Adversarial Spatiotemporal Contrastive Learning for Electrocardiogram Signals

Ning Wang, *Member, IEEE*, Panpan Feng, *Member, IEEE*, Zhaoyang Ge, Yanjie Zhou, *Member, IEEE*, Bing Zhou, and Zongmin Wang

***Abstract*— Extracting invariant representations in unlabeled electrocardiogram (ECG) signals is a challenge for deep neural networks (DNNs). Contrastive learning is a promising method for unsupervised learning. However, it should improve its robustness to noise and learn the spatiotemporal and semantic representations of categories, just like cardiologists. This article proposes a patient-level adversarial spatiotemporal contrastive learning (ASTCL) framework, which includes ECG augmentations, an adversarial module, and a spatiotemporal contrastive module. Based on the ECG noise attributes, two distinct but effective ECG augmentations, ECG noise enhancement, and ECG noise denoising, are introduced. These methods are beneficial for ASTCL to enhance the robustness of the DNN to noise. This article proposes a self-supervised task to increase the antiperturbation ability. This task is represented as a game between the discriminator and encoder in the adversarial module, which pulls the extracted representations into the shared distribution between the positive pairs to discard the perturbation representations and learn the invariant representations. The spatiotemporal contrastive module combines spatiotemporal prediction and patient discrimination to learn the spatiotemporal and semantic representations of categories. To learn category representations effectively, this article only uses patient-level positive pairs and alternately uses the predictor and the stop-gradient to avoid model collapse. To verify the effectiveness of the proposed method, various groups of experiments are conducted on four ECG benchmark datasets and one clinical dataset compared with the state-of-the-art methods. Experimental results showed that the proposed method outperforms the state-of-the-art methods.**

***Index Terms*— Adversarial learning, contrastive learning, data augmentation, electrocardiogram (ECG).**

NOMENCLATURE

Notation	Definition
x	Single instance ECG original data.
$\widetilde{x}$	Augmented view of instance.
N, C, and T	Number of instances, number of leads, and length of signals.
e	Group of instances under the same patient.
u	Number of elements in group e.
a and d	ECG noise enhancement tag and ECG noise denoising tag.
μ	Ratio of signal to noise.
z	Representation extracted by encoder $f_E(\bullet)$.
$\widehat{z}$ and $\widetilde{z}$	No-gard representation and recombined representation.
y	Pseudolabel of representation.
K and D	Length of representation and dimension of the encoder.
τ	Length of the past segment in representation.
sv	Semantic vector extracted by transformer $f_T(\bullet)$.
M	Dimension of the transformer.
p	Projection extracted by projector $h_P(\bullet)$.
q	Prediction extracted by predictor $h_Q(\bullet)$.
H	Dimension of the projector.
w_1, w_2, and w_3	Weight of loss function.

I. INTRODUCTION

REDUCING cardiovascular diseases (CVDs) mortality is one of the nine global noncommunicable disease goals proposed by the World Health Organization [1]. Electrocardiogram (ECG) is a crucial clinical medical detection tool for CVDs. More than 300 million ECGs are obtained every year worldwide [2]. The paroxysm from CVDs causes a change in ECG signals that can be detected by a trained deep neural network (DNN) [3]. Nevertheless, the performance of a DNN inevitably depends on the quality and quantity of the labeled data, but annotating ECG signals are very time and capital consuming [4]. Therefore, a DNN has a weak generalization ability when there is limited labeled data. To tackle this, a large amount of unlabeled data must be effectively utilized in the clinical application of intelligent diagnosis.

Self-supervised learning can learn useful representations in unlabeled data with pretext tasks [5], such as jigsaw puzzles [6], rotation prediction [7], multiview association [8], instance discrimination [9], and so on. Contrastive learning is a promising self-supervised learning method [10] that generates views of input instances through data augmentation and considers views from the same instance as positive pairs

Manuscript received 1 September 2022; revised 17 March 2023; accepted 19 April 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1401200, in part by the Key Research, Development, and Dissemination Program of Henan Province (Science and Technology for the People) under Grant 182207310002, and in part by the Key Science and Technology Project of Xinjiang Production and Construction Corps under Grant 2018AB017. *(Corresponding author: Yanjie Zhou.)*

Ning Wang, Panpan Feng, Zhaoyang Ge, Bing Zhou, and Zongmin Wang are with the School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450000, China (e-mail: wning@ha.edu.cn).

Yanjie Zhou is with the School of Management, Zhengzhou University, Zhengzhou 450000, China (e-mail: ieyjzhou@zzu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3272153.

