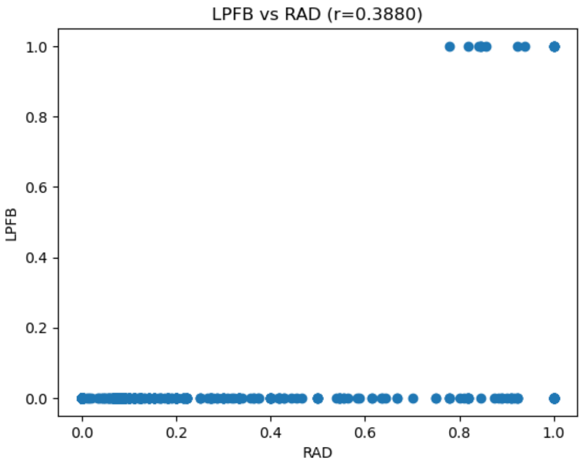


Figure 21 Ground truth LPFB vs objective RAD

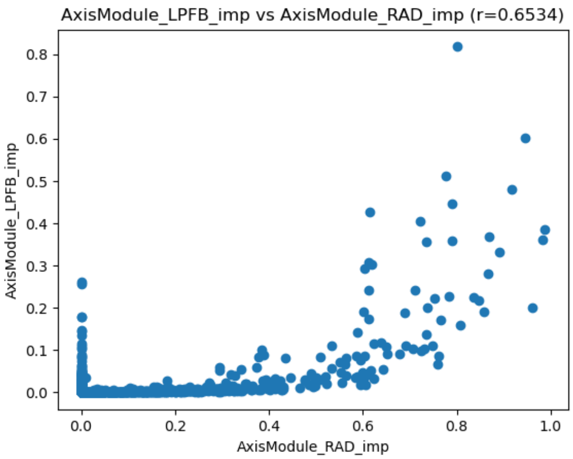


Figure 22 LPFB_{imp} vs RAD_{imp} (MPAV-version soft rule system with $\rho = \mathbf{8}$)

In contrast, if the soft rule system uses MPAV, the implication statement serves more as a suggestion and the system need not enforce a monotone relationship. In such cases, the relationship between the LPFB_{imp} and the RAD_{imp} is depicted in Figure 22. In a way, the MPAV system can automatically fill in the gaps when some diagnosis criteria/rules are not provided. Meanwhile, for rules that are fed into the system, the MPAV method can provide guidance on the prediction according to those rules.

Another observation that can be made is regarding the relative importance of diagnosis impressions from different modules for the same diagnosis. For instance, in the case of inferior MI (IMI), we can examine the absolute weights of the IMI ensemble linear layer, which ultimately produces the IMI prediction by combining the diagnosis impressions obtained according to ST elevation (STE), ST depression (STD), pathological Q wave, and inverted T wave. The respective absolute weight is 2.331, 1.536, 0.772, and 0.532. By dividing these values by their sum, one can get the relative importance of each diagnosis impression: 0.451, 0.297, 0.149, and 0.103. This aligns with the common ECG DDx practice as the STE is considered the most significant evidence of MI, followed by STD and pathological Q, while the inverted T does not always appear in patients with MI and therefore has a relatively small weight.

5.2 ρ in MPAV

ρ value	ACC	AUROC
0	0.9126	0.7778
2	0.9106	0.7968
4	0.9131	0.8086
8	**0.9202**	**0.8360**
16	0.9195	0.8298
32	0.9125	0.7368

Table 5 Performances of MPAV Systems with Different ρ

Since the MPAV system is the best-performing architecture, we will further explore the role of the scaling factor ρ in MPAV. In this experiment, ρ was grid searched in {0, 2, 4, 8, 16, 32} and the corresponding MPAV system's performances are encapsulated in Table 5.

It is evident that the ρ = 8 performed the best. Meanwhile, it is noteworthy that we had undesirable performances of MPAV when ρ was either too small (e.g., ρ = 0) or too large (e.g., ρ = 32).

On one hand, if ρ is too small, little suggestion is provided by the implication rule and the MPAV system is not leveraging the domain knowledge sufficiently. As a result, the performance may be even worse than the baseline CNN. In some sense, MPAV with very small ρ such as 0 is introducing extra structures while failing to harness the benefits provided by the extra structures (i.e., the guidance provided by the FOL). Thus, it is not a surprise to see that very small ρ results in poor performance.

On the other hand, if ρ is too large, the impression of a consequent will be nearly 1 even for a small truth value of the matching antecedent. To illustrate, we will again plot $\mathrm{LPFB}_{\mathrm{imp}}$ against $\mathrm{RAD}_{\mathrm{imp}}$ for a large ρ like 32 and the result is shown in Figure 23. This plot reveals that the modification of PAV is too drastic when ρ is set to 32, as compared to the plot generated when ρ = 8, shown in Figure 22. In such cases, the MPAV system loses useful information contained in antecedent impressions, as most consequent impressions are close to 1 due to the overly large value of ρ.

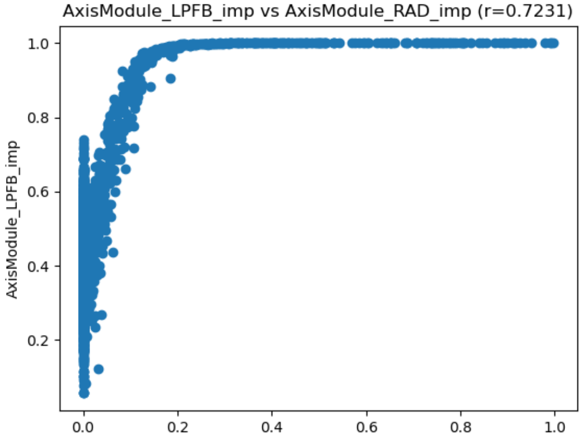


*Figure 23 LPFB_{imp} vs RAD_{imp} (MPAV-version soft rule system with ρ = **32**)*

Therefore, it is essential to identify an appropriate value of ρ that injects the right amount of domain knowledge into the model.

5.3 Model Inspection on the Test Set

To determine whether the best model selected (i.e., MPAV system with ρ = 8) during the training process can perform well on new, unseen data, the model was evaluated on the test set. The test set was not used in any way to train or adjust the model, ensuring that the evaluation provides a reliable measure of the model's generalization ability. The evaluation revealed that the model performed well on the test set, achieving an accuracy of 0.9157 and an AUROC of 0.8047. These results indicate that the performance of the model did not drop significantly when evaluated on unseen data, suggesting that the model has good generalization ability.

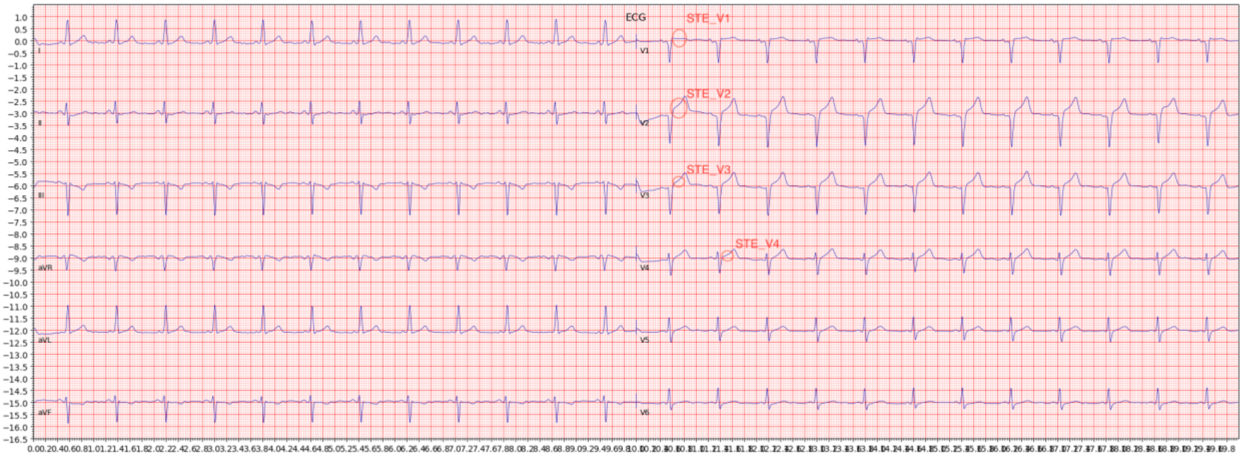


Figure 24 12-lead plot for an ECG record with SR and AMI

Moreover, to ensure that the explanations generated by our ECG-XAI system are meaningful to doctors, a diagnosis report is generated for a test set ECG record with diagnosis labels sinus rhythm (SR) and anterior MI (AMI). The ECG record's 12-lead plot is shown in Figure 24, and the detailed diagnosis report can be found in Appendix A5. According to the report, the system's prediction of SR and AMI are 1.000 and 0.997 respectively. Although they closely match the ground truth labels, we will explore the report and inspect whether the explanations that lead to these conclusions make sense.

For SR, we will take a look at Step 1's report, which focuses on assessing the rhythm and heart rate of the ECG record. Following the flowchart for Step 1, we should first check whether the rhythm is sinus[8]. This is in fact the case as shown in lead II. Hence, we rule out AFIB and AFLT. Moreover, ARRH[9] is not observed as the R-R intervals are generally consistent. Hence SARRH should be excluded. The generated report confirms that the system has successfully ruled out AFIB, AFLT, and SARRH, using the correctly calculated features. Furthermore, the heart rate can be estimated by multiple 6 to the number of cycles in a lead, as the recording is 10 seconds long. In the case of this ECG record, there are 12 cardiac cycles during this 10-second period and the estimated heart rate is 72 bpm which closely matches the system's calculated heart rate of

[8] Each P wave in lead II is positive AND precedes a QRS complex
[9] max R-R interval – min R-R interval > 120ms

71.813 bpm. Since the heart rate falls within the range of 60 to 100 bpm, we can conclude that the patient has SR, which matches the result in the diagnosis report.

While diagnoses of SR are generally straightforward, diagnoses for MI such as AMI are challenging as there are multiple factors to consider. The hallmark of the AMI is ST segment elevations (STE) in at least two contiguous[10] precordial leads (V1 to V6). The diagnosis report shows that the system's impressions for STEs in leads V1 to V6 are 0.718, 0.868, 0.799, 0.641, 0.253, and 0.254, respectively. We can confirm this by examining the ST segments in precordial leads in Figure 24, where STEs are observed in V1 to V4. In addition to STE, ST segment depression (STD), pathological Q wave, and inverted T wave are the three ancillary features of AMI. The diagnosis report shows that the system's corresponding feature impressions for these features are generally small (i.e., less than 0.5 and close to 0), which aligns with what we can observe in Figure 24, where there is no evident STD, pathological Q wave, or inverted T wave.

In short, our system shows great generalizability, and the above case study of a test set ECG record demonstrated our system's capability to provide comprehensible explanations for its generated differential diagnoses list.

[10] contiguous leads are next to each other anatomically (e.g., V1 and V2)

6. Conclusion

In this paper, we have developed an XAI framework for incorporating ECG rules in the form of FOL into ECG AI models. Although the current framework only supports FOL rules, the framework is designed with scalability in mind and can be efficiently extended to include other types of constraints and rules. With the concept of feature impression, we can reveal the underlying ML model’s understanding of explainable ECG features. Additionally, using probabilistic soft logic and logical connectives outlined in the “Methodology” section, feature impressions can be further combined to create other interpretable features.

Our experiments showcase the benefits of incorporating ECG rules into our system. The first experiment illustrates how the inclusion of these rules can enhance model performance, and it also demonstrates the system's ability to automatically fill in gaps in the rules when rules cannot be provided in the form of FOL. Moreover, the second experiment emphasizes the importance of controlling the amount of domain knowledge injected into the system. Furthermore, our system's test set performance highlights its great generalizability. The generated diagnosis report provides valuable insights into the model's decision-making process, which is beneficial for cardiologists to interpret the diagnoses predicted by our ECG AI.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework* (arXiv:1907.10902). arXiv. https://doi.org/10.48550/arXiv.1907.10902

Anand, A., Kadian, T., Shetty, M. K., & Gupta, A. (2022). Explainable AI decision model for ECG data of cardiac disorders. *Biomedical Signal Processing and Control*, *75*, 103584. https://doi.org/10.1016/j.bspc.2022.103584

Ashley, E. A., & Niebauer, J. (2004). Conquering the ECG. In *Cardiology Explained*. Remedica. http://www.ncbi.nlm.nih.gov/books/NBK2214/

Beltagy, I., Erk, K., & Mooney, R. (2014). Probabilistic Soft Logic for Semantic Textual Similarity. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1210–1219. https://doi.org/10.3115/v1/P14-1114

Besold, T. R., Garcez, A. d'Avila, Bader, S., Bowman, H., Domingos, P., Hitzler, P., Kuehnberger, K.-U., Lamb, L. C., Lowd, D., Lima, P. M. V., de Penning, L., Pinkas, G., Poon, H., & Zaverucha, G. (2017). *Neural-Symbolic Learning and Reasoning: A Survey and Interpretation* (arXiv:1711.03902). arXiv. https://doi.org/10.48550/arXiv.1711.03902

Du, X., Dalvi, B., Tandon, N., Bosselut, A., Yih, W., Clark, P., & Cardie, C. (2019). Be Consistent! Improving Procedural Text Comprehension using Label Consistency. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2347–2356. https://doi.org/10.18653/v1/N19-1244

Gupta, M., Cotter, A., Pfeifer, J., Voevodski, K., Canini, K., Mangylov, A., Moczydlowski, W., & Esbroeck, A. van. (2016). Monotonic Calibrated Interpolated Look-Up Tables. *Journal of Machine Learning Research*, *17*(109), 1–47.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2018). *Bag of Tricks for Image Classification with Convolutional Neural Networks* (arXiv:1812.01187). arXiv. https://doi.org/10.48550/arXiv.1812.01187

Hegadi, R. (2014). *A Literature Review on Approaches of ECG Pattern Recognition*. https://www.academia.edu/49169207/A_Literature_Review_on_Approaches_of_ECG_Pattern_Recognition

Hicks, S. A., Isaksen, J. L., Thambawita, V., Ghouse, J., Ahlberg, G., Linneberg, A., Grarup, N., Strümke, I., Ellervik, C., Olesen, M. S., Hansen, T., Graff, C., Holstein-Rathlou, N.-H., Halvorsen, P., Maleckar, M. M., Riegler, M. A., & Kanters, J. K. (2021). Explaining deep neural networks for knowledge discovery in electrocardiogram analysis. *Scientific Reports*, *11*(1), Article 1. https://doi.org/10.1038/s41598-021-90285-5

Hu, Z., Ma, X., Liu, Z., Hovy, E., & Xing, E. (2020). *Harnessing Deep Neural Networks with Logic Rules* (arXiv:1603.06318). arXiv. https://doi.org/10.48550/arXiv.1603.06318

Hughes, J. W., Olgin, J. E., Avram, R., Abreau, S. A., Sittler, T., Radia, K., Hsia, H., Walters, T., Lee, B., Gonzalez, J. E., & Tison, G. H. (2021). Performance of a Convolutional Neural Network and Explainability Technique for 12-Lead Electrocardiogram Interpretation. *JAMA Cardiology*, *6*(11), 1285–1295. https://doi.org/10.1001/jamacardio.2021.2746

Jahmunah, V., Ng, E. Y. K., San, T. R., & Acharya, U. R. (2021). Automated detection of coronary artery disease, myocardial infarction and congestive heart failure using

GaborCNN model with ECG signals. *Computers in Biology and Medicine*, *134*, 104457. https://doi.org/10.1016/j.compbiomed.2021.104457

Jo, Y.-Y., Cho, Y., Lee, S. Y., Kwon, J., Kim, K.-H., Jeon, K.-H., Cho, S., Park, J., & Oh, B.-H. (2021). Explainable artificial intelligence to detect atrial fibrillation using electrocardiogram. *International Journal of Cardiology*, *328*, 104–110. https://doi.org/10.1016/j.ijcard.2020.11.053

Kalyakulina, A. I., Yusipov, I. I., Moskalenko, V. A., Nikolskiy, A. V., Kosonogov, K. A., Osipov, G. V., Zolotykh, N. Y., & Ivanchenko, M. V. (2020). *LUDB: A new open-access validation tool for electrocardiogram delineation algorithms* (arXiv:1809.03393). arXiv. https://doi.org/10.48550/arXiv.1809.03393

Khan, M. G. (Ed.). (2008). *Rapid ECG Interpretation*. Humana Press. https://doi.org/10.1007/978-1-59745-408-7

Li, T., & Srikumar, V. (2019). Augmenting Neural Networks with First-order Logic. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 292–302. https://doi.org/10.18653/v1/P19-1028

Lih, O. S., Jahmunah, V., San, T. R., Ciaccio, E. J., Yamakawa, T., Tanabe, M., Kobayashi, M., Faust, O., & Acharya, U. R. (2020). Comprehensive electrocardiographic diagnosis based on deep learning. *Artificial Intelligence in Medicine*, *103*, 101789. https://doi.org/10.1016/j.artmed.2019.101789

Lilly, L. S. (2012). *Pathophysiology of Heart Disease: A Collaborative Project of Medical Students and Faculty*. Lippincott Williams & Wilkins.

Liu, H., Chen, D., Chen, D., Zhang, X., Li, H., Bian, L., Shu, M., & Wang, Y. (2022). A large-scale multi-label 12-lead electrocardiogram database with standardized diagnostic statements. *Scientific Data*, *9*(1), Article 1. https://doi.org/10.1038/s41597-022-01403-5

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., & Chen, S. H. A. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods*, *53*(4), 1689–1696. https://doi.org/10.3758/s13428-020-01516-y

Meek, S., & Morris, F. (2002). ABC of clinical electrocardiography.Introduction. I-Leads, rate, rhythm, and cardiac axis. *BMJ (Clinical Research Ed.)*, *324*(7334), 415–418. https://doi.org/10.1136/bmj.324.7334.415

Minervini, P., & Riedel, S. (2018). Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge. *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 65–74. https://doi.org/10.18653/v1/K18-1007

Parsi, A. (2021). *Improved Cardiac Arrhythmia Prediction Based on Heart Rate Variability Analysis*. https://doi.org/10.13140/RG.2.2.15748.40322

Raza, A., Tran, K. P., Koehl, L., & Li, S. (2022). Designing ECG monitoring healthcare system with federated transfer learning and explainable AI. *Knowledge-Based Systems*, *236*, 107763. https://doi.org/10.1016/j.knosys.2021.107763

Schrepel, C., Amick, A. E., Sayed, M., & Chipman, A. K. (2021). Ischemic ECG Pattern Recognition to Facilitate Interpretation While Task Switching: A Parallel Curriculum. *MedEdPORTAL: The Journal of Teaching and Learning Resources*, *17*, 11182. https://doi.org/10.15766/mep_2374-8265.11182

Siontis, K. C., Noseworthy, P. A., Attia, Z. I., & Friedman, P. A. (2021). Artificial intelligence-enhanced electrocardiography in cardiovascular disease management. *Nature Reviews Cardiology*, *18*(7), Article 7. https://doi.org/10.1038/s41569-020-00503-2

Somani, S., Russak, A. J., Richter, F., Zhao, S., Vaid, A., Chaudhry, F., De Freitas, J. K., Naik, N., Miotto, R., Nadkarni, G. N., Narula, J., Argulian, E., & Glicksberg, B. S. (2021). Deep learning and the electrocardiogram: Review of the current state-of-the-art. *Europace*, *23*(8), 1179–1191. https://doi.org/10.1093/europace/euaa377

Strodthoff, N., Wagner, P., Schaeffter, T., & Samek, W. (2020). *Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL* (arXiv:2004.13701). arXiv. https://doi.org/10.48550/arXiv.2004.13701

Surawicz, B., & Knilans, T. (2008). *Chou's Electrocardiography in Clinical Practice: Adult and Pediatric*. Elsevier Health Sciences.

Taniguchi, H., Takata, T., Takechi, M., Furukawa, A., Iwasawa, J., Kawamura, A., Taniguchi, T., & Tamura, Y. (2021). Explainable Artificial Intelligence Model for Diagnosis of Atrial Fibrillation Using Holter Electrocardiogram Waveforms. *International Heart Journal*, *62*(3), 534–539. https://doi.org/10.1536/ihj.21-094

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, *3*(2), 334–340. https://doi.org/10.1016/j.fcij.2018.10.003

Teodoridis, A. G. and F. (2022, March 9). Why is AI adoption in health care lagging? *Brookings*. https://www.brookings.edu/research/why-is-ai-adoption-in-health-care-lagging/

Timmis, A., Townsend, N., Gale, C. P., Torbica, A., Lettino, M., Petersen, S. E., Mossialos, E. A., Maggioni, A. P., Kazakiewicz, D., May, H. T., De Smedt, D., Flather, M., Zuhlke, L., Beltrame, J. F., Huculeci, R., Tavazzi, L., Hindricks, G., Bax, J., Casadei, B., … European Society of Cardiology. (2020). European Society of Cardiology: Cardiovascular Disease Statistics 2019. *European Heart Journal, 41*(1), 12–85. https://doi.org/10.1093/eurheartj/ehz859

Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., & Schaeffter, T. (2020). PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, *7*(1), Article 1. https://doi.org/10.1038/s41597-020-0495-6

Wang, Z., Yan, W., & Oates, T. (2017). *Time series classification from scratch with deep neural networks: A strong baseline*. 1578–1585. https://doi.org/10.1109/IJCNN.2017.7966039

WHO. (n.d.). *Cardiovascular diseases (CVDs)*. Retrieved October 30, 2022, from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Xu, J., Zhang, Z., Friedman, T., Liang, Y., & Broeck, G. V. den. (2018). *A Semantic Loss Function for Deep Learning with Symbolic Knowledge* (arXiv:1711.11157). arXiv. https://doi.org/10.48550/arXiv.1711.11157

Yanagisawa, H., Miyaguchi, K., & Katsuki, T. (2022). Hierarchical Lattice Layer for Partially Monotone Neural Networks. *Advances in Neural Information Processing Systems*, *35*, 11092–11103.

Zhang, D., Yang, S., Yuan, X., & Zhang, P. (2021). Interpretable deep learning for automatic diagnosis of 12-lead electrocardiogram. *IScience, 24*(4), 102373. https://doi.org/10.1016/j.isci.2021.102373

Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., & Rakovski, C. (2020). A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific Data*, *7*(1), 48. https://doi.org/10.1038/s41597-020-0386-x

Appendix

A1 List of Diagnoses

The current ECG-XAI framework can predict a wide range of diagnoses summarized in the following table. Moreover, the table also shows the diagnoses' abbreviations, superclass, and the number of training ECG records with those diagnoses. The definition of each diagnosis superclass and its corresponding abbreviations can be found in Table 2 and Table 3, respectively.

Diagnosis Name	Abbreviation	Diagnosis Superclass	# Training Records
Normal	NORM	NORM	4896
Sinus Arrhythmia	SARRH		383
Sinus Bradycardia	SBRAD		279
Sinus Rhythm	SR		8070
Sinus Tachycardia	STACH		276
Atrial Fibrillation	AFIB	ARR	580
Atrial Flutter	AFLT		12
1st Degree AV Block	AVB	CD	305
Intraventricular Conduction Disturbance	IVCD		297
Left Anterior Fascicular Block	LAFB		703
Left Bundle Branch Block	LBBB		82
Left Posterior Fascicular Block	LPFB		61
Right Bundle Branch Block	RBBB		678
Wolff-Parkinson-White syndrome	WPW		16
Left Atrial Enlargement	LAE	HYP	155
Left Ventricular Hypertrophy	LVH		721
Right Atrial Enlargement	RAE		41
Right Ventricular Hypertrophy	RVH		41
Anterior Myocardial Infarction	AMI	MI/ISC	1185
Inferior Myocardial Infarction	IMI		1493
Lateral Myocardial Infarction	LMI		473

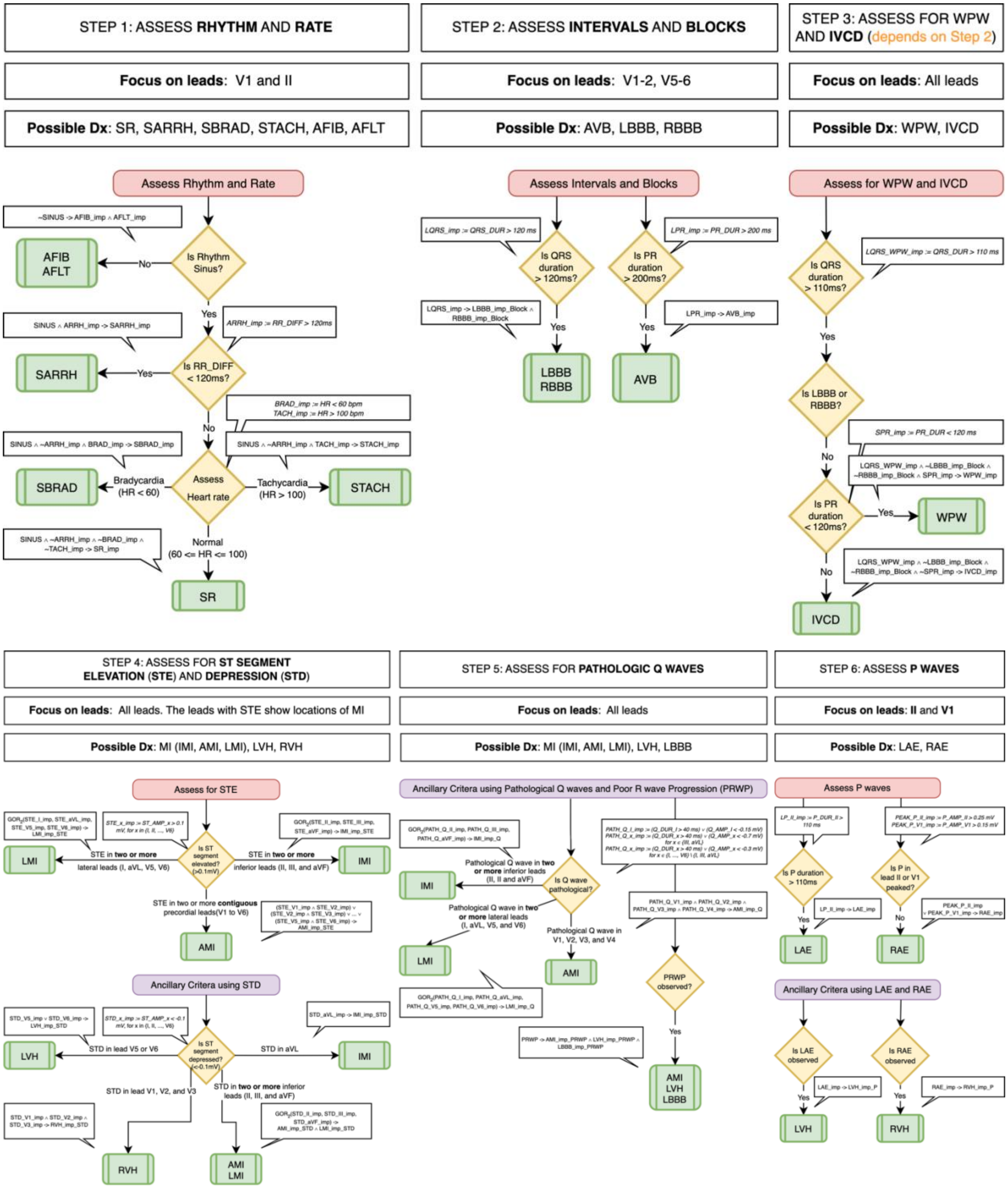
A2 Objective Features

All objective features used in this project, their abbreviations, and their explanations are encapsulated in the table below. If x appeared in the table, it refers to one of the leads (i.e., x ∊ {I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6}).

Feature Name	Abbreviation	Explanation
Heart Rate	HR	Heart Rate of the patient
Bradycardia	BRAD	Whether the patient has bradycardia (HR < 60 bpm)
Tachycardia	TACH	Whether the patient has tachycardia (HR > 100 bpm)
Sinus	SINUS	Whether the rhythm is sinus: Each P wave in lead II should be positive AND precedes a QRS complex
RR interval range	RR_DIFF	max R-R interval – min R-R interval
PR duration	PR_DUR	Duration of the PR segment
Prolonged PR	LPR	Whether the PR interval is prolonged
QRS duration	QRS_DUR	Duration of the QRS complex
Prolonged QRS	LQRS	Whether the QRS complex is prolonged
Prolonged QRS for WPW	LQRS_WPW	Whether the QRS complex is prolonged by WPW's standards
Short PR	SPR	Whether the PR interval is shortened
ST segment amplitude	ST_AMP_x	Mean Amplitude of ST segment in lead x
ST Elevation	STE_x	Whether the ST segment is elevated in lead x
ST Depression	STD_x	Whether the ST segment is depressed in lead x
Poor R-wave Progression	PRWP	Whether R waves are not within desired ranges for at least one lead in V1-V4
Q wave duration	Q_DUR_x	Duration of the QRS complex in lead x
Q wave amplitude	Q_AMP_x	Amplitude of Q wave in lead x
Pathological Q wave	PATH_Q_x	Whether the Q wave in lead x is pathological
P wave duration	P_DUR_x	Duration of P wave in lead x
P wave amplitude	P_AMP_x	Amplitude of P wave in lead x
Prolonged P wave	LP_x	Whether the P wave is prolonged in lead x
Peaked P wave	PEAK_P_x	Whether the P wave is peaked (has high amplitude) in lead x
Age	AGE	Age of the patient
Old age	AGE_OLD	Whether the patient's age is greater than 30
Male	MALE	Whether the patient is male
R wave amplitude	R_AMP_x	Amplitude of R wave in lead x
S wave amplitude	S_AMP_x	Amplitude of S wave in lead x
R/S Ratio	RS_RATIO_x	Ratio between amplitudes of R and S waves in lead x
Peaked R wave	PEAK_R_x	Whether the R wave is peaked in lead x
Deep S wave	DEEP_S_x	Whether the S wave is deep (has low amplitude) in lead x
Dominant R wave	DOM_R_x	Whether R wave amplitude is greater than that of the S wave
Dominant S wave	DOM_S_x	Whether S wave amplitude is greater than that of the R wave
T wave amplitude	T_AMP_x	Amplitude of T wave in lead x
Inverted T wave	INVT_x	Whether the T wave is inverted in lead x
Sum of QRS	QRS_SUM_x	The QRS area above the baseline minus the QRS area below
Postive QRS	POS_QRS_x	The QRS is positive in lead x
Normal cardiac axis	NORM_AXIS	Whether the patient has a normal cardiac axis
Left axis deviation	LAD	Whether the patient's cardiac axis deviates towards the left
Right axis deviation	RAD	Whether the patient's cardiac axis deviates towards the right

A3 ECG Interpretation Flowchart

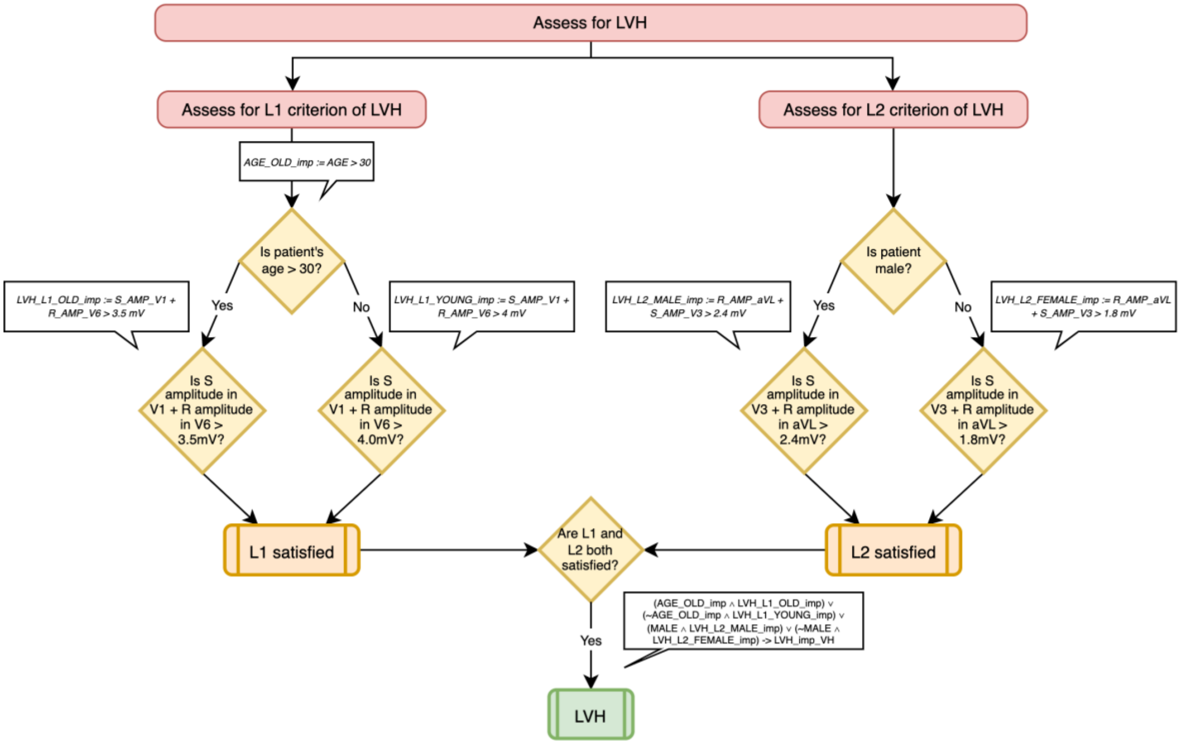
The flowcharts below summarize the 9 steps of ECG interpretation adapted from the book "Rapid ECG Interpretation" (Khan, 2008). While the meanings of most symbols/shapes are introduced in the "Step Module" section, there are some additional symbols/shapes to take note of. Firstly, a flowchart begins with a start node that is either red or purple. The red start node indicates that the following rules/decisions are parts of the main criteria of possible diagnoses and the purple one indicates that the following rules are ancillary criteria that should be used in combination with other main criteria. Moreover, we have some orange nodes which indicate some intermediate features that aid our diagnosis process (e.g., whether each of the 5 criteria of RVH is satisfied at step.

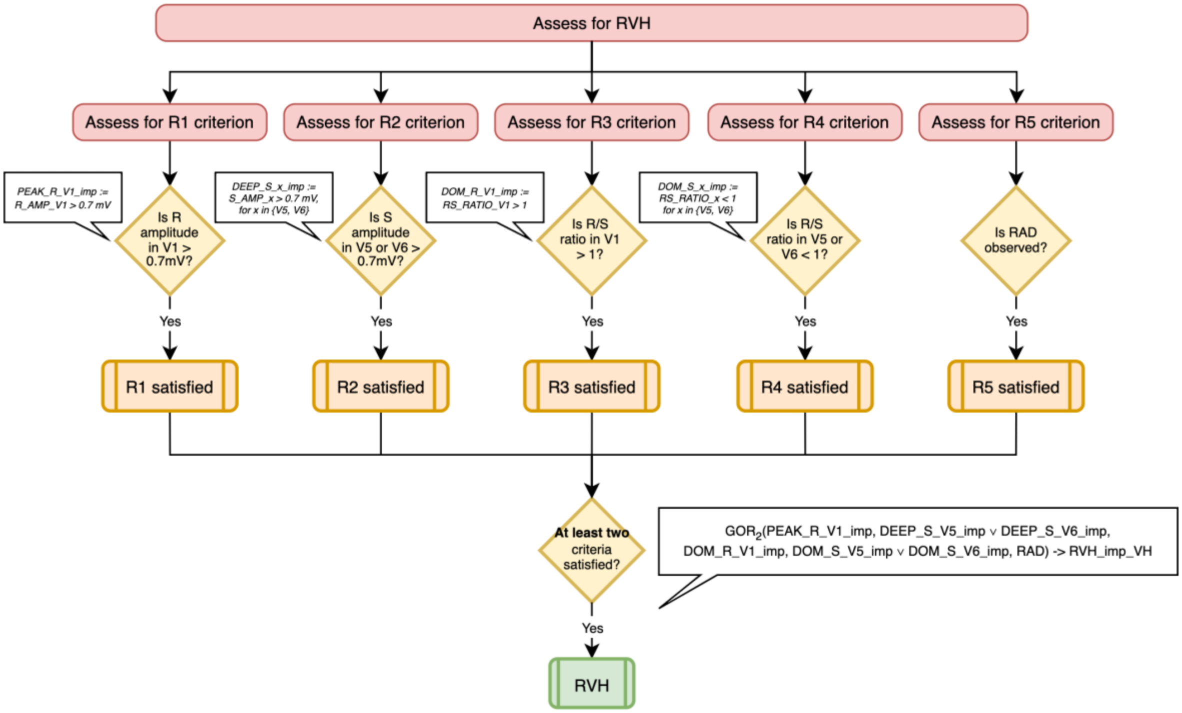

STEP 1: ASSESS RHYTHM AND RATE
Focus on leads: V1 and II
Possible Dx: SR, SARRH, SBRAD, STACH, AFIB, AFLT
Assess Rhythm and Rate
~SINUS -> AFIB_imp ∧ AFLT_imp
Is Rhythm Sinus?
No
AFIB AFLT
Yes
SINUS ∧ ARRH_imp -> SARRH_imp
ARRH_imp := RR_DIFF > 120ms
Is RR_DIFF < 120ms?
Yes
SARRH
No
BRAD_imp := HR < 60 bpm
TACH_imp := HR > 100 bpm
SINUS ∧ ~ARRH_imp ∧ BRAD_imp -> SBRAD_imp
SINUS ∧ ~ARRH_imp ∧ TACH_imp -> STACH_imp
Assess Heart rate
Bradycardia (HR < 60)
SBRAD
Tachycardia (HR > 100)
STACH
SINUS ∧ ~ARRH_imp ∧ ~BRAD_imp ∧ ~TACH_imp -> SR_imp
Normal (60 <= HR <= 100)
SR
STEP 2: ASSESS INTERVALS AND BLOCKS
Focus on leads: V1-2, V5-6
Possible Dx: AVB, LBBB, RBBB
Assess Intervals and Blocks
LQRS_imp := QRS_DUR > 120 ms
Is QRS duration > 120ms?
LQRS_imp -> LBBB_imp_Block ∧ RBBB_imp_Block
Yes
LBBB RBBB
LPR_imp := PR_DUR > 200 ms
Is PR duration > 200ms?
LPR_imp -> AVB_imp
Yes
AVB
STEP 3: ASSESS FOR WPW AND IVCD (depends on Step 2)
Focus on leads: All leads
Possible Dx: WPW, IVCD
Assess for WPW and IVCD
LQRS_WPW_imp := QRS_DUR > 110 ms
Is QRS duration > 110ms?
Yes
Is LBBB or RBBB?
No
SPR_imp := PR_DUR < 120 ms
LQRS_WPW_imp ∧ ~LBBB_imp_Block ∧ ~RBBB_imp_Block ∧ SPR_imp -> WPW_imp
Is PR duration < 120ms?
Yes
WPW
No
LQRS_WPW_imp ∧ ~LBBB_imp_Block ∧ ~RBBB_imp_Block ∧ ~SPR_imp -> IVCD_imp
IVCD
STEP 4: ASSESS FOR ST SEGMENT ELEVATION (STE) AND DEPRESSION (STD)
Focus on leads: All leads. The leads with STE show locations of MI
Possible Dx: MI (IMI, AMI, LMI), LVH, RVH
Assess for STE
GOR2(STE_I_imp, STE_aVL_imp, STE_V5_imp, STE_V6_imp) -> LMI_imp_STE
STE_x_imp := ST_AMP_x > 0.1 mV, for x in (I, II, ..., V6)
GOR2(STE_II_imp, STE_III_imp, STE_aVF_imp) -> IMI_imp_STE
LMI
STE in two or more lateral leads (I, aVL, V5, V6)
Is ST segment elevated? (>0.1mV)
STE in two or more inferior leads (II, III, and aVF)
IMI
STE in two or more contiguous precordial leads(V1 to V6)
(STE_V1_imp ∧ STE_V2_imp) ∨ (STE_V2_imp ∧ STE_V3_imp) ∨ ... ∨ (STE_V5_imp ∧ STE_V6_imp) -> AMI_imp_STE
AMI
Ancillary Critera using STD
STD_V5_imp ∨ STD_V6_imp -> LVH_imp_STD
STD_x_imp := ST_AMP_x < -0.1 mV, for x in (I, II, ..., V6)
STD_aVL_imp -> IMI_imp_STD
LVH
STD in lead V5 or V6
Is ST segment depressed? (<-0.1mV)
STD in aVL
IMI
STD in lead V1, V2, and V3
STD in two or more inferior leads (II, III, and aVF)
STD_V1_imp ∧ STD_V2_imp ∧ STD_V3_imp -> RVH_imp_STD
GOR2(STD_II_imp, STD_III_imp, STD_aVF_imp) -> AMI_imp_STD ∧ LMI_imp_STD
RVH
AMI LMI
STEP 5: ASSESS FOR PATHOLOGIC Q WAVES
Focus on leads: All leads
Possible Dx: MI (IMI, AMI, LMI), LVH, LBBB
Ancillary Critera using Pathological Q waves and Poor R wave Progression (PRWP)
GOR2(PATH_Q_II_imp, PATH_Q_III_imp, PATH_Q_aVF_imp) -> IMI_imp_Q
PATH_Q_I_imp := (Q_DUR_I > 40 ms) ∨ (Q_AMP_I < -0.15 mV)
PATH_Q_x_imp := (Q_DUR_x > 40 ms) ∨ (Q_AMP_x < -0.7 mV) for x ∈ (III, aVL)
PATH_Q_x_imp := (Q_DUR_x > 40 ms) ∨ (Q_AMP_x < -0.3 mV) for x ∈ (I, ..., V6) \ (I, III, aVL)
Pathological Q wave in two or more inferior leads (II, II and aVF)
IMI
Is Q wave pathological?
Pathological Q wave in two or more lateral leads (I, aVL, V5, and V6)
LMI
Pathological Q wave in V1, V2, V3, and V4
AMI
PATH_Q_V1_imp ∧ PATH_Q_V2_imp ∧ PATH_Q_V3_imp ∧ PATH_Q_V4_imp -> AMI_imp_Q
GOR2(PATH_Q_I_imp, PATH_Q_aVL_imp, PATH_Q_V5_imp, PATH_Q_V6_imp) -> LMI_imp_Q
PRWP observed?
Yes
PRWP -> AMI_imp_PRWP ∧ LVH_imp_PRWP ∧ LBBB_imp_PRWP
AMI LVH LBBB
STEP 6: ASSESS P WAVES
Focus on leads: II and V1
Possible Dx: LAE, RAE
Assess P waves
LP_II_imp := P_DUR_II > 110 ms
PEAK_P_II_imp := P_AMP_II > 0.25 mV
PEAK_P_V1_imp := P_AMP_V1 > 0.15 mV
Is P duration > 110ms
Is P in lead II or V1 peaked?
LP_II_imp -> LAE_imp
Yes
LAE
PEAK_P_II_imp ∨ PEAK_P_V1_imp -> RAE_imp
No
RAE
Ancillary Critera using LAE and RAE
Is LAE observed
LAE_imp -> LVH_imp_P
Yes
LVH
Is RAE observed
RAE_imp -> RVH_imp_P
Yes
RVH

STEP 7: ASSESS FOR **LVH/RVH**

Focus on leads: **V1-V6, aVL**

Possible Dx: LVH, RVH

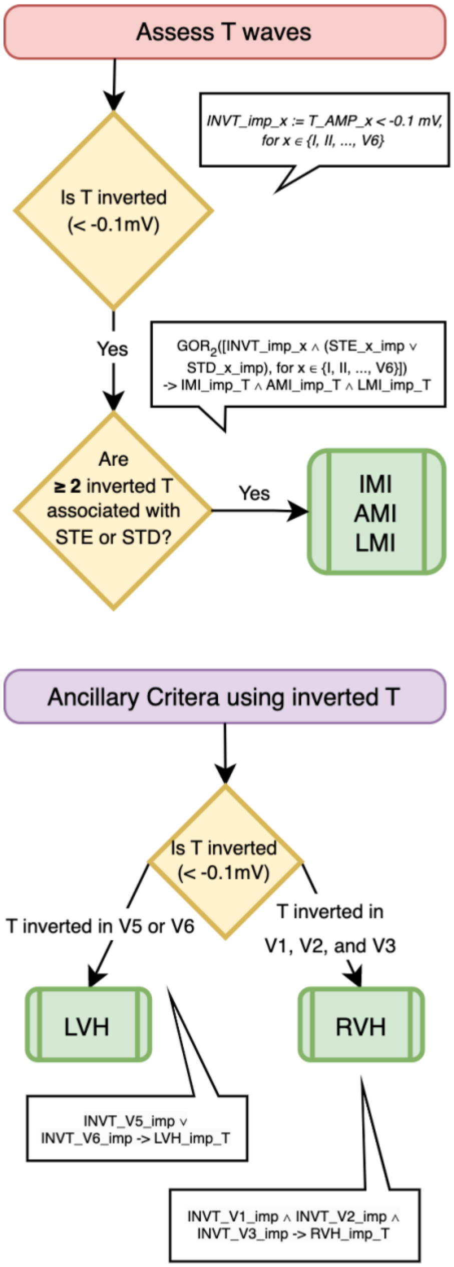

Assess for LVH
Assess for L1 criterion of LVH
Assess for L2 criterion of LVH
AGE_OLD_imp := AGE > 30
Is patient's age > 30?
Yes
No
LVH_L1_OLD_imp := S_AMP_V1 + R_AMP_V6 > 3.5 mV
LVH_L1_YOUNG_imp := S_AMP_V1 + R_AMP_V6 > 4 mV
Is S amplitude in V1 + R amplitude in V6 > 3.5mV?
Is S amplitude in V1 + R amplitude in V6 > 4.0mV?
Is patient male?
Yes
No
LVH_L2_MALE_imp := R_AMP_aVL + S_AMP_V3 > 2.4 mV
LVH_L2_FEMALE_imp := R_AMP_aVL + S_AMP_V3 > 1.8 mV
Is S amplitude in V3 + R amplitude in aVL > 2.4mV?
Is S amplitude in V3 + R amplitude in aVL > 1.8mV?
L1 satisfied
L2 satisfied
Are L1 and L2 both satisfied?
(AGE_OLD_imp ∧ LVH_L1_OLD_imp) ∨ (~AGE_OLD_imp ∧ LVH_L1_YOUNG_imp) ∨ (MALE ∧ LVH_L2_MALE_imp) ∨ (~MALE ∧ LVH_L2_FEMALE_imp) -> LVH_imp_VH
Yes
LVH


Assess for RVH
Assess for R1 criterion
Assess for R2 criterion
Assess for R3 criterion
Assess for R4 criterion
Assess for R5 criterion
PEAK_R_V1_imp := R_AMP_V1 > 0.7 mV
Is R amplitude in V1 > 0.7mV?
DEEP_S_x_imp := S_AMP_x > 0.7 mV, for x in {V5, V6}
Is S amplitude in V5 or V6 > 0.7mV?
DOM_R_V1_imp := RS_RATIO_V1 > 1
Is R/S ratio in V1 > 1?
DOM_S_x_imp := RS_RATIO_x < 1 for x in {V5, V6}
Is R/S ratio in V5 or V6 < 1?
Is RAD observed?
Yes
Yes
Yes
Yes
Yes
R1 satisfied
R2 satisfied
R3 satisfied
R4 satisfied
R5 satisfied
At least two criteria satisfied?
GOR2(PEAK_R_V1_imp, DEEP_S_V5_imp ∨ DEEP_S_V6_imp, DOM_R_V1_imp, DOM_S_V5_imp ∨ DOM_S_V6_imp, RAD) -> RVH_imp_VH
Yes
RVH

STEP 8: ASSESS **T WAVES** (depends on Step 4)

Focus on leads: **All leads**

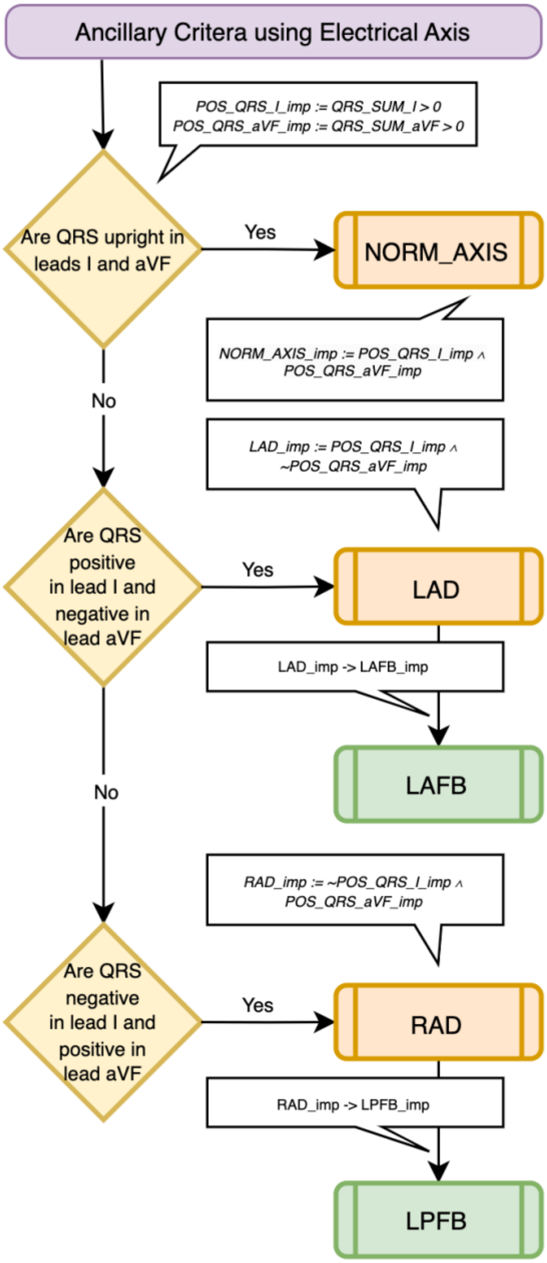
Possible Dx: MI, LVH, RVH


Assess T waves
INVT_imp_x := T_AMP_x < -0.1 mV, for x ∈ {I, II, ..., V6}
Is T inverted (< -0.1mV)
Yes
GOR$_2$([INVT_imp_x ∧ (STE_x_imp ∨ STD_x_imp), for x ∈ {I, II, ..., V6}]) -> IMI_imp_T ∧ AMI_imp_T ∧ LMI_imp_T
Are ≥ 2 inverted T associated with STE or STD?
Yes
IMI
AMI
LMI
Ancillary Critera using inverted T
Is T inverted (< -0.1mV)
T inverted in V5 or V6
T inverted in V1, V2, and V3
LVH
RVH
INVT_V5_imp ∨ INVT_V6_imp -> LVH_imp_T
INVT_V1_imp ∧ INVT_V2_imp ∧ INVT_V3_imp -> RVH_imp_T

STEP 9: ASSESS **ELECTRICAL AXIS**

Focus on leads: **I, aVF**

Possible Dx: LAFB, LPFB


Ancillary Critera using Electrical Axis
POS_QRS_I_imp := QRS_SUM_I > 0
POS_QRS_aVF_imp := QRS_SUM_aVF > 0
Are QRS upright in leads I and aVF
Yes
NORM_AXIS
NORM_AXIS_imp := POS_QRS_I_imp ∧ POS_QRS_aVF_imp
No
LAD_imp := POS_QRS_I_imp ∧ ~POS_QRS_aVF_imp
Are QRS positive in lead I and negative in lead aVF
Yes
LAD
LAD_imp -> LAFB_imp
LAFB
No
RAD_imp := ~POS_QRS_I_imp ∧ POS_QRS_aVF_imp
Are QRS negative in lead I and positive in lead aVF
Yes
RAD
RAD_imp -> LPFB_imp
LPFB

A4 Hyperparameter Tuning

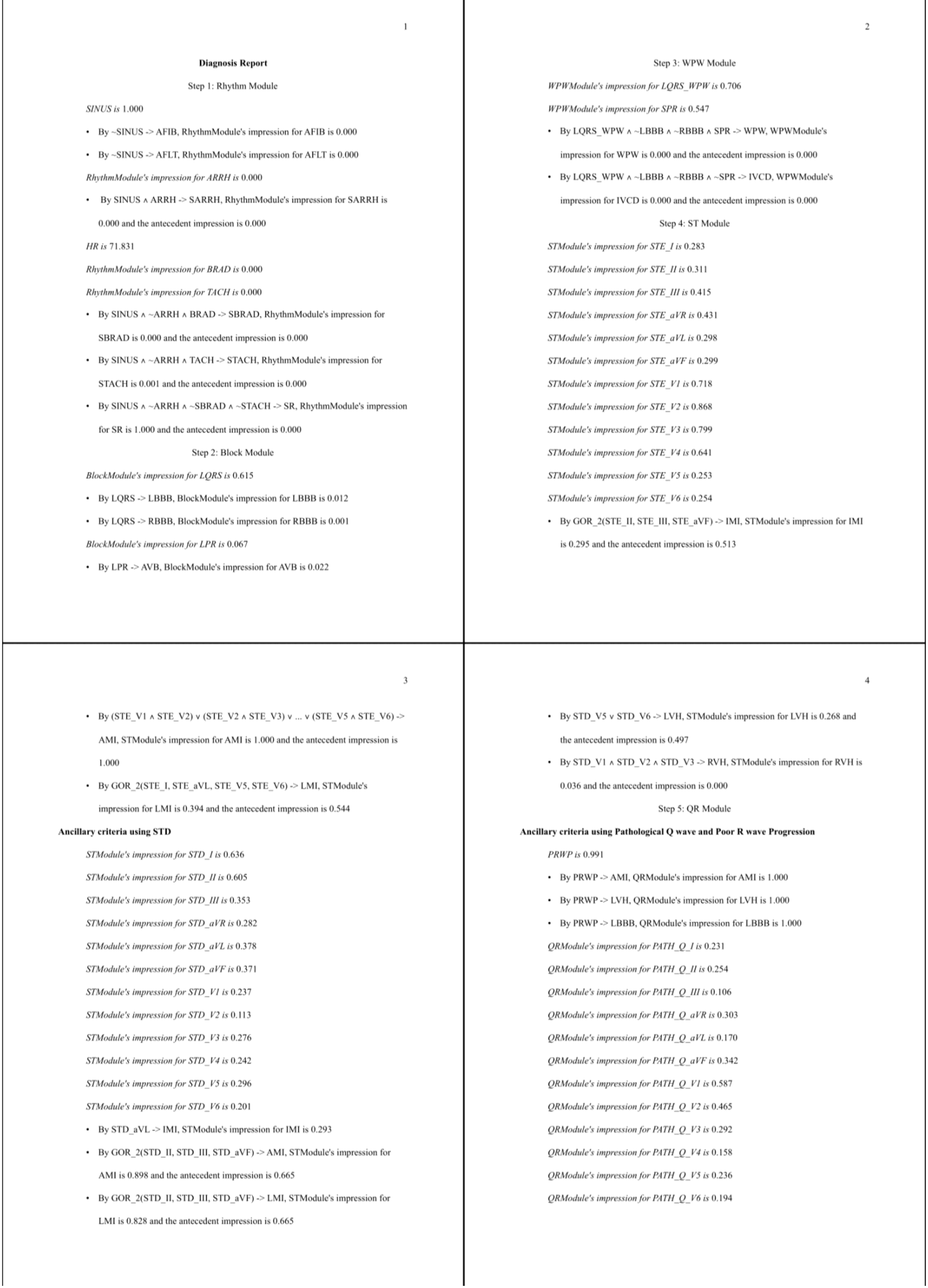
The hyperparameter tuning framework used in this project is Optuna (Akiba et al., 2019), which uses Bayesian optimization techniques to search hyperparameters in the search space provided by the user. The search space for different hyperparameters is summarized in the Table below.

Hyperparameter Type	Is discrete	Is sampled in the log domain	Range
Learning rate	False	True	[1e-5, 1e-1]
Adam's β_1	False	False	[0.9, 0.99]
Exponential learning rate scheduler's multiplicative factor	False	False	[0.95, 1)
Number of convolution layers in CNN	True	False	[1, 5]
Number of output channels of a convolution layer	True	True	[4, 256]
Convolution's kernel size	True	False	[2, 24]
Convolution's stride	True	False	[1, 3]
Max-pooling's kernel size	True	False	[1, 3]
Max-pooling's stride	True	False	[1, 3]
Number of linear layers in MLP	True	False	[1, 5]
Number of hidden neurons in a linear layer	True	True	[4, 256]
HL version implication's lattice size (the granularity of the lattice)	True	False	[2, 6]

The models were trained using a maximum of 50 epochs for each hyperparameter configuration. A total of 100 sets of hyperparameter configurations were attempted for each architecture, and the best-performing configuration was selected for comparison in the "Experiment" section.

A5 Diagnosis Report

The ECG-XAI framework can automatically generate diagnosis reports according to the ECG DDx process in Appendix A3. As an example, a diagnosis report has been generated for an ECG record in the test set that exhibits SR and Acute Myocardial Infarction AMI. The generated report is appended at the end of this document, and the corresponding 12-lead plot for this ECG record can be found in Figure 24.



Diagnosis Report

Step 1: Rhythm Module

SINUS is 1.000

- By ~SINUS -> AFIB, RhythmModule's impression for AFIB is 0.000
- By ~SINUS -> AFLT, RhythmModule's impression for AFLT is 0.000

RhythmModule's impression for ARRH is 0.000

- By SINUS ∧ ARRH -> SARRH, RhythmModule's impression for SARRH is 0.000 and the antecedent impression is 0.000

HR is 71.831

RhythmModule's impression for BRAD is 0.000

RhythmModule's impression for TACH is 0.000

- By SINUS ∧ ~ARRH ∧ BRAD -> SBRAD, RhythmModule's impression for SBRAD is 0.000 and the antecedent impression is 0.000
- By SINUS ∧ ~ARRH ∧ TACH -> STACH, RhythmModule's impression for STACH is 0.001 and the antecedent impression is 0.000
- By SINUS ∧ ~ARRH ∧ ~SBRAD ∧ ~STACH -> SR, RhythmModule's impression for SR is 1.000 and the antecedent impression is 0.000

Step 2: Block Module

BlockModule's impression for LQRS is 0.615

- By LQRS -> LBBB, BlockModule's impression for LBBB is 0.012
- By LQRS -> RBBB, BlockModule's impression for RBBB is 0.001

BlockModule's impression for LPR is 0.067

- By LPR -> AVB, BlockModule's impression for AVB is 0.022

Step 3: WPW Module

WPWModule's impression for LQRS_WPW is 0.706

WPWModule's impression for SPR is 0.547

- By LQRS_WPW ∧ ~LBBB ∧ ~RBBB ∧ SPR -> WPW, WPWModule's impression for WPW is 0.000 and the antecedent impression is 0.000
- By LQRS_WPW ∧ ~LBBB ∧ ~RBBB ∧ ~SPR -> IVCD, WPWModule's impression for IVCD is 0.000 and the antecedent impression is 0.000

Step 4: ST Module

STModule's impression for STE_I is 0.283

STModule's impression for STE_II is 0.311

STModule's impression for STE_III is 0.415

STModule's impression for STE_aVR is 0.431

STModule's impression for STE_aVL is 0.298

STModule's impression for STE_aVF is 0.299

STModule's impression for STE_V1 is 0.718

STModule's impression for STE_V2 is 0.868

STModule's impression for STE_V3 is 0.799

STModule's impression for STE_V4 is 0.641

STModule's impression for STE_V5 is 0.253

STModule's impression for STE_V6 is 0.254

- By GOR_2(STE_II, STE_III, STE_aVF) -> IMI, STModule's impression for IMI is 0.295 and the antecedent impression is 0.513

- By (STE_V1 ∧ STE_V2) ∨ (STE_V2 ∧ STE_V3) ∨ ... ∨ (STE_V5 ∧ STE_V6) -> AMI, STModule's impression for AMI is 1.000 and the antecedent impression is 1.000
- By GOR_2(STE_I, STE_aVL, STE_V5, STE_V6) -> LMI, STModule's impression for LMI is 0.394 and the antecedent impression is 0.544

Ancillary criteria using STD

STModule's impression for STD_I is 0.636

STModule's impression for STD_II is 0.605

STModule's impression for STD_III is 0.353

STModule's impression for STD_aVR is 0.282

STModule's impression for STD_aVL is 0.378

STModule's impression for STD_aVF is 0.371

STModule's impression for STD_V1 is 0.237

STModule's impression for STD_V2 is 0.113

STModule's impression for STD_V3 is 0.276

STModule's impression for STD_V4 is 0.242

STModule's impression for STD_V5 is 0.296

STModule's impression for STD_V6 is 0.201

- By STD_aVL -> IMI, STModule's impression for IMI is 0.293
- By GOR_2(STD_II, STD_III, STD_aVF) -> AMI, STModule's impression for AMI is 0.898 and the antecedent impression is 0.665
- By GOR_2(STD_II, STD_III, STD_aVF) -> LMI, STModule's impression for LMI is 0.828 and the antecedent impression is 0.665

- By STD_V5 ∨ STD_V6 -> LVH, STModule's impression for LVH is 0.268 and the antecedent impression is 0.497
- By STD_V1 ∧ STD_V2 ∧ STD_V3 -> RVH, STModule's impression for RVH is 0.036 and the antecedent impression is 0.000

Step 5: QR Module

Ancillary criteria using Pathological Q wave and Poor R wave Progression

PRWP is 0.991

- By PRWP -> AMI, QRModule's impression for AMI is 1.000
- By PRWP -> LVH, QRModule's impression for LVH is 1.000
- By PRWP -> LBBB, QRModule's impression for LBBB is 1.000

QRModule's impression for PATH_Q_I is 0.231

QRModule's impression for PATH_Q_II is 0.254

QRModule's impression for PATH_Q_III is 0.106

QRModule's impression for PATH_Q_aVR is 0.303

QRModule's impression for PATH_Q_aVL is 0.170

QRModule's impression for PATH_Q_aVF is 0.342

QRModule's impression for PATH_Q_V1 is 0.587

QRModule's impression for PATH_Q_V2 is 0.465

QRModule's impression for PATH_Q_V3 is 0.292

QRModule's impression for PATH_Q_V4 is 0.158

QRModule's impression for PATH_Q_V5 is 0.236

QRModule's impression for PATH_Q_V6 is 0.194

- By GOR_2(PATH_Q_II, PATH_Q_III, PATH_Q_aVF) -> IMI, QRModule's impression for IMI is 0.025 and the antecedent impression is 0.351
- By PATH_Q_V1 ∧ PATH_Q_V2 ∧ PATH_Q_V3 ∧ PATH_Q_V4 -> AMI, QRModule's impression for AMI is 0.001 and the antecedent impression is 0.000
- By GOR_2(PATH_Q_I, PATH_Q_aVL, PATH_Q_V5, PATH_Q_V6) -> LMI, QRModule's impression for LMI is 0.051 and the antecedent impression is 0.4155

Step 6: P Module

PModule's impression for LP_II is 0.249

- By LP_II -> LAE, PModule's impression for LAE is 0.000

PModule's impression for PEAK_P_II is 0.157

PModule's impression for PEAK_P_V1 is 0.266

- By PEAK_P_II ∨ PEAK_P_V1 -> RAE, PModule's impression for RAE is 0.000 and the antecedent impression is 0.423

Ancillary criteria using LAE and RAE

- By LAE -> LVH, PModule's impression for LVH is 0.000
- By RAE -> RVH, PModule's impression for RVH is 0.000

Step 7: VH Module

VHModule's impression for AGE_OLD is 1.000

VHModule's impression for LVH_L1_OLD is 0.000

VHModule's impression for LVH_L1_YOUNG is 0.000

MALE is 1.000

VHModule's impression for LVH_L2_MALE is 0.000

VHModule's impression for LVH_L2_FEMALE is 0.001

- By (AGE_OLD ∧ LVH_L1_OLD) ∨ (~AGE_OLD ∧ LVH_L1_YOUNG) ∨ (MALE ∧ LVH_L2_MALE) ∨ (~MALE ∧ LVH_L2_FEMALE) -> LVH, VHModule's impression for LVH is 0.000 and the antecedent impression is 0.000

VHModule's impression for PEAK_R_V1 is 0.010

VHModule's impression for DEEP_S_V5 is 0.056

VHModule's impression for DEEP_S_V6 is 0.025

VHModule's impression for DOM_R_V1 is 0.024

VHModule's impression for DOM_S_V5 is 0.026

VHModule's impression for DOM_S_V6 is 0.027

RAD is 0.000

- By GOR_2(PEAK_R_V1, DEEP_S_V5 ∨ DEEP_S_V6, DOM_R_V1, DOM_S_V5 ∨ DOM_S_V6, RAD) -> RVH, VHModule's impression for RVH is 0.000 and the antecedent impression is 0.084

Step 8: T Module

TModule's impression for INVT_I is 0.062

TModule's impression for INVT_II is 0.012

TModule's impression for INVT_V1 is 0.142

TModule's impression for INVT_V2 is 0.002

TModule's impression for INVT_V3 is 0.000

TModule's impression for INVT_V4 is 0.001

TModule's impression for INVT_V5 is 0.006

TModule's impression for INVT_V6 is 0.016

- By GOR_2([INVT_x ∧ (STE_x ∨ STD_x), for x ∈ {I, II, V3-V6}]) -> IMI, TModule's impression for IMI is 0.000 and the antecedent impression is 0.000
- By GOR_2([INVT_x ∧ (STE_x ∨ STD_x), for x ∈ {I, II, V3-V6}]) -> AMI, TModule's impression for AMI is 0.047 and the antecedent impression is 0.000
- By GOR_2([INVT_x ∧ (STE_x ∨ STD_x), for x ∈ {I, II, V3-V6}]) -> LMI, TModule's impression for LMI is 0.000 and the antecedent impression is 0.000

Ancillary criteria using inverted T wave

- By INVT_V5 ∨ INVT_V6 -> LVH, TModule's impression for LVH is 0.000 and the antecedent impression is 0.022
- By INVT_V1 ∧ INVT_V2 ∧ INVT_V3 -> RVH, TModule's impression for RVH is 0.000 and the antecedent impression is 0.000

Step 9: Axis Module

Ancillary criteria using electrical axis

AxisModule's impression for POS_QRS_I is 0.982

AxisModule's impression for POS_QRS_aVF is 0.484

AxisModule's impression for NORM_AXIS is 0.466

AxisModule's impression for LAD is 0.499

- By LAD -> LAFB, AxisModule's impression for LAFB is 0.000

AxisModule's impression for RAD is 0.000

- By RAD -> LPFB, AxisModule's impression for LPFB is 0.000

Differential Diagnoses from the ECG-XAI system

```
SR: 1.000

AMI: 0.997
```

```
NORM: 0.041

AVB: 0.022

LBBB: 0.002

IMI: 0.002

LMI: 0.001

STACH: 0.001

RBBB: 0.001

LVH: 0.001

SARRH: 0.000

RVH: 0.000

LPFB: 0.000

LAFB: 0.000

SBRAD: 0.000

RAE: 0.000

WPW: 0.000

IVCD: 0.000

AFLT: 0.000

AFIB: 0.000

LAE: 0.000
```