

Article

# SF-ECG: Source-Free Intersubject Domain Adaptation for Electrocardiography-Based Arrhythmia Classification

Taki Hasan Rafi <sup>†</sup>  and Young-Woong Ko <sup>\*†</sup> 

Department of Computer Engineering, Hallym University, Chuncheon 24252, Republic of Korea; takihanrafi@gmail.com

\* Correspondence: youngwoongKo@hallym.ac.kr

<sup>†</sup> This work has been done during this author's work at Hallym University.

**Abstract:** Electrocardiography (ECG)-based arrhythmia classification intends to have a massive role in cardiovascular disease monitoring and early diagnosis. However, ECG datasets are mostly imbalanced and have regularization to use real-time patient data due to privacy concerns. Traditional models do not generalize on unseen cases and are also unable to preserve data privacy. Which incentivizes performance degradation in existing models with privacy limitations. To tackle generalization and privacy issues together, we introduce the framework SF-ECG, a source-free domain adaptation approach for patient-specific ECG classification. This framework does not require source data during adaptation, which solves the privacy issue during adaptation. We adopt a generative model (GAN) that learns to synthesize patient-specific ECG data in data-inefficient classes to make additional source data for imbalanced classes. Then, we use the local structure clustering method to strongly align target ECG features with similar neighbors. After seizing clustered target features, we use a classifier that is trained on source data with generated source samples, which makes the model generalizable in classifying unseen data. Empirical results under different experimental conditions in various interdomain datasets prove that the proposed framework achieves 0.8% improvements in UDA settings, along with preserving privacy and generalizability.

**Keywords:** electrocardiography; source-free domain adaptation; generative adversarial networks



**Citation:** Rafi, T.H.; Ko, Y.-W.

SF-ECG: Source-Free Intersubject Domain Adaptation for Electrocardiography-Based Arrhythmia Classification. *Appl. Sci.* **2023**, *13*, 8551. <https://doi.org/10.3390/app13148551>

Academic Editor: Atsushi Mase

Received: 2 July 2023

Revised: 20 July 2023

Accepted: 21 July 2023

Published: 24 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Electrocardiography is a noninvasive tool that is accessible globally for the initial diagnosis of cardiovascular diseases [1,2]. There are many forms of cardiovascular disease that can be identified via the ECG graph of a patient, including arrhythmia [3] Due to the huge volume of its application, researchers are developing several effective tools to classify ECG signals effectively. Arrhythmia is considered the most common form of heart disease, which refers to a condition where heart rhythms are irregular. It can be detected with abnormalities in the ECG graph. But analyzing the ECG graph abnormalities is time-consuming and difficult without expert supervision. On the other hand, the morphology and characteristics of different patients' ECG signals have diversity due to different physiological situations. Some of the same diseases can have different patterns in the ECG signals [4], which can be difficult to interpret or annotate for generating training samples' deep learning models. Recently, deep learning (DL) appears with strong baseline capabilities in many real-life applications. DL-based ECG classification has become so popular and mainstream nowadays due to the growth of AI in disease diagnosis. As DL has the powerful attribute to learn features from large-scale clinical data in various conditions, several successful DL-based diagnosis tools are designed [3]. Mostly, CNNs and RNNs are used to learn the high-level features from raw time series ECG data, and these features are utilized in the classification task. But these classification models are not generalizable in new subject data during testing. In most real-time applications, these models fail to deliver desired precision in classification.

ECG classification can be referred into two schemes, namely, intrasubject and intersubject. However, the training and test examples are differently distributed but belong to the same subject and can be referred to as intrasubject. The main limitation of developing the intrasubject scheme is that models are not well generalized in new subject cases; hence, there is a noticeable performance drop, whereas if the training and test examples are not overlapping and belong to different subjects, that can be referred to as intersubject. In the intersubject scheme, the samples have differences individually. Individual differences refer to differences in sex, age, physiological conditions, etc. In this case, the classification performance is based on being subject-to-subject. Therefore, differences in the training and testing examples make the whole learning generalizability difficult; hence, it is considered the most challenging and realistic setting for developing ECG classification models [5].

Even though DL methods have made great advances in the intrasubject scheme, they are still facing considerable performance degradation in the intersubject scheme, as the intersubject scheme is a more realistic setting for developing applications, since each subject/patient has different morphological characteristics in ECG signal data due to the variance in individual cases. Considering the application reliability, a method called subject-independent refers to the training of a model with labeled training samples and testing the model into the new test samples strictly. But in this case, the model performance deteriorates by a significant margin, which can not be relied on while building models. The problem can be solved via subject-specific training with a specific model, where the whole scheme can be divided into two phases, such as initial training and fine tuning [5]. Usually, in the initial training, the model uses the training samples to train, and in the next stage, the model is fine-tuned by partially labeled samples from test samples. The whole phase increases the adaptation capacity of intersubject variation. But this scheme is associated with various underlying issues, such as developing models for each individual subject or patient is not feasible for real-time deployment where computational resources are limited. Another major drawback is that annotating such a large amount of subject-wise data tends to be expensive and laborious. Instead of annotating in a conventional way, researchers adopt domain adaptation (DA) methods to solve these limitations, where domain shift refers to distribution diversity between train and test samples [6]. And DA is a prominent method that solves the issue of domain shift. In that case, training and test samples are considered as source and target samples, respectively.

However, ECG datasets are often limited for real-time testing, and data imbalance issues are also there. For example, the MIT-BIH arrhythmia dataset is an imbalanced dataset, where normal class data are higher than rare disease class data. In other datasets, this sparse representation of data is presented. Another challenge is collecting well-representable data for model training. Technically, deep learning models show an inherent bias towards classes that are more common in training [7]. But individual performance in minor classes is not appreciable considering the relative overall performance based on accuracy. Hence, we need a well-occupied data augmentation method that can create synthetic data with respect to the original data for the minor classes. But it causes a marginal performance boost that might not be sufficient in medical diagnosis. Hence, generative models can learn the distribution of the training data so that it generates samples that have the same distribution. Thus, it can improve the classification performance in minor classes.

Most of the unsupervised domain adaptation (UDA)-based methods require a common connection between source and target distributions [8], and also need access to source data during real-time deployment. But in medical cases, it is intended to be highly impractical due to the nature of privacy concerns in patient data due to several ethical implication issues, such as how most UDA cases and other traditional models tend to access the direct patient data in the real-time adaptation, which breaks data confidentiality. It is also highly risky that traditional approaches are vulnerable in preserving data anonymization since most of the deep learning approaches tend to memorize specific information of the patient. There are some other ethical concerns that are associated with data privacy, such as data sharing without explicit consent, bias, and discrimination, etc. And it is not feasible to train

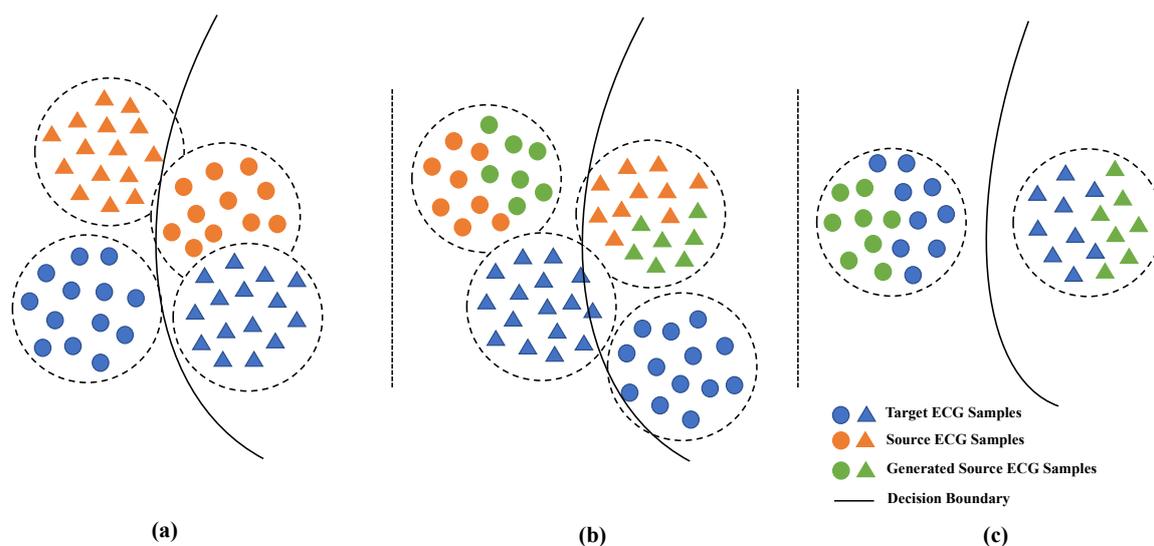
a model with limited computational resources. On the other hand, an accidental loss of source data can make it impossible for traditional UDA techniques to generalize in new subject data. Therefore, there are many underlying challenges even in the UDA setting for ECG classification tasks. In Figure 1, we illustrate the source and target sample before adaptation and also illustrate the effect of source-free adaptation with generated samples.

To tackle the aforementioned issues in traditional DL-based models and UDA methods in ECG classification, we adopt a novel problem-setting source-free adaptation. Source-free adaptation (SFA) [8,9] is a new extension of the area of DA/UDA, where in real-time adaptation, a source dataset is not required. Perhaps, it is suitable for overcoming all the open challenges in ECG classification can be solved via the SFA method. So, we reinforce the SFA idea into ECG classification, which is totally a novel scenario in this problem. We propose a source-free intersubject method, SF-ECG, for ECG classification. Our proposed framework can improve the intersubject performance of DNN without requiring source data in adaptation. Our framework consists of three modules: data adjustment, local clustering, and data adaptation. More of the details can be found in Section 3.

Our contributions are summarized into four folds:

- To the best of our knowledge, we first incorporate the source-free adaptation strategy to address interpatient ECG classification.
- We propose an SF-ECG framework, which is an unsupervised domain adaptation method that predicts unlabeled target samples without requiring source data in the test-time adaptation setting.
- We adopt a generative model that generates high-quality ECG training samples for the source classifier. It alleviates the data insufficiency problem in class-wise training. Which exhibits a strong source classifier that is utilized in the test-time adaptation.
- Empirical experiments show that our framework is capable of classifying interpatient ECG samples with high precision even in this novel scenario, outperforming many state-of-art methods in the same settings with a better generalization ability.

**Organizations.** The remaining part of the paper is organized as follows, Section 2 discusses related work. Section 3 provides an extensive demonstration of our proposed framework in detail. Section 4 presents the experimental setting, results, and ablation studies. And finally, Section 5 ends with a discussion and conclusion of this work.



**Figure 1.** Illustration of domain adaptation method. (a) Source and target samples before adaptation. (b) New source samples are generated by a generative model for imbalanced datasets with the same distribution as the original source distribution. (c) After adaptation by our SF-ECG using the generated samples.

## 2. Related Work

This section is comprised of recent related works on unsupervised domain adaptation (UDA) and generative adversarial network (GAN) for ECG classification tasks.

### 2.1. UDA for ECG Signal Classification

Recently, there are many recent works that adopt the UDA method in interclass ECG classification tasks. Chen et al. [1] proposed a multipath network as a baseline classification model assigned with cluster-aligning separating loss to reduce the discrepancy between train and test distributions for ECG-enabled arrhythmia classification. Deng et al. [2] proposed a multisource UDA method that utilizes multiple data sources to incorporate generalization. This method comprises a two-stage adaptation and imbalance-aware mixing strategy to learn features across all domains from multiple sources. Wang et al. [3] introduced a UDA method that learns discriminative features from clustering characteristics of source and target data. They also utilized cluster-aligning and maintaining losses to regulate and structure feature information of the source and target data in invariant space. Yuan et al. [4] introduced a hypergraph-based UDA method that exploits cross-attention dual-channel networks in adaptation tasks. They also used a domain alignment method based on Wasserstein distance for edge features, which also applies a pseudo-label generation technique for retaining category-level fine-grained information of the distribution. He et al. [5] proposed a multilevel UDA method, which comprises an ASPP-R module to extract SP features; then, they used Graph-CN to extract structural features of the data. The whole framework exhibits three feature alignment stages, namely, domain, semantic, and structure alignment. Bazi et al. [10] utilized domain transfer SVM and weighted KLR methods to compare within the MIT-BIH dataset to see the statistical shift between the train and test ECG samples. Niu et al. [11] proposed an adversarial DA method that comprises three different modules, namely, feature extractor, discriminator, and classification tasks. These modules reduced the performance gap under discrepant ECG data distribution. Yin et al. [12] introduced a self-adjustable domain adaptation method to dispute the overfitting issue and unlabeled data exploitation. They adapted data augmentation methods to extend the ECG database in training conditions and a self-obtained map of the unlabeled data features with the transfer learning strategy. He et al. [13] introduced an online cross-domain DA approach, which is based on a shared subspace classifier. It reduces the interclass discrepancy. It also includes an online adaptation method for emotion recognition via ECG data, which apparently defers from the ECG-based disease classification.

Natarajan et al. [14] addressed the generalization issue of DA methods in lab-to-field data scenarios in ECG-based cocaine use detection. This works limits the prior probability, covariate, and label granularity shifts in lab-field data usage. Hang et al. [15] proposed a UDA method that uses a CNN to extract deep features from raw data; then, they used MMD to increase distribution gaps between source and target features. Ammour [16] proposed an asymmetric DANN method that uses a denoising autoencoder to learn features and FCNN layers to alienate data-shift. Helm et al. [17] introduced a Fisher's linear discriminant method for DA that determines the expected risk of the combined hypothesis of the target distribution and also estimates a convex coefficient of BV-trade between source and target samples. This method explores both ECG and EEG data in the DA setting. Carrera [18] et al. proposed a DA method that learns the user-independent linear transformation of the ECG data. They also modeled heartbeat-based sparse representations. This transformation method maps the patient-specific dictionaries in modeling ECG heartbeats, which is usable for online ECG monitoring. Chen et al. [19] proposed a UDA method based on an adaptive region network; they utilized adversarial training to reduce the discrepancy of the domain. The proposed model learns the invariant features with a generative network, consisting of one generator and one discriminator. Yamac et al. [20] proposed a null space-based analysis for healthy cardiac signal space, reducing computational complexity. However, they also introduced a sparse-representation-based UDA method that learns new signals without

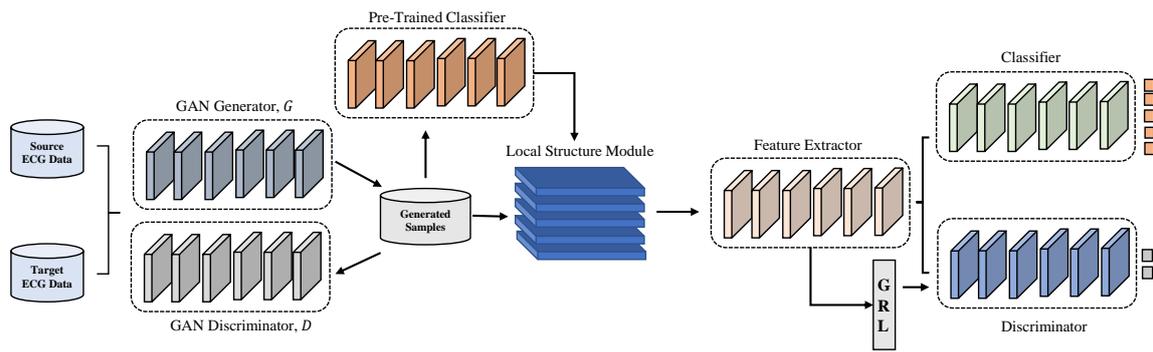
having a domain shift between existing signals and new signals. Ye et al. [21] introduced a clustering-enhanced UDA method that relies on a discriminative attribute. They used a self-supervised loss on the target data and obtained pseudolabels based on the target data. The framework encourages ambiguous target data to be correctly classified. Li et al. [22] proposed a signal quality assessment strategy based on the UDA method, which increases the feature extraction from target data. The framework is based on a feature distribution alignment strategy by an adaptation layer. Peng et al. [23] introduced a UDA method that learns universal features by adversarial autoencoder. The embedding space after adversarial training aligned with the Riemannian manifold prior to obtaining cross-domain features. They also used a variational interface method to increase the generalization capacity of the framework.

## 2.2. GAN for ECG Signal Classification

Previous works have also explored generative adversarial networks (GANs) for generating synthetic data for imbalanced datasets, whereas most of the well-known ECG datasets and real-world databases are not balanced. Shaker et al. [7] proposed a data augmentation method with GAN; they utilized a CNN to extract features and leverage a feature reduction strategy for their framework that increases the high number of samples in the training set. Golany et al. [24] proposed a simulation-based GAN framework that understands the complex dynamics of generating close-to-real ECG training samples with the help of ordinary differential equations. Golany and Radinsky [25] introduced a GAN-based semisupervised framework that learns to generate synthetic per-patient ECG data to improve the patient-specific ECG classification. Golany et al. [26] studied GAN for generating improved training ECG samples following LSTM-based ECG classification. Wang et al. [27] introduced a data augmentation method based on an auxiliary GAN classifier. They constructed a classification model based on a stacked residual network with LSTM. Yang et al. [28] propose ProGAN method based on GAN that solves the data imbalance issue. They generated a diverse state of ECG samples so that the model obtains higher fidelity and diversity in training samples. Hence, that ensures better generalization in the test set. Ye et al. [29] introduced a policy gradient-based sequence GAN framework for negotiating with low-quality training without insufficient extracting global ECG features. The proposed framework shows a stable variance in generation loss. Zhu et al. [30] proposed a bidirectional LSTM-CNN-based GAN framework for ECG training sample generation with a diverse combination of generators and discriminators. Wang et al. [31] proposed a conditional GAN-based ECG denoising method that comprises a convolutional autoencoder as the generator, which preserves spatial locality and feature representations. Several hidden layers are the discriminator. Harada et al. [32] proposed a recurrent neural network-based GAN framework for ECG data augmentation and generation. Singh and Pradhan [33] proposed a CNN-GAN framework for ECG noise filtering for denoising. Adib et al. [34] studied five different GANs, including BiLSTM-DC-GAN and WGAN, for generating synthetic ECG signals for robust training ECG samples. They also proposed a residual ECG network for ECG classification. Rafi and Young [35] proposed a multihead attention-based CNN framework that adopts GAN to tackle data insufficiency issues in imbalanced datasets by generating additional ECG data samples.

## 3. Proposed Method

Our framework for source-free intersubject ECG classification is divided into three modules, namely, data adjustment, local structure learning, and adaptation network. In the data adjustment module, we use a generative model to generate synthetic samples based on the source sample distribution. The local structure module learns the target features and clusters these features based on the neighbors. An illustration of the overall framework is shown in Figure 2.



**Figure 2.** Illustration of our SF-ECG framework. Here, GAN model is used to generate synthetic samples from the training distribution, and the local structure module clusters the generated samples in a local neighborhood. Lastly, another GAN model exhibits the prediction without explicitly requiring the source data.

### 3.1. Preliminaries

In the source-free domain adaptation task (SFDA), we consider  $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ ; here,  $n_s$  is the number of samples in the source set that are labeled.  $x_i^s$  and  $y_i^s$  are the source samples and their corresponding labels, respectively. We also have the target set  $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$ ; here,  $n_t$  is the number of samples that have no corresponding labels. In the SFDA settings, we only have access to the source set in the model pretraining condition. However, it is not available in the real-time adaptation task.

### 3.2. Generative Model for Data Adjustment

In various cases, ECG datasets often face data imbalance in minor classes. To generate synthesized data, we use a generative model, GAN [25–27]. Generally, GAN is formulated into two modules: (1) generator and (2) discriminator. The generator takes source samples with random noise,  $z$ . This random noise  $z$  learns the distribution of the source data  $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$  and outputs synthetic samples  $\mathcal{D}_f = (\mathcal{X}_f, \mathcal{Y}_f)$ . The output of the discriminator is usually the synthetic and original data. The discriminators’ objective is to distinguish the synthetic and original data precisely. Basically, the generator is trained to generate synthetic data to classify the original by the discriminator. On the other hand, the discriminator tends to recognize the synthetic and original data as accurately as possible. The training objective of GAN can be described as the value function  $\mathcal{V}(\mathcal{D}, \mathcal{G})$ , where the generator value function  $\mathcal{V}(\mathcal{G})$  is expected to maximize and the discriminator value function  $\mathcal{V}(\mathcal{D})$  is expected to minimize:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{G}, \mathcal{D}) = E_{x \sim P_{data}(x)} [\log \mathcal{D}(x)] + E_{z \sim P_z(z)} [\log(1 - \mathcal{D}(\mathcal{G}(z)))]. \tag{1}$$

Here,  $\mathcal{D}(x)$  and  $\mathcal{G}(z)$  are denoted as the probability of  $x$ , which belongs to the original data distribution  $P_{data}$ , and a substantial mapping function obtained from the noise vector to the generated vector, respectively. However, the optimal parameters can be obtained by maximizing and minimizing the value function of the discriminator and generator, respectively. Discriminator  $\mathcal{D}$  is trained with both positive and negative ECG samples in each iteration. And the generator  $\mathcal{G}$  is updated with policy gradients two times, while  $\mathcal{D}$  is updated once; we consider this strategy from [25]. Training converges until  $\mathcal{G}$  is indistinguishable. Generated synthetic samples on the class-deficient category alleviate this issue, and certainly make the dataset considerably balanced. We use the balanced dataset in the local structure module (C). Adversarial loss and cross-entropy loss are adopted for the GAN module. Table 1 depicts the model description.

**Table 1.** Model description of GAN framework for data adjustment.

Discriminator			Generator		
Layers	Output Shape	Parameters	Layers	Output Shape	Parameters
Conv1D	(None, 180, 8)	72	Reshape	(None, 100, 1)	0
LeakyReLU	(None, 180, 8)	0	Bidirectional	(None, 100, 32)	2304
Dropout	(None, 180, 8)	0	Conv1D	(None, 100, 32)	8224
MaxPooling1D	(None, 60, 8)	0	LeakyReLU	(None, 100, 32)	0
Conv1D	(None, 60, 16)	1040	UpSampling1D	(None, 200, 32)	0
Dropout	(None, 60, 16)	0	Conv1D	(None, 200, 16)	4112
MaxPooling1D	(None, 29, 16)	0	LeakyReLU	(None, 200, 16)	0
Conv1D	(None, 15, 32)	4128	UpSampling	(None, 200, 16)	0
LeakyReLU	(None, 15, 32)	0	Conv1D	(None, 200, 16)	1032
Dropout	(None, 15, 32)	0	LeakyReLU	(None, 400, 8)	0
MaxPooling1D	(None, 7, 32)	0	Conv1D	(None, 400, 1)	65
Conv1D	(None, 4, 64)	16,448	Flatten	(None, 400)	0
LeakyReLU	(None, 4, 64)	0	Dense	(None, 180)	72,180
Dropout	(None, 4, 64)	0	Activation	(None, 180)	0
MaxPooling1D	(None, 64)	0	Reshape	(None, 180, 1)	0
Flatten	(None, 64)	0	-	-	-
Dense	(None, 1)	65	-	-	-

### 3.3. Local Structure Module

In traditional settings, UDA methods are based on the feature alignment strategy of source and target features. But in the SFA setting, it is not feasible to access the source data. We usually obtain a class prediction  $p(x)$  and feature embedding space  $f(x)$ . Following [9], the local structure method intends to shift the target features with the source domain. In the first place, due to the expected domain shift, some of the target features can be deviated by far from the source feature region. But in this case, we still consider that feature space retains the clusters that are formed by the classes. For this, we actually measure the distance and move the data point into the close likely data point cluster. Due to the domain shift between the source and target interpatient data, there can be wrong predictions by the classifier. So, it is essential to cluster target features that are in the same class clustered together. Therefore, it is obvious that the nearest neighbor target features tend to have share category labels. As a result, the clustered features are most likely to have an inclination toward a common label jointly. As shown in Figure 2, this scheme helps to classify target features correctly that possibly have the wrong classification.

To ensure semantically close neighborhoods, we use a feature extractor bank  $\mathcal{F} = \{f(x_i)\}_{i \in \mathcal{D}_i}$  to accumulate joint target features. Then, we utilize a neighborhood selection procedure to ensure the best clusters among target features. It explicitly encourages correct classification. We take a few nearest neighbors from the feature extractors with a consistency regularization. To store the softmax predicted scores, we use a score bank  $\mathcal{S} = \{g(f(x_i))\}_{i \in \mathcal{D}_i}$ . Here,  $g, f(x_i)$  are the classifier and feature extractor, respectively. To achieve this, we adopt the ResNet-50 model with a fully connected layer as the feature extractor  $f(x_i)$ , and an additional fully connected layer as classifier  $g$ . The local clustering is obtained by adopting the following loss in Equation (2) for consistent predictions in the  $k$ -nearest features [9].

$$\mathcal{L}_{LC} = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K \log[p(x_i) \cdot s(\mathcal{N}_k)] + \sum_{c=1}^C \text{KL}(\bar{p}_c || q_c)$$

$$\mathcal{N}_{\{1, \dots, K\}} = \{\mathcal{F}_j \mid \text{top} - K(\cos(f(x_i), \mathcal{F}_j), \forall \mathcal{F}_j \in \mathcal{F})\}$$

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_c(x_i), \text{ and } q_{\{c=1, \dots, C\}} = \frac{1}{C}.$$
(2)

Here,  $\mathcal{N}$  is the k-nearest neighbors stored in the feature extractors for each target feature; it is calculated based on the cosine similarities. The dot product between stored prediction scores  $s(\mathcal{N}_k)$  and target sample  $x_i$  is minimized of the negative log values. Equation (2) intends to ensure the consistent correct prediction between the nearest neighbors and features. Here, the term  $p_c$  and  $q$  are the empirical label distributions and uniform distribution, respectively. And finally, the old items are replaced in the bank with the new items using the corresponding minibatch. Figure 3 illustrates the local structure module.

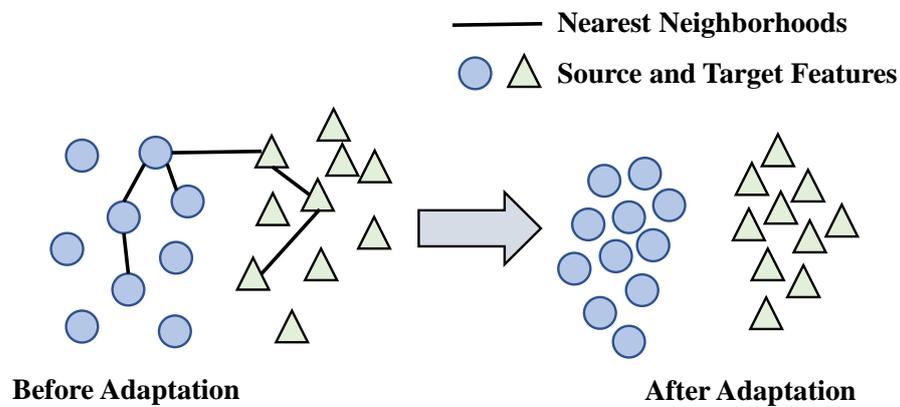


Figure 3. Illustration of local structure module.

### 3.4. Source-Free Domain Adaptation Module

Our domain adaptation module has the feature extractor from the local structure module, which is a shared feature extractor with source and target features aligned. We also have a classifier network and a discriminator; the whole setting is replicated from [36]. As we know, in the SFDA setting, source data are not accessible during the adaptation. So, in this case, using the GAN framework we generated identically distributed samples from the source samples to use in the training time with  $\mathcal{D}_t$ . As we already have the feature extractor, which exhibits the neighborhood-based domain-invariant target feature alignment with the source samples, the discriminator tends to separate the generated samples from the unlabeled target samples. The classifier network is utilized to classify the target samples while it is trained and fine-tuned with the  $\mathcal{D}_f$ . As such, no source database  $\mathcal{D}_s$  is required in the adaptation time, since we are only using the generated source samples. All the feature extractors, classifiers, and discriminators have learnable parameters. However, in the adaptation task, fixed numbers of generated source samples from  $\mathcal{D}_f$  are taken; hence, the adaptation performance is based on the number of source samples that are used during the adaptation.

### 3.5. Objective Function

Following the work in [36], we use four types of loss functions in our framework to train. Here is the description of each loss function.

**Adversarial Loss ( $\mathcal{L}_{adv}$ ):** Adversarial loss is used during the GAN discriminator training; it actually helps the discriminator to discriminate between synthetic data and original data. There are two adversarial losses for the generator and discriminator. Here,  $\mathcal{L}_{g(adv)}$  and  $\mathcal{L}_{d(adv)}$  are the generator and discriminator loss, respectively. Generator loss  $\mathcal{L}_{g(adv)}$  is denoted as:

$$\mathcal{L}_{g(adv)} = \sum_i^n \log(1 - \mathcal{D}(\mathcal{G}(z_i, y_i))) \quad (3)$$

where  $z$  and  $y$  are the random noise and generated class labels, respectively. The discriminator loss  $\mathcal{L}_{d(adv)}$  is denoted as:

$$\mathcal{L}_{d(adv)} = \sum_i^n \log \mathcal{D}(\mathcal{G}(z_i, y_i)) + \sum_{t_i \sim \tau} \log \mathcal{D}(t_i) \quad (4)$$

Here,  $t_i$  is the target data, which are sampled from  $\tau$  distribution.

**Cross-Entropy Loss ( $\mathcal{L}_{CE}$ ):** This loss is calculated when the generated samples are used in the pretrained classifier. This loss ensures consistent parameters by not updating the pretrained classifier. ( $\mathcal{L}_{CE}$ ) is denoted as:

$$\mathcal{L}_{CE} = \frac{1}{\mathcal{D}_{nf}} \sum_{\mathcal{G}(z_i, y_i) \in \mathcal{D}} \mathcal{L}_t(P_c(\mathcal{G}(z_i, y_i)), y_i) \quad (5)$$

Here,  $P_c$ ,  $\mathcal{L}_t$  is the pre-trained classifier and traditional cross-entropy [36] loss respectively.  $\mathcal{D}_{nf}$  is the number of synthetically generated samples.

**Discriminative Loss ( $\mathcal{L}_D$ ):** Domain invariant features are obtained from the feature extractor, and discriminative loss is used in the feature extractor. This is apparently a binary loss between generated source and target samples. We train this feature extractor using a gradient reversal layer [37]. ( $\mathcal{L}_D$ ) is denoted as:

$$\mathcal{L}_D = \frac{1}{n} \sum_{x_i \in \mathcal{D}_f \cup \mathcal{D}_t} \mathcal{L}_t(\mathcal{D}_d(\mathcal{F}(x_i)), l_i) \quad (6)$$

Here,  $n$  and  $l_i$  are the total number of generated and target samples and domain labels, respectively.

**Classification Loss ( $\mathcal{L}_{cls}$ ):** The classification loss for the classifier in training with the generated samples. The gradient of classification loss is also utilized in the feature extractor training.

$$\mathcal{L}_{cls} = \frac{1}{\mathcal{D}_{nf}} \sum_{\mathcal{G}(z_i, y_i) \in \mathcal{D}} \mathcal{L}_t(C(\mathcal{F}(\mathcal{G}(z_i, y_i))), y_i) \quad (7)$$

Here,  $C$  is the classifier and  $\mathcal{D}_{nf}$  is the number of synthetically generated samples.

**Total Loss ( $\mathcal{L}_{total}$ ):** The total loss is expressed by:

$$\mathcal{L}_{total} = \alpha \times \mathcal{L}_{adv} + \beta \times \mathcal{L}_{CE} + \gamma \times \mathcal{L}_D + \omega \times \mathcal{L}_{cls}. \quad (8)$$

Here,  $\alpha, \beta, \gamma, \omega$  are the tuning factors. Ideally,  $\alpha, \gamma$ , and  $\beta$  are set to 1, and  $\omega$  is set to 0 until 50 epochs, and then it is set to 1.

## 4. Experiments

### 4.1. Datasets

In this experiment, we evaluate our proposed model in six publicly available databases, Which makes our experiment thorough and robust in the domain adaptation context: MIT-BTH arrhythmia database (MITDB), St. Petersburg Institute of Cardiological Technics 12-lead Arrhythmia Database (INCARTDB), Physikalisch-Technische Bundesanstalt (PTB), PTB\_XL, e MITBIH Long-Term ECG database (LTDB), and the MIT-BIH Supraventricular Arrhythmia Database (SVDB). These databases are labeled extensively by domain experts.

Following the [5] standard protocol based on the AAMI, as per the AAMI standard suggested, all the heartbeats are categorized into five different types, namely, **N** (normal), **V** (ventricular ectopic), **S** (supraventricular ectopic), and **F** (fusion). Description of respective databases are given below:

**MITDB [38]**: MITDB database is the most commonly utilized database for arrhythmia classification. This dataset has 48 records, with 47 subjects and a sampling rate of 360 Hz. For this experimental purpose, we divided this database into two parts: DS1 and DS2. We use them as a source and target set jointly. This conversion is performed based on the intersubject scheme;

This database has 75 records from 30 individual subjects. The sampling frequency chosen for each recording is 257 Hz. This database contains around 175,000 samples.;

**SVDB [39]**: This database contains nearly 78 records, which subsequently belong to the S class of MITDB database. Each sample is recorded at 128 Hz;

**LTDB [39]**: This database contains seven recordings that are long-term, ranging from 14 h to 22 h. Each sample is recorded at 128 Hz, same as the SVDB database;

**PTB [39]**: This database has 549 records from 290 individuals. Each of the recordings is sampled at 100 Hz and the length ranges from 30 s to 120 s. This database has five subtypes, unlike other databases such as (AMI, ASMI, ALMI, IMI, and ILMI);

**PTB\_XL [40]**: This database is a large-scale database that consists of 21,887 records from 18,885 individuals. Each record is sampled at 500 Hz and the length is 10 s for each sample. In Tables 2 and 3, the number of samples in each databases are given.

**Table 2.** Number of preprocessed samples in each database.

Database	N	V	S	F	Total	Records	Datasets
MITDB	89,507	6872	2764	801	100,035	44	-
INCARTDB	152,889	19,914	1952	219	174,974	75	-
SVDB	161,509	9900	12,450	23	183,582	78	-
LTDB	600,167	64,080	1499	2906	668,652	7	-
MITDB	45,642	3776	943	413	50,774	22	DS1
MITDB	43,865	3196	1821	388	49,261	22	DS2

**Table 3.** Number of preprocessed samples in PTB and PTB\_X databases.

Database	AMI	ASMI	IMI	ALMI	ILMI	HC	Total
PTB	6178	11,062	12,402	6456	7513	10,318	53,929
PTB_X	850	13,898	10,371	1345	3228	9678	39,370

#### 4.2. Evaluation Protocols

We use four evaluation protocols to evaluate our proposed method: (1) accuracy ( $Acc$ ), (2) sensitivity ( $Se$ ), (3) positive predictivity ( $P_p$ ), and (4) F1 score (F1) [3,5]. These metrics are expressed as follows:

$$Acc = \frac{XP + XN}{XP + XN + YP + YN} \quad (9)$$

$$Se = \frac{XP}{XP + YN} \quad (10)$$

$$P_p = \frac{XP}{XP + YP} \quad (11)$$

$$F1 = \frac{2 \times Se \times Pp}{Se + Pp}. \quad (12)$$

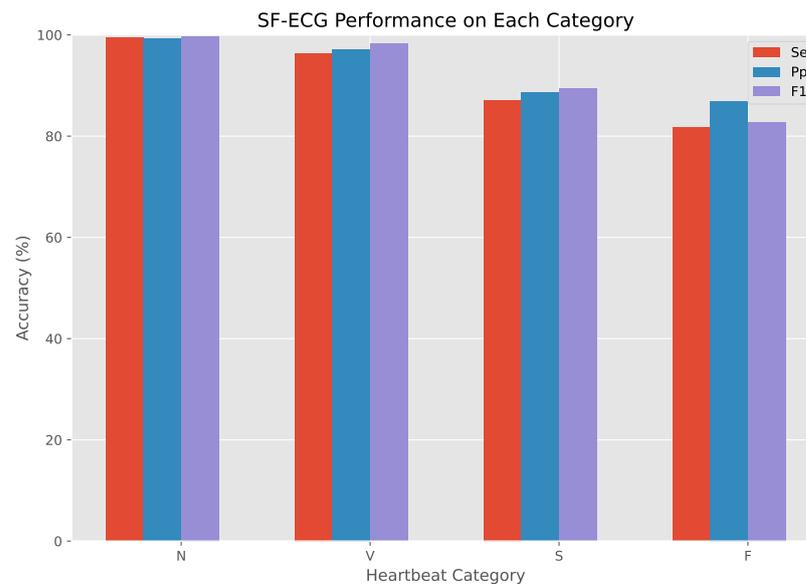
Here,

XP = true positive samples  
 XN = true negative samples  
 YP = false positive samples  
 YN = false negative samples

### 4.3. Experimental Results

#### 4.3.1. Results on DS1 and DS2

Tables 4 and 5 show the results on the MITDB dataset based on DS1 → DS2 and DS2 → DS2, respectively. As we mentioned, they divided the MITDB dataset into two parts, DS1 and DS2, for the cross-domain task evaluation. We preprocessed the dataset and increased the number of samples in the V, S, and F segments for fair training. In particular, in the sample, deficient classes such as S and F received a boost in Se, Pp, and F1 scores (Figure 4). Overall, both DS1 → DS2 and DS2 → DS1 achieved an acceptable overall accuracy of 98.4% and 96.5% respectively. However, we also tested our model performance in terms of K-fold cross-validation to understand its generalization ability; we also show the confusion matrix of our model in Tables 6 and 7, respectively.



**Figure 4.** Class-wise performance of our framework in MITDB at DS1 → DS2 setting.

**Table 4.** Results on DS1 → DS2. Here, DS1 and DS2 are utilized as source and target domains. But in the adaptation time, DS1 (source) is not utilized. (All results shown here are **without requiring source data**).

DS1 → DS2	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)
N	99.4	99.2	99.7
V	96.4	97.2	98.2
S	97.1	98.6	99.4
F	85.4	83.4	88.2
Overall Accuracy (%)		<b>98.4</b>	

**Table 5.** Results on DS2 → DS1. Here, DS1 and DS2 are utilized as source and target domains. But in the adaptation time, DS2 (source) is not utilized. (All results shown here are **without requiring source data**).

DS2 → DS1	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)
N	94.2	94.7	96.2
V	96.2	92.3	92.3
S	96.4	97.4	96.2
F	85.6	86.8	83.2
Overall Accuracy (%)		<b>96.5</b>	

**Table 6.** K-fold cross validation results on DS1 → DS2 with standard deviation.

Fold, K	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)	Accuracy (%)
K = 1	93.2 ± 0.6	93.4 ± 0.6	94.2 ± 0.5	92.2 ± 0.3
K = 2	95.6 ± 0.2	95.3 ± 0.3	94.2 ± 0.3	94.3 ± 0.2
K = 3	96.1 ± 0.7	96.7 ± 0.4	96.2 ± 0.5	96.8 ± 0.3
K = 4	98.2 ± 0.8	97.0 ± 0.7	98.0 ± 0.7	97.9 ± 0.5
K = 5	99.1 ± 0.5	99.2 ± 0.5	98.5 ± 0.3	98.2 ± 0.4

**Table 7.** Confusion matrix on DS2 → DS1 adaptation.

	N	V	S	F
N	44,176	226	46	39
V	43	3196	76	31
S	23	7	3681	6
F	24	30	38	1779

#### 4.3.2. Results on Original Source to Target Data

In Table 8, we provide the results on the original source and target domains. Here, we consider the source domain as a train set and the target domain as a test set. In this scenario, we do not use the adaptation task to understand how our model can perform in traditional conditions. We observe that the results we obtained in the traditional setting are nearly the same as in the adaptation setting. Hence, our model has generalization power, even without the adaptation setting.

**Table 8.** Results on the original source to target domains.

Class	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)
N	98.3	98.7	99.1
V	97.2	96.8	98.1
S	96.2	95.3	96.6
F	85.4	87.2	85.1
Overall Accuracy (%)		<b>98.1</b>	

#### 4.3.3. Results on MITBD, INCARTDB, SVDB, and LTDB

In Table 9, we show the results of the generalization of our proposed source-free domain adaptation model in different scenarios. In this case, we only consider overall accuracy by combining each class, such as N, V, S, and F. Here, S and F tend to have

fewer samples, but we increase the number of samples adopting the GAN framework. We evaluate the MITDB database as the base database for the adaptation task. Therefore, at first, we use the MITDB database for our initial source training and evaluate each database such as INCARTDB, SVDB, and LTDB. We denote each of the databases as MITDB ( $\mathcal{D}_M$ ), INCARTDB ( $\mathcal{D}_I$ ), SVDB ( $\mathcal{D}_S$ ), and LTDB ( $\mathcal{D}_L$ ). Then, we also use INCARTDB, SVDB, and LTDB as source sets and MITDB as target sets for experimental purposes. In every case, we have a boost after adopting our source-free domain adaptation method, even in critical S and F classes where the samples are very less than the N and V classes. Our strategy of adopting GAN for generating more synthetic samples increases the generalization of our method.

**Table 9.** Cross-domain experimental results on each domain. Here, MITDB ( $\mathcal{D}_M$ ), INCARTDB ( $\mathcal{D}_I$ ), SVDB ( $\mathcal{D}_S$ ), and LTDB ( $\mathcal{D}_L$ ). Results without domain adaptation and with source-free adaptation are shown.

Task		N (%)			V (%)			S (%)			F (%)			Accuracy (%)
		Se	Pp	F1										
$D_M \rightarrow D_I$	w/o DA	96.5	96.2	93.1	70.4	91.4	71.2	69.2	40.1	51.6	32.4	42.2	45.7	94.2
	w SF-DA	98.3	97.2	96.4	76.3	95.2	82.2	72.6	43.8	62.9	37.1	56.2	66.2	<b>96.1</b>
$D_M \rightarrow D_S$	w/o DA	93.2	92.1	91.4	83.3	71.7	81.4	46.3	68.3	34.6	23.2	33.1	34.0	92.2
	w SF-DA	96.8	94.2	95.3	87.7	77.5	89.5	51.1	78.2	43.2	29.0	38.9	34.0	<b>94.7</b>
$D_M \rightarrow D_L$	w/o DA	86.8	96.3	95.7	72.1	76.3	67.1	41.9	29.0	25.0	30.3	79.2	41.2	88.9
	w SF-DA	96.8	96.4	91.2	77.8	89.0	72.3	46.2	36.2	28.0	34.0	76.2	48.2	<b>94.2</b>
$D_I \rightarrow D_M$	w/o DA	94.1	97.2	92.3	87.4	67.9	72.5	39.0	40.2	39.8	42.0	46.2	53.6	94.6
	w SF-DA	99.1	98.4	96.0	80.6	73.3	81.0	40.5	45.1	49.1	29.1	59.0	56.2	<b>96.5</b>
$D_S \rightarrow D_M$	w/o DA	92.4	97.3	96.3	79.7	90.4	85.4	52.5	38.7	24.0	39.0	43.1	63.2	92.9
	w SF-DA	95.1	98.1	96.4	88.7	94.3	85.8	64.2	46.2	27.6	50.5	47.8	71.4	<b>96.1</b>
$D_L \rightarrow D_M$	w/o DA	85.2	93.2	92.4	82.5	58.8	59.0	26.1	30.0	29.2	23.2	22.0	23.0	88.3
	w SF-DA	92.0	95.9	95.3	77.4	75.5	71.4	27.2	37.4	38.2	25.0	22.0	23.0	<b>95.4</b>

#### 4.3.4. Results on PTB and PTB\_XL

For experimental purposes, we divided the PTB database into two non-overlapping subsets: PTB1 and PTB2. Here, the PTB1 consists of 74 patient data, whereas PTB2 consists of 90 patient data in the PTB1  $\rightarrow$  PTB2 test case. Table 10 shows the results of each class adaptation. In the second case, PTB  $\rightarrow$  PTB\_XL, we utilized all 164 patient data from the PTB database, where PTB\_XL is used as a target set. Table 11 shows the detailed results of each class adaptation. Overall, both PTB1  $\rightarrow$  PTB2 and PTB  $\rightarrow$  PTB\_XL achieved an acceptable overall accuracy of 95.5% and 93.2%, respectively.

**Table 10.** Results on PTB<sub>1</sub>  $\rightarrow$  PTB<sub>2</sub>, cross-patient classification task.

MI-Sub Category	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)	Accuracy (%)
AMI	95.5	89.2	90.7	<b>95.5</b>
ASMI	97.2	93.3	93.0	
IMI	94.6	96.8	95.2	
ALMI	91.8	95.8	94.1	
ALMI	91.8	95.8	94.1	
HC	89.2	92.5	88.3	
Average	94.5	94.4	93.5	

**Table 11.** Results on PTB → PTB\_XL, cross-patient classification task.

MI-Sub Category	Sensitivity (%)	Positive Predictivity (%)	F1 Score (%)	Accuracy (%)
AMI	73.5	70.3	69.3	
ASMI	88.3	89.2	92.3	
IMI	88.2	92.9	86.2	
ALMI	71.3	73.6	74.3	<b>93.2</b>
ALMI	86.6	70.5	85.2	
HC	96.4	96.3	93.6	
Average	85.7	82.1	83.4	

#### 4.3.5. Comparison with State-of-Art Methods

Table 12 shows the comparison between our model with other methods. We consider all the results from [3] and compared the results with our method. All results in the table are based on only the MITDB database, as it is widely used by other works, and also convenient to use for UDA tasks. For a fair comparison, we utilized several traditional methods [37,41–46] methods to compare results. These traditional methods are based on interpatient and patient-specific settings. For comparison purposes, we used DS1 → DS2. These methods can be classified into interpatient, patient-specific, unsupervised domain adaptation, and source-free domain adaptation. In all settings, the results are considerably good in the ECG classification task. But the main challenge of generalization in unseen samples is still unfolding in traditional approaches. On the other hand, UDA methods can unfold this issue with its generalization capability in unseen domains. But it still lacks privacy in real-world settings. On the other hand, our proposed method SF-ECG outperformed the traditional-DL methods as well as UDA methods, even in critical new settings of UDA. Our method has many advantages, since it does not require a source set during training. So, it preserves privacy and generalization. This is considered a great advantage in many ECG classification applications.

**Table 12.** Comparison between recent state of the art with our source-free adaptation method and unsupervised domain adaptation method on MITDB DS2. All the results are taken from [3]. Here, we denote interpatient = IP, patient-specific = PS, unsupervised domain adaptation = UDA, and source-free domain adaptation = SFDA.

Methods	Types	N (%)			V (%)			S (%)			F (%)			Accuracy (%)
		Se	Pp	F1										
[37]	IP	99.2	95.2	90.0	93.9	90.9	92.0	91.1	42.2	58.0	-	-	-	93.8
[41]	IP	-	-	-	91.25	88.3	90.0	62.7	61.2	62.0	-	-	-	-
[42]	IP	98.9	97.4	98.0	85.7	94.1	89.0	76.5	76.6	76.0	25.0	1.79	0.0	96.4
[43]	IP	88.5	98.8	93.0	92.0	72.1	80.0	82.0	30.4	44.0	68.3	26.6	38.0	95.3
[44]	IP	91.8	98.9	95.0	95.1	90.1	92.0	<b>89.0</b>	35.4	50.0	32.2	20.3	25.0	91.4
[45]	PS	97.6	98.5	98.0	93.8	92.4	93.0	76.8	74.0	75.0	79.6	62.4	70.0	96.1
[46]	PS	99.7	97.2	98.0	91.8	<b>98.0</b>	95.0	61.4	<b>90.7</b>	73.0	9.0	28.3	14.0	97.0
[10]	UDA	-	-	-	-	-	-	-	-	-	-	-	-	93.0
[3]	UDA	99.1	98.4	99.0	94.0	92.3	93.0	76.5	90.2	83.0	57.9	74.1	65.0	97.6
[5]	UDA	99.4	98.0	98.7	91.4	87.2	89.2	57	88.6	69.4	37.3	47.6	41.8	96.8
<b>SF-ECG (Ours)</b>	<b>SFDA</b>	<b>99.4</b>	<b>99.2</b>	<b>99.7</b>	<b>96.4</b>	<b>97.2</b>	<b>98.2</b>	87.1	88.6	<b>89.4</b>	<b>81.7</b>	<b>86.9</b>	<b>82.8</b>	<b>98.4</b>

#### 4.4. Ablation Studies

##### 4.4.1. Effect of Part by Part Loss Function

In Tables 13 and 14, we show the ablation of the impact of the loss functions. We show the effect based on the MITDB database, considering two tasks: DS1 → DS2, and DS2 → DS1. In both cases, if we remove the adversarial loss  $\mathcal{L}_{adv}$ , the model does not converge. But if we use all the loss functions together, we can have a significant performance boost compared to not using all the loss functions. Each loss function has its own contribution to the overall performance of our framework. Figure 5 shows the effectiveness of using all loss functions together.

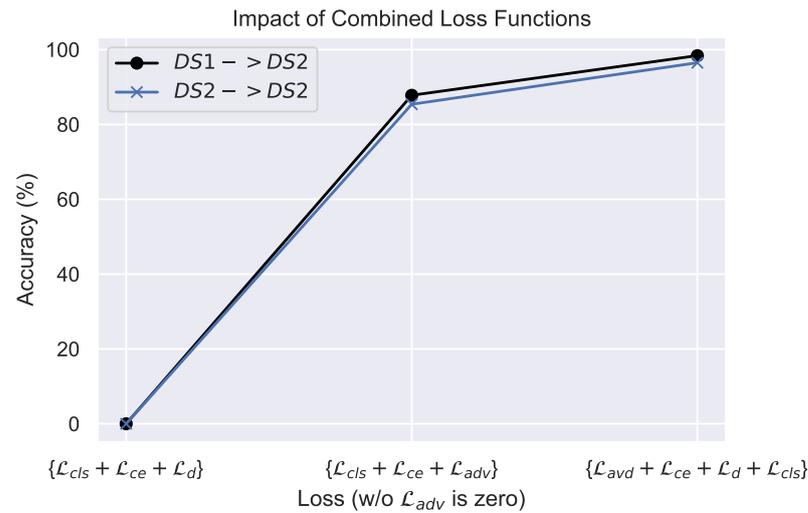


Figure 5. Ablation on the effect of loss functions.

Table 13. Ablation on the effect of loss functions. Here, we use the MITDB database. Here, results on DS1 → DS2 are presented.

$\mathcal{L}_{adv}$	$\mathcal{L}_{CE}$	$\mathcal{L}_D$	$\mathcal{L}_{cls}$	Accuracy (%)
-	✓	✓	✓	Not converge
✓	✓	-	✓	87.8
✓	✓	✓	✓	98.4

Table 14. Ablation on the effect of loss functions. Here, we use the MITDB database. Here, results on DS2 → DS1 are presented.

$\mathcal{L}_{adv}$	$\mathcal{L}_{CE}$	$\mathcal{L}_D$	$\mathcal{L}_{cls}$	Accuracy (%)
-	✓	✓	✓	Not converge
✓	✓	-	✓	85.4
✓	✓	✓	✓	96.5

##### 4.4.2. Effectiveness in Number of Generated Source Samples

We experiment to examine the effectiveness of our proposed framework on generated source samples. Table 15 shows the results of the effectiveness of our framework on the number of generated samples. We tested our model without generating synthetic samples on the S and F classes. As we can see, both with (source-free) and without domain adaptation, the model has a performance drop, whereas a higher number of samples leads to a better result in all possible cases.

**Table 15.** Results on the effectiveness of increasing the number of samples in adaptation task.

Class	No. of Samples	w/o UDA	SFDA
S	3712	92.2	<b>96.2</b>
	2764	81.2	<b>83.2</b>
F	1871	79.4	<b>86.2</b>
	801	77.6	<b>84.8</b>

#### 4.4.3. Generalization of Source Free Adaptation

Table 16 shows the generalization of source-free domain adaptation in different case scenarios. As we can see, not utilizing the UDA technique can hinder the performance by a large margin, whereas utilizing the UDA technique can increase the performance by a considerable margin. In all cases, we offer better performance than the traditional UDA method, even altering source-free training during the adaptation, which also offers many advantages over traditional UDA techniques.

**Table 16.** The accuracy of without/with UDA-based tasks in both MITDB, PTB, and PTB\_XL, which demonstrates the generalization and impact of our SFDA method.

Tasks	w/o UDA	UDA with Source	SFDA (Ours)
DS1 → DS2	91.7	94.0	<b>98.4</b>
DS2 → DS1	89.5	93.2	<b>96.5</b>
PTB_1 → PTB_2	61.4	93.2	<b>95.5</b>
PTB → PTB_XL	59.8	90.3	<b>93.2</b>

## 5. Conclusions

In this paper, we propose a novel domain adaptation technique called SF-ECG, and also devise a new domain adaptation task in ECG classification, which is source-free domain adaptation. Our framework is developed for interpatient adaptation tasks, mostly for arrhythmia classification. We design a framework that consists of three modules, namely, data adjustment, local clustering, and source-free adaptation. With a more balanced performance across multiple categories, we achieve results that are comparable to those of other contemporary arts. To efficiently improve the deep learning model, we use four loss functions to reduce the performance-degrading distribution discrepancies of various records. In the inference phase, our method does not require additional data or even sources, and does not necessitate the addition of annotations to new records. The proposed approach is adaptable to new data and has the potential to significantly enhance deep learning models' interpatient performance. Our approach is also capable of data privacy and generalizability in unseen samples. We obtain more stable results in minor categories where samples are deficient.

**Author Contributions:** Conceptualization, T.H.R.; methodology, T.H.R.; software, T.H.R. and Y.-W.K.; validation, T.H.R. and Y.-W.K.; investigation, T.H.R. and Y.-W.K.; writing—original draft preparation, T.H.R. and Y.-W.K.; visualization, T.H.R.; writing—review and editing, T.H.R. and Y.-W.K.; supervision, Y.-W.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2021R1F1A1064069).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

GAN	Generative Adversarial Networks
UDA	Unsupervised Domain Adaptation
SFDA	Source-Free Domain Adaptation

## References

1. Chen, M.; Wang, G.; Ding, Z.; Li, J.; Yang, H. Unsupervised Domain Adaptation for ECG Arrhythmia Classification. In Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 304–307. [\[CrossRef\]](#)
2. Deng, F.; Tu, S.; Xu, L. Multi-source unsupervised domain adaptation for ECG classification. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 854–859. [\[CrossRef\]](#)
3. Wang, G.; Chen, M.; Ding, Z.; Li, J.; Yang, H.; Zhang, P. Inter-patient ECG arrhythmia heartbeat classification based on unsupervised domain adaptation. *Neurocomputing* **2021**, *454*, 339–349. [\[CrossRef\]](#)
4. Yuan, S.; He, Z.; Zhao, J.; Yuan, Z.; Alhudhaif, A.; Alenezi, F. Hypergraph and cross-attention-based unsupervised domain adaptation framework for cross-domain myocardial infarction localization. *Inf. Sci.* **2023**, *633*, 245–263. [\[CrossRef\]](#)
5. He, Z.; Chen, Y.; Yuan, S.; Zhao, J.; Yuan, Z.; Polat, K.; Alhudhaif, A.; Alenezi, F.; Hamid, A. A novel unsupervised domain adaptation framework based on graph convolutional network and multi-level feature alignment for inter-subject ECG classification. *Expert Syst. Appl.* **2023**, *221*, 119711. [\[CrossRef\]](#)
6. Liu, X.; Yoo, C.; Xing, F.; Oh, H.; El Fakhri, G.; Kang, J.W.; Woo, J. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Trans. Signal Inf. Process.* **2022**, *11*, e25. [\[CrossRef\]](#)
7. Shaker, A.M.; Tantawi, M.; Shedeed, H.A.; Tolba, M.F. Generalization of convolutional neural networks for ECG classification using generative adversarial networks. *IEEE Access* **2020**, *8*, 35592–35605. [\[CrossRef\]](#)
8. Kundu, J.N.; Venkat, N.; Babu, R.V. Universal source-free domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4544–4553.
9. Yang, S.; Wang, Y.; Van De Weijer, J.; Herranz, L.; Jui, S. Generalized source-free domain adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 8978–8987.
10. Bazi, Y.; Alajlan, N.; AlHichri, H.; Malek, S. Domain adaptation methods for ECG classification. In Proceedings of the International Conference on Computer Medical Applications (ICCM), Sousse, Tunisia, 20–22 January 2013; pp. 1–4.
11. Niu, L.; Chen, C.; Liu, H.; Zhou, S.; Shu, M. A deep-learning approach to ECG classification based on adversarial domain adaptation. *Healthcare* **2020**, *8*, 437. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Yin, W.; Yang, X.; Li, L.; Zhang, L.; Kitsuwon, N.; Shinkuma, R.; Oki, E. Self-adjustable domain adaptation in personalized ECG monitoring integrated with IR-UWB radar. *Biomed. Signal Process. Control* **2019**, *47*, 75–87. [\[CrossRef\]](#)
13. He, W.; Ye, Y.; Li, Y.; Pan, T.; Lu, L. Online Cross-subject Emotion Recognition from ECG via Unsupervised Domain Adaptation. In Proceedings of the 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Mexico City, Mexico, 1–5 November 2021; pp. 1001–1005.
14. Natarajan, A.; Angarita, G.; Gaiser, E.; Malison, R.; Ganesan, D.; Marlin, B.M. Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ECG. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 875–885.
15. Hang, W.; Feng, W.; Du, R.; Liang, S.; Chen, Y.; Wang, Q.; Liu, X. Cross-subject EEG signal recognition using deep domain adaptation network. *IEEE Access* **2019**, *7*, 128273–128282. [\[CrossRef\]](#)
16. Ammour, N. Atrial fibrillation detection with a domain adaptation neural network approach. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018; pp. 738–743.
17. Helm, H.S.; Yang, W.; De Silva, A.; Vogelstein, J.T.; Priebe, C.E. Approximately optimal domain adaptation with Fisher’s Linear Discriminant Analysis. *arXiv* **2023**, arXiv:2302.14186.
18. Carrera, D.; Rossi, B.; Fragneto, P.; Boracchi, G. Domain adaptation for online ecg monitoring. In Proceedings of the IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; pp. 775–780.
19. Chen, M.; Wang, G.; Chen, H.; Ding, Z. Adaptive region aggregation network: Unsupervised domain adaptation with adversarial training for ECG delineation. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtually, 4–8 May 2020; pp. 1274–1278.
20. Yamaç, M.; Duman, M.; Adalıoğlu, İ.; Kiranyaz, S.; Gabbouj, M. A Personalized Zero-Shot ECG Arrhythmia Monitoring System: From Sparse Representation Based Domain Adaption to Energy Efficient Abnormal Beat Detection for Practical ECG Surveillance. *arXiv* **2022**, arXiv:2207.07089.

21. Ye, Y.; Luo, T.; Huang, W.; Sun, Y.; Li, L. ECG-based Cross-Subject Mental Stress Detection via Discriminative Clustering Enhanced Adversarial Domain Adaptation. In Proceedings of the 16th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 21–24 October 2022; Volume 1, pp. 495–499.
22. Li, H.; Zhang, Y.; Han, J.; Yan, Y.; Liu, Y.; Yang, H. AdapSQA: Adaptive ECG Signal Quality Assessment Model for Inter-Patient Paradigm using Unsupervised Domain Adaptation. In Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, NV, USA, 6–8 December 2022; pp. 3378–3384.
23. Peng, P.; Xie, L.; Zhang, K.; Zhang, J.; Yang, L.; Wei, H. Domain adaptation for epileptic EEG classification using adversarial learning and Riemannian manifold. *Biomed. Signal Process. Control* **2022**, *75*, 103555. [[CrossRef](#)]
24. Golany, T.; Radinsky, K.; Freedman, D. SimGANs: Simulator-based generative adversarial networks for ECG synthesis to improve deep ECG classification. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 3597–3606.
25. Golany, T.; Radinsky, K. Pgens: Personalized generative adversarial networks for ecg synthesis to improve patient-specific deep ecg classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 557–564.
26. Golany, T.; Lavee, G.; Yarden, S.T.; Radinsky, K. Improving ECG classification using generative adversarial networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13280–13285.
27. Wang, P.; Hou, B.; Shao, S.; Yan, R. ECG arrhythmias detection using auxiliary classifier generative adversarial network and residual network. *IEEE Access* **2019**, *7*, 100910–100922. [[CrossRef](#)]
28. Yang, H.; Liu, J.; Zhang, L.; Li, Y.; Zhang, H. Proegan-ms: A progressive growing generative adversarial networks for electrocardiogram generation. *IEEE Access* **2021**, *9*, 52089–52100. [[CrossRef](#)]
29. Ye, F.; Zhu, F.; Fu, Y.; Shen, B. ECG generation with sequence generative adversarial nets optimized by policy gradient. *IEEE Access* **2019**, *7*, 159369–159378. [[CrossRef](#)]
30. Zhu, F.; Ye, F.; Fu, Y.; Liu, Q.; Shen, B. Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network. *Sci. Rep.* **2019**, *9*, 6734. [[CrossRef](#)]
31. Wang, X.; Chen, B.; Zeng, M.; Wang, Y.; Liu, H.; Liu, R.; Tian, L.; Lu, X. An ECG Signal Denoising Method Using Conditional Generative Adversarial Net. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 2929–2940. [[CrossRef](#)]
32. Haradal, S.; Hayashi, H.; Uchida, S. Biosignal data augmentation based on generative adversarial networks. In Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 368–371.
33. Singh, P.; Pradhan, G. A new ECG denoising framework using generative adversarial network. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *18*, 759–764. [[CrossRef](#)]
34. Adib, E.; Afghah, F.; Prevost, J.J. Synthetic ecg signal generation using generative neural networks. *arXiv* **2021**, arXiv:2112.03268.
35. Rafi, T.H.; Ko, Y.W. HeartNet: Self Multihead Attention Mechanism via Convolutional Network with Adversarial Data Synthesis for ECG-Based Arrhythmia Classification. *IEEE Access* **2022**, *10*, 100501–100512. [[CrossRef](#)]
36. Kurmi, V.K.; Subramanian, V.K.; Namboodiri, V.P. Domain impression: A source data free domain adaptation method. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 615–625.
37. Huang, H.; Liu, J.; Zhu, Q.; Wang, R.; Hu, G. A new hierarchical method for inter-patient heartbeat classification using random projections and RR intervals. *Biomed. Eng. Online* **2014**, *13*, 90. [[CrossRef](#)] [[PubMed](#)]
38. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [[CrossRef](#)]
39. Goldberger, A.L.; Amaral, L.A.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **2000**, *101*, e215–e220. [[CrossRef](#)]
40. Wagner, P.; Strodthoff, N.; Bousseljot, R.D.; Kreiseler, D.; Lunze, F.I.; Samek, W.; Schaeffter, T. PTB-XL, a large publicly available electrocardiography dataset. *Sci. Data* **2020**, *7*, 154. [[CrossRef](#)] [[PubMed](#)]
41. Guo, L.; Sim, G.; Matuszewski, B. Inter-patient ECG classification with convolutional and recurrent neural networks. *Biocybern. Biomed. Eng.* **2019**, *39*, 868–879. [[CrossRef](#)]
42. Niu, J.; Tang, Y.; Sun, Z.; Zhang, W. Inter-patient ECG classification with symbolic representations and multi-perspective convolutional neural networks. *IEEE J. Biomed. Health Inform.* **2019**, *24*, 1321–1332. [[CrossRef](#)] [[PubMed](#)]
43. Sellami, A.; Hwang, H. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Syst. Appl.* **2019**, *122*, 75–84. [[CrossRef](#)]
44. Li, F.; Xu, Y.; Chen, Z.; Liu, Z. Automated heartbeat classification using 3-D inputs based on convolutional neural network with multi-fields of view. *IEEE Access* **2019**, *7*, 76295–76304. [[CrossRef](#)]

45. Zhai, X.; Tin, C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access* **2018**, *6*, 27465–27472. [[CrossRef](#)]
46. Ye, C.; Kumar, B.V.; Coimbra, M.T. An automatic subject-adaptable heartbeat classifier based on multiview learning. *IEEE J. Biomed. Health Inform.* **2015**, *20*, 1485–1492. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.