Digital Object Identifier 10.1109/TNNLS.2023.3272153

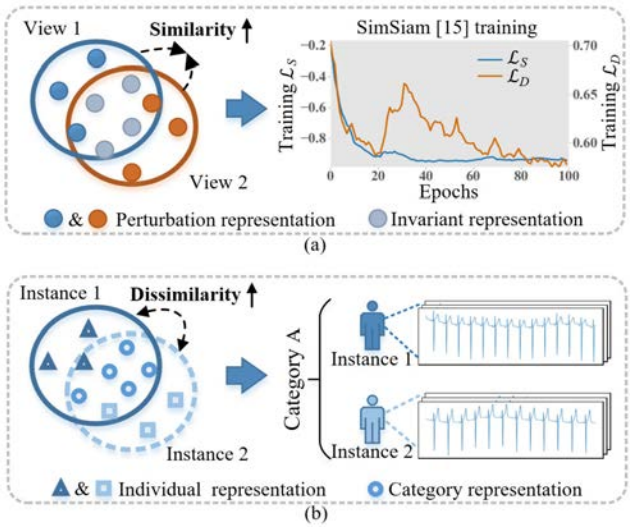


Fig. 1. Research Motivation. (a) In the process of learning invariant representations between positive pairs, the learned representations also include perturbation representations caused by data augmentations. (b) In the process of maximizing the dissimilarity of negative pairs, using negative pairs of the same category is disadvantageous to learn the category invariant representations.

and views from different instances as negative pairs. Invariant representations are learned by maximizing the agreement of positive pairs and minimizing the agreement of negative pairs [11]. An ECG signal, whose lead reflects the activity of each part of the heart, is a physiological temporal signal (PTS) that is often accompanied by noise [12]. Clinically, cardiologists pay attention to the temporal changes of each ECG lead and judge the CVDs by observing invariant category features of the waveform. Even when noise exists, CVDs can still be diagnosed through these invariant category features. Thus, we believe that a DNN should, like cardiologists, on the one hand, *improve its robustness to noise for learning invariant representations*, and on the other hand, *learn spatiotemporal and semantic representations of categories*. Nevertheless, to achieve these two goals, there following issues in contrastive learning should be addressed.

1) Are previously proposed augmentation methods suitable for enhancing the robustness of DNN to noise? Choosing an appropriate augmentation method is essential for the success of contrastive learning [13]. ECG noise is one of the key factors in reducing the performance of the DNN [14]. Existing augmentation methods are not customized based on ECG noise, and some even change the shape of the signal or disrupt the sequence, which is not conducive to improving noise robustness.

2) Do the invariant representations obtained by contrastive learning contain perturbation representations? In Fig. 1(a), the two curves $\mathcal{L}_D$ and $\mathcal{L}_S$ represent the training processes of the loss function of the augmentation discriminator and contrastive learning framework (e.g., SimSiam [15]) on the Chapman dataset. When $\mathcal{L}_S$ has converged, the gradient of $\mathcal{L}_D$ is still declining. This means that although the agreement of the positive pair is maximized, the perturbation representations caused by augmentation are also captured, which will weaken the antiperturbation ability of the DNN.

3) Are the ECG negative pairs in contrastive learning reasonable for learning category representation? Some contrastive learning studies [13], [16] have focused on spatiotemporal and semantic representations. They encourage the negative pairs to be dissimilar to avoid model collapse. However, as shown in Fig. 1(b), when two instances of negative pairs belong to the same category, their separation prevents the DNN from learning category representations. For better learning spatiotemporal and semantic representations of categories, abandoning negative pairs is promising.

To address the above-mentioned issues, this article proposes a patient-level adversarial spatiotemporal contrastive learning (ASTCL) framework to learn useful invariant category representations from unlabeled ECG data. According to ECG noise and common filters, two distinct but effective ECG augmentations for ECG signals, including ECG noise enhancement and ECG noise denoising, are proposed. The proposed two augmentation methods could reduce the impact of ECG noise on representation learning and preserve the properties of space-time and semantics of waveforms simultaneously, which are conducive to improving noise robustness and learning invariant representations. To improve the antiperturbation ability of the DNN, a self-supervised task is developed in the adversarial module, which is a game between a discriminator and an encoder. The discriminator is used to identify which augmentation generates views, and the encoder is used to learn invariant representations. By completing this task, the extracted representations of the positive pair can be pulled into the shared representation distribution to avoid learning perturbation representations and promote learning invariant representations. To learn spatiotemporal and semantic representations of categories, we construct a spatiotemporal contrastive module that employs spatiotemporal prediction and patient discrimination as pretext tasks. This module only utilizes patient-level positive pairs to better extract the category representations. Predictor and stop-gradient are alternately performed in contrastive branches to replace the role of negative pairs in preventing model collapse.

The main contributions of this article are summarized as follows.

1) This article proposes a novel patient-level contrastive learning framework ASTCL for unsupervised representation learning, which can, like cardiologists, improve the robustness of the model to noise and learn the spatiotemporal and semantic representations of categories.

2) Two distinct but effective augmentation methods are introduced by utilizing the ECG noise enhancement and denoising to facilitate the noise robustness. Based on adversarial learning, a self-supervised task is proposed, which can improve the antiperturbation ability by introducing a game between a discriminator and an encoder.

3) Spatiotemporal prediction and patient discrimination are used as pretext tasks to learn the spatiotemporal and semantic representations of categories. The task only uses patient-level positive pairs and alternately uses the predictor and the stop-gradient to avoid model collapse, which can better extract category representations.

4) Various experiments are conducted to verify learned representations with ASTCL on four ECG benchmark datasets and one clinical dataset. The experimental results show that ASTCL outperforms the state-of-the-art methods.

II. Related Work

This section reviews the previous related to contrastive learning and self-supervised learning for PTS. The details are summarized as follows.

A. Contrastive Learning

The pretext tasks of contrastive learning research mainly include time series prediction and instance discrimination. Hyvärinen and Morioka [17] divided the time series data according to the time window and predicted the position of these segments to find effective representations. To extract a general representation of variable length and multivariate time series, Franceschi et al. [18] built a novel triple loss based on temporal negative sampling and used an encoder based on causal expansion convolution to learn universal embeddings of time series. Oord et al. [19] proposed a temporal contrastive learning framework, which used the past segment to predict the future segment via an autoregressive model and learned the representations of data by maximizing mutual information between prediction and actuality. Although these studies can learn temporal representations well, they are ineffective in semantic representation learning.

Contrastive learning based on instance discrimination is more suitable for learning semantic representations. Hjelm et al. [20] constructed positive and negative pairs on the sample's local and global features to maximize the mutual information. Recently, some methods with data augmentations can better learn semantic representations by increasing the difficulty of instance recognition tasks, such as the studies of MoCo [11] and simple framework for contrastive learning of visual representations (SimCLR) [21]. Tian et al. [22] improved the efficiency of comparison by expanding the sample views to increase comparison pairs. Jiang et al. [23] and Ho and Vasconcelos [24] added unsupervised adversarial training in contrastive learning, which generated adversarial examples to attack the model for increasing the model's robustness. Zhu et al. [25] proposed a novel feature-level data operation to replace the augmentation, resulting in facilitating representation learning. Caron et al. [26] and Xu et al. [27] abandoned the traditional pairwise comparison. Their methods clustered the data and compulsorily unified different views produced by cluster assignments. Liu et al. [28] proposed a GNN-based contrastive learning framework from high-dimensional attributes and local structure and measured the agreement of each instance pair with its outputted scores for graph anomaly detection. This framework can capture the relationship between each node and its neighboring substructure in an unsupervised way. Kermiche [29] combined the back propagation and contrastive Hebbian learning into the Boltzmann machines for applying to networks when optimizing a loss objective and networks with stochastic binary outputs. Grill et al. [30] and Chen et al. [15] achieved state-of-the-art performance without establishing negative pairs by the "stop grad" of the encoder. The above-mentioned methods are without considering the learning of temporal representation.

ECG signals belong to PTS, and besides semantic representations, their spatiotemporal representations should also deserve attention in representation learning [31]. In addition, data augmentations of the above-mentioned studies are unsuitable for ECG signals.

B. Self-Supervised Learning for PTS

With the continuous progress of self-supervised learning technology, more and more researchers pay more attention to self-supervised learning for PTS. Sarkar and Etemad [32] extended six ECG transformation methods to learn the high-level representations of data by identifying different transformations. Yet, it could also learn transformation representations. Banville et al. [33], [34] proposed two kinds of pretext tasks: relative positioning and temporary shuffling for EEG classification. These heuristic tasks may limit the generality of the learned representations. Fan et al. [35], [36] intercepted the EEG data into segments and learned the data representations by predicting the relation of time series segments. The heartbeats of ECG signals are approximately periodic. The task of predicting relation is not suitable for ECG signals. Zhao et al. [37] matched the feature distribution offset between the source domain and the target domain through the adversarial learning strategy to reduce the distribution difference between related but different domains. Even with fewer available tags, it can still improve the performance of a single-subject performance in the target domain. Ma et al. [38] combined time series prediction and clustering and employed spectral analysis to constrain the pseudolabels and align the predicted labels with the pseudolabels to optimize the self-supervised training.

Due to the outstanding performance of contrastive learning, more unsupervised representation learning research of PTS improves on this basis. Cheng et al. [39] utilized contrastive learning and adversarial training to learn the invariant representation of subject specific. Alsentzer et al. [40] performed contrastive learning on the fused features of multilead to learn the sequential representation of EEG, but it also dramatically increases the amount of calculation. Shen et al. [41] added the idea of alignment intersubject in contrastive learning, maximizing the similarity in EEG signals across subjects when they received the same stimulus. Lan et al. [42] used the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) module to judge whether the heartbeat of the intrasubject is "mutated" to optimize the definition of heartbeat-level positive and negative pairs and performed the intersubject task to learn the heartbeat representations. Eldele et al. [13] proposed a contrastive learning framework, which learns the temporal and contextual representations of EEG via the temporal contrastive module and contrastive contextual module. Kiyasseh et al. [16] expanded the positive and negative pairs of samples from the perspectives of time, space, and patients. They assume that the multilead of ECG is spatially invariant, but some CVDs not reflected in all leads, such as myocardial infarction [43].

The mentioned studies have not specially designed data augmentations based on ECG noise, and there is no way to prevent the model from learning perturbation representations. Moreover, these PTS studies still use negative pairs for pretraining. To solve the above-mentioned problems, this article proposes ECG augmentations, an adversarial module, and a spatiotemporal contrastive module with only patient-level positive pairs.

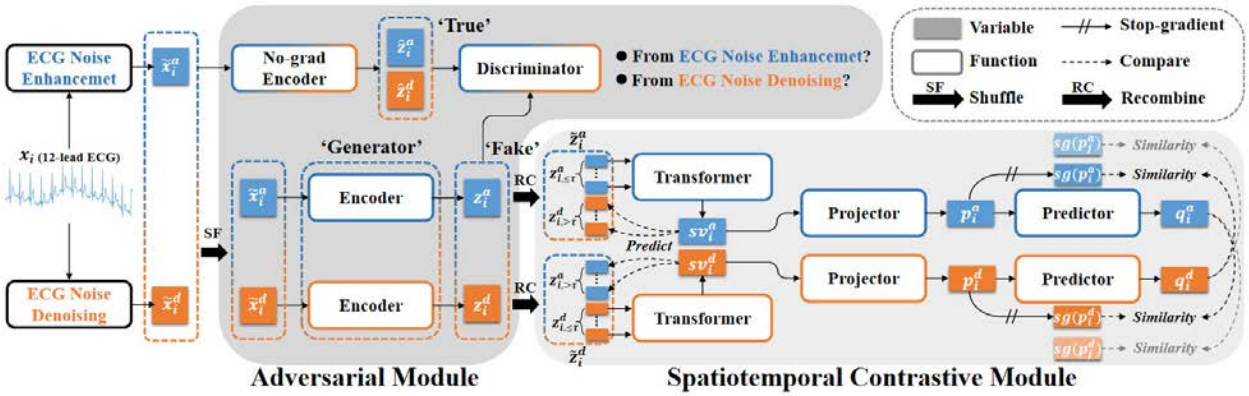


Fig. 2. ASTCL Architecture. The 12-lead ECG signal x_i is randomly transformed by the proposed ECG augmentations to produce views $\widehat{x}_i^a$ or $\widehat{x}_i^d$. The adversarial module defines $\widehat{z}_i^a$ and $\widehat{z}_i^d$ as the "True" set, z_i^a and z_i^d as the "Fake" set, which aims to promote z_i^a and z_i^d into the shared representation distribution via gaming between discriminator and encoder. The spatiotemporal contrastive module uses past segment $z_{i,\leq\tau}^a$ and $z_{i,\leq\tau}^d$ to predict the future segment $z_{i,>\tau}^d$ and $z_{i,>\tau}^a$ for spatiotemporal representation learning, and maximize the similarity between projection and prediction in the same patient [i and j are the same patients and maximize the similarity of q_i^a and $sg(p_i^d)$ and q_i^a, and $sg(p_j^d)$] for semantic representation learning.

III. Method

The overview of our proposed ASTCL is introduced in Section III-A. The details of ECG augmentations, adversarial module, and spatiotemporal contrastive module are explained in Sections III-B–III-D, respectively. The complexity of ASTCL is analyzed in section III-E. The notations used in this article are summarized in the Nomenclature.

A. Overview

Each lead ECG records the electric potential changes of the corresponding parts of the heart over a period [44]. In this article, the most commonly used in the clinic, a 12-lead 10-s ECG signal, is used as input data. The dataset is expressed as $X = \{x_1, \ldots, x_i, \ldots, x_n\} \in \mathbb{R}^{N*C*T}$, where N is the total number of instances, C is the number of leads, and T is the length of signals. ASTCL is composed of ECG augmentations, an adversarial module, and a spatiotemporal contrastive module, specifically including an ECG noise enhancement $A_a(\bullet)$, an ECG noise denoising $A_d(\bullet)$, an encoder $f_E(\bullet)$, a no-grad encoder $f_{\widehat{E}}(\bullet)$, a discriminator $h_D(\bullet)$, a transformer $f_T(\bullet)$, a projector $h_P(\bullet)$ and a predictor $h_Q(\bullet)$. The overall structure of ASTCL is illustrated in Fig. 2. By using our proposed ECG augmentations, the input data x_i is transformed into two distinct augmented views, which are input into the adversarial module and spatiotemporal contrastive module. The adversarial module aims to promote the extracted representations of two views by the encoders into the shared distribution to increase the antiperturbation ability of the encoder. The spatiotemporal contrastive module aims to learn spatiotemporal and semantic representations of categories. It only uses patient-level positive pairs to learn category representations more effectively. After unsupervised pretraining, the trained encoder of ASTCL can be used for downstream classification tasks.

B. ECG Augmentations

ECG noise is one of the key factors in reducing the performance of representation learning in ECG [14], mainly including baseline drift (BD) noise, muscle artifacts (MA) noise, and power frequency (PF) noise [45]. Nevertheless, the existing PTS augmentation methods include waveform flip, channel resize, random resized crop, and so on [16], [46]. These methods are not conducive to improving the robustness of DNN to ECG noise. Therefore, we expend ECG noise enhancement as one of the ECG augmentations by adding BD, MA, and PF noise into the data. In contrast, we take ECG noise denoising as the other ECG augmentations by removing the original noise of data. The experimental data are converted by ECG augmentations lead by lead. To reduce the effect of amplitude differences caused by the instrument and individual, the ECG signal is processed by Z-score normalization [47] before ECG augmentations. The processed signal is shown in Fig. 3(a).

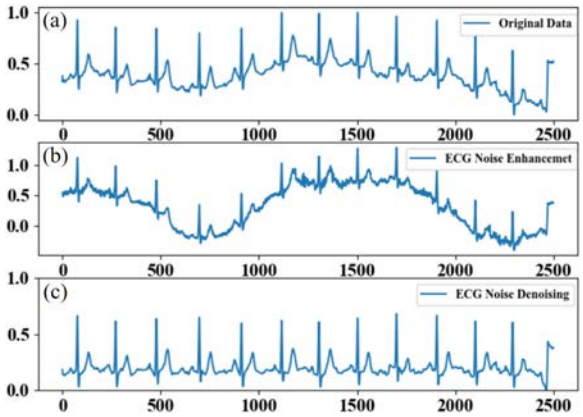


Fig. 3. ECG augmentations cases. (a) Original data. (b) Noisy view generated by ECG noise enhancement. (c) Clean view generated by ECG noise denoising.

In ECG noise enhancement, the MIT-BIH noise stress test database is used as a noise source. This database includes three half-hour recordings of noise typical in two-lead ambulatory ECG recordings, and the sampling rate is 360 Hz [48]. We resample the BD and MA noise of MIT-BIH to the same as the frequency of input data. To ensure the noise difference between leads, the randomly 10-s BD noise and MA noise from any lead are selected as $AB(t)_c$ and $AM(t)_c$ and $t \in T$ and $c \in C$, respectively. Since there is no PF

noise in MIT-BIH, this article generates the PF noise $AP(t)_c$ based on the modeling method of the literature [49]. We add $AB(t)_c$, $AM(t)_c$, and $AP(t)_c$ to the cth lead original ECG signals $x(t)_{i,c}$ at every moment in (1). Next, we repeat the above-mentioned operations C times to obtain the noisy view $\widetilde{x}_i^a$ in (2). The view $\widetilde{x}_i^a$ is shown in Fig. 3(b), where μ is the signal-to-noise ratio (SNR)

$$A_a(x(t)_{i,c}) = x(t)_{i,c} + \mu AB(t)_c + \mu AM(t)_c + \mu AP(t)_c \tag{1}$$

$$\widetilde{x}_i^a = \begin{bmatrix} A_a(x(1)_{i,1}) & \cdots & A_a(x(T)_{i,1}) \\ \vdots & \ddots & \vdots \\ A_a(x(1)_{i,C}) & \cdots & A_a(x(T)_{i,C}) \end{bmatrix}. \tag{2}$$

In ECG noise denoising, according to the study of [50] and [51], Butterworth (BW) filter, finite impulse response (FIR) filter, and infinite impulse response (IIR) filter are utilized to remove the original noise that may exist in the data. First, we employ the 0.5-Hz BW low-pass filter to extract BD noise as $DB(t)_c$ and remove it from the data. The MA noise is white noise with a frequency higher than 60 Hz. We use the 60-Hz FIR high-pass filter to delete the MA noise $DM(t)_c$ of data. Moreover, the 50-Hz IIR notch filter is used to remove PF noise, which is denoted as $DP(t)_c$. Then, we remove $DB(t)_c$, $DM(t)_c$, and $DP(t)_c$ from the original signal $x(t)_{i,c}$ in (3). Finally, as shown in Fig. 3(c), the clean view $\widetilde{x}_i^d$ are generated by repeating denoising operations, which is formulated as follows:

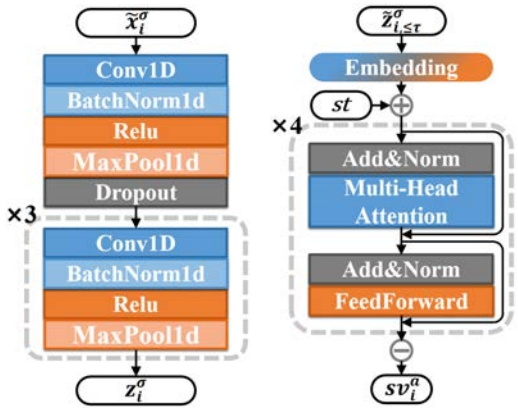
$$A_d(x(t)_{i,c}) = x(t)_{i,c} - DB(t)_c - DM(t)_c - DP(t)_c \tag{3}$$

$$\widetilde{x}_i^d = \begin{bmatrix} A_d(x(1)_{i,1}) & \cdots & A_d(x(T)_{i,1}) \\ \vdots & \ddots & \vdots \\ A_d(x(1)_{i,C}) & \cdots & A_d(x(T)_{i,C}) \end{bmatrix}. \tag{4}$$

After ECG noise enhancement and ECG noise denoising, the view $\widetilde{x}_i^a$ becomes noisier, and the other view $\widetilde{x}_i^d$ becomes cleaner. Although their differences in noise are more significant, their comparison helps to reduce the influence of ECG noise in contrastive learning, which is conducive to improving noise robustness for learning useful invariant representations by adversarial module and spatiotemporal contrastive module.

C. Adversarial Module

The mechanism of adversarial learning [52] is a game between the generator and the discriminator. Its goal is to make the discriminator unable to distinguish between the fake set and the true set. Zhao et al. [37] and Feng et al. [53] leverage this idea to obtain domain-invariant features across domains. Inspired by the above-mentioned studies, in the adversarial module, an adversarial game task via adversarial learning is proposed to reduce the gap between distribution $P(\widetilde{x}_i^a)$ and distribution $P(\widetilde{x}_i^d)$. This task aims to discard perturbation representations and learn invariant representations for enhancing the antiperturbation ability. As shown in Fig. 2, the adversarial module includes an encoder $f_E(\bullet)$, a no-grad encoder $f_{\widehat{E}}(\bullet)$, and a discriminator $h_D(\bullet)$. The structures of $f_E(\bullet)$ and $f_{\widehat{E}}(\bullet)$ are the same, and both are four-block



(a) (b)

Fig. 4. Architecture of main components. (a) Encoder. (b) Transformer.

convolutional architecture in Fig. 4(a), whose weights are shared. We employ encoder $f_E(\bullet)$ as the "Generator," and define a pseudolabel y^σ for $\widehat{z}_i^\sigma = f_{\widehat{E}}(\widetilde{x}_i^\sigma) \in \mathbb{R}^{K*D}, \sigma \in \{a, d\}$ as follows in (5), where K is the feature-length, and D is the hidden dimension of the encoder. The pseudolabels are divided into two categories: "0" means $\sigma = a$ and "1" means $\sigma = d$. The discriminator $h_D(\bullet)$ is a classifier with four linear layers, it conducts self-supervised training based on pseudolabels to identify which augmentations the view comes from

$$y^\sigma = \begin{cases} 0, & (if\ \sigma = a) \\ 1, & (if\ \sigma = d). \end{cases} \tag{5}$$

In classic adversarial learning, the encoder and discriminator play the following two-player minimax game with a value function $V(f_E, h_D)$ in [54, (eq. 6)]. Based on pseudolabels, if $\widehat{z}_i^d$ and $f_E(\widetilde{x}_i^a)$ are defined as the true set and the fake set, respectively, then $P(\widetilde{x}_i^a)$ will approximate to $P(\widehat{z}_i^d)$, which only allows the encoder to learn the representations of $\widetilde{x}_i^d$, not the invariant representations between views $\widetilde{x}_i^a$ and $\widetilde{x}_i^d$

$$\min_{f_E}\max_{h_D} V(f_E, h_D) = \mathbb{E}_{\widehat{z}_i^d \sim P(\widehat{z}_i^d)}[\log h_D(\widehat{z}_i^d)] + \mathbb{E}_{\widetilde{x}_i^a \sim P(\widetilde{x}_i^a)}[\log(1 - h_D(f_E(\widetilde{x}_i^a)))]. \tag{6}$$

To overcome this, the representations $\widehat{z}_i^a$ and $\widehat{z}_i^d$ are defined as the true set, and $f_E(\widetilde{x}_i^a))$ and $f_E(\widetilde{x}_i^d)$ are defined as the fake set. During the training process of the discriminator $h_D(\bullet)$, the gradient of the no-grad encoder $f_{\widehat{E}}(\bullet)$ is not updated. $\widehat{z}_i^a$ and $\widehat{z}_i^d$ are input into the discriminator $h_D(\bullet)$, and the discriminator $h_D(\bullet)$ predicts the probabilities $h_D(\widehat{z}_i^a)$ and $h_D(\widehat{z}_i^d)$ that $\widehat{z}_i^a$ and $\widehat{z}_i^d$ belong to pseudolabel categories. The parameters of discriminator $h_D(\bullet)$ are updated to minimize $\mathcal{L}_D$ in training process. The $\mathcal{L}_D$ is defined in the following, which includes two binary cross entropy (BCE) functions:

$$\mathcal{L}_D = \frac{1}{2N}\sum_{i=1}^{N}\left[\mathrm{BCE}\left(h_D\left(\widehat{z}_i^d\right), y^d\right) + \mathrm{BCE}\left(h_D\left(\widehat{z}_i^a\right), y^a\right)\right] = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log h_D\left(\widehat{z}_i^d\right) + \log\left(1 - h_D\left(\widehat{z}_i^a\right)\right)\right]. \tag{7}$$

In the process of training the encoder $f_E(\bullet)$, the parameters of discriminator $h_D(\bullet)$ are fixed, and its input is $f_E(\widetilde{x}_i^a)$

and $f_E(\widetilde{x}_i^d)$. The pseudolabels of $f_E(\widetilde{x}_i^a)$ and $f_E(\widetilde{x}_i^d)$ are exchanged. The pseudolabel y^d is assigned to $f_E(\widetilde{x}_i^a)$, and the pseudolabel y^a is assigned to $f_E(\widetilde{x}_i^d)$. We train encoder $f_E(\bullet)$ to minimize $\mathcal{L}_G$, which is formulated in the following. The learning objective of the encoder $f_E(\bullet)$ is to make $h_D(f_E(\widetilde{x}_i^d)) = h_D(\widetilde{z}_i^a)$ and $h_D(f_E(\widetilde{x}_i^a)) = h_D(\widetilde{z}_i^d)$ simultaneously:

$$\begin{aligned}\mathcal{L}_G &= \frac{1}{2N}\sum_{i=1}^{N}\Big[\mathrm{BCE}\big(h_D\big(f_E\big(\widetilde{x}_i^d\big)\big), y^a\big) \\ &\qquad + \mathrm{BCE}\big(h_D\big(f_E\big(\widetilde{x}_i^a\big)\big), y^d\big)\Big] \\ &= -\frac{1}{2N}\sum_{i=1}^{N}\Big[\log\big(1 - h_D\big(f_E\big(\widetilde{x}_i^d\big)\big) + \log h_D\big(f_E\big(\widetilde{x}_i^a\big)\big)\Big].\end{aligned} \tag{8}$$

Based on the value function $\widetilde{V}(f_E, h_D)$ in (9), the encoder $f_E(\bullet)$ and discriminator $h_D(\bullet)$ continuously game each other during pretraining. The $P(\widetilde{x}_i^a)$ is similar to $P(\widetilde{z}_i^d)$, and $P(\widetilde{x}_i^d)$ is similar to $P(\widetilde{z}_i^a)$. Thus, the encoder $f_E(\bullet)$ increasingly focuses on the invariant representations between the views $\widetilde{x}_i^a$ and $\widetilde{x}_i^d$, rather than perturbation representations caused by ECG augmentations, which can increase the antiperturbation ability

$$\begin{aligned}\min_{f_E}\max_{h_D}\widetilde{V}(f_E, h_D) &= \mathbb{E}_{\widetilde{z}_i^d\sim P(\widetilde{z}_i^d)}\big[\log h_D\big(\widetilde{z}_i^d\big)\big] \\ &\quad + \mathbb{E}_{\widetilde{x}_i^a\sim P(\widetilde{x}_i^a)}\big[\log\big(1 - h_D\big(f_E\big(\widetilde{x}_i^a\big)\big)\big)\big] \\ &\quad + \mathbb{E}_{\widetilde{x}_i^d\sim P(\widetilde{x}_i^d)}\big[\log h_D\big(f_E\big(\widetilde{x}_i^d\big)\big)\big] \\ &\quad + \mathbb{E}_{\widetilde{z}_i^a\sim P(\widetilde{z}_i^a)}\big[\log\big(1 - h_D\big(\widetilde{z}_i^a\big)\big)\big].\end{aligned} \tag{9}$$

D. Spatiotemporal Contrastive Module

The spatiotemporal contrastive module is used to learn spatiotemporal and semantic representations of categories via spatiotemporal prediction and patient discrimination in unsupervised training. As shown in Fig. 2, this module has two branches, whose inputs are views $\widetilde{x}_i^a$ and $\widetilde{x}_i^d$ after shuffling. The representation $z_i^\sigma = f_E(\widetilde{x}_i^\sigma)$ is expressed as $z_i^\sigma = \{z_{i,1}^\sigma, \ldots, z_{i,\tau}^\sigma, \ldots, z_{i,K}^\sigma\}$, where $z_{i,\le\tau}^\sigma$ is past segment of z_i^σ, $z_{i,>\tau}^\sigma$ is future segment of z_i^σ, and τ is length of past segment. By performing recombination $RC(\bullet)$, the future and past segments of z_i^a and z_i^d are recombined to generate $\widetilde{z}_i^a$ and $\widetilde{z}_i^d$, respectively, which are defined in the following:

$$\widetilde{z}_i^a = RC\big(z_i^a, z_i^d\big) = \big\{z_{i,1}^a, \ldots, z_{i,\tau}^a, z_{i,\tau+1}^d, \ldots, z_{i,K}^d\big\} \tag{10}$$

$$\widetilde{z}_i^d = RC\big(z_i^d, z_i^a\big) = \big\{z_{i,1}^d, \ldots, z_{i,\tau}^d, z_{i,\tau+1}^a, \ldots, z_{i,K}^a\big\}. \tag{11}$$

In spatiotemporal prediction, because transformer [55] has an excellent performance in regression tasks, it is adopted in this module to capture spatiotemporal representations. As shown in Fig. 4(b), the architecture of our used transformer $f_T(\bullet)$ is composed of an embedding block, four multihead attention blocks, and four feed-forward blocks. To make the gradient more stable, the residual connection used in transformer $f_T(\bullet)$ is the prenorm residual connection proposed in the literature [56]. $\widetilde{z}_{i,\le\tau}^\sigma$ is converted via embedding block and concatenated with the generated token st. Afterward, $\widetilde{z}_{i,\le\tau}^\sigma$ is input into the multihead attention blocks and feed-forward blocks for extracting semantic vector $sv_i^\sigma = f_T(\widetilde{z}_{i,\le\tau}^\sigma) \in \mathbb{R}^M$, where M is the hidden dimension of transformer $f_T(\bullet)$. The

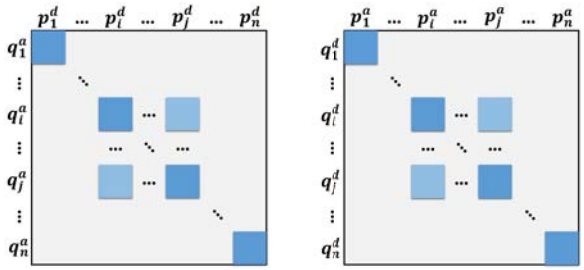


Fig. 5. Similarity matrix between prediction and projection.

semantic vector sv_i^σ is used to predict each latent representation $\widetilde{z}_{i,\tau+k}^\sigma$ of $\widetilde{z}_{i,>\tau}^\sigma$, $1 \le k \le K - \tau$. The sequence $\widetilde{x}_{i,\tau+k}^\sigma$ of view $\widetilde{x}_i^\sigma$ is mapped to $\widetilde{z}_{i,\tau+k}^\sigma$ via the encoder. Inspired by the study of [19], the log-bilinear model $f_k(\bullet)$ is used to preserve the mutual information between the sequence $\widetilde{x}_{i,\tau+k}^\sigma$ and semantic vector sv_i^σ. The $W_k(\bullet)$ is a linear function, which is used for mapping sv_i^σ to $\widetilde{z}_{i,\tau+k}^\sigma$. To maximize the mutual information of two branches, we minimize the $\mathcal{L}_T$, which is defined in the following.

$$f_k\big(\widetilde{x}_{i,\tau+k}^\sigma, sv_i^\sigma\big) = \exp\Big(\big(\widetilde{z}_{i,\tau+k}^\sigma\big)^T W_k\big(sv_i^\sigma\big)\Big) \tag{12}$$

$$\mathcal{L}_a = -\frac{1}{K-\tau}\sum_{k=1}^{K-\tau}\log\frac{f_k\big(\widetilde{x}_{i,\tau+k}^a, sv_i^a\big)}{\sum_{h=1}^{N} f_k\big(\widetilde{x}_{h,\tau+k}^a, sv_i^a\big)} \tag{13}$$

$$\mathcal{L}_d = -\frac{1}{K-\tau}\sum_{k=1}^{K-\tau}\log\frac{f_k\big(\widetilde{x}_{i,\tau+k}^d, sv_i^d\big)}{\sum_{h=1}^{N} f_k\big(\widetilde{x}_{h,\tau+k}^d, sv_i^d\big)} \tag{14}$$

$$\mathcal{L}_T = \frac{1}{2N}\sum_{i=1}^{N}(\mathcal{L}_a + \mathcal{L}_d). \tag{15}$$

In patient discrimination, the projector $h_P(\bullet)$ used in this module maps semantic vector sv_i^σ to projection space for obtaining projection $p_i^\sigma = h_P(sv_i^\sigma) \in \mathbb{R}^H$, where H is the hidden dimension of the projector $h_P(\bullet)$. To learn semantic representations of categories effectively, we add the predictor $h_Q(\bullet)$ and the stop-gradient $sg(\bullet)$, as in the studies of [30] and [15], to avoid model collapse without using negative pairs. The goal of the predictor $h_Q(\bullet)$ is to promote the similarity of two branches representations via forecasting the projection of the other branch, its prediction is defined as $q_i^\sigma = h_Q(p_i^\sigma) \in \mathbb{R}^H$. The comparisons between q_i^σ and p_i^σ are illustrated in the similarity matrix as Fig. 5. Kiyasseh et al. [16] set forth that the clinical object should be at the patient-level, rather than the instance-level, so we define positive pairs at the patient-level. Based on the patient ID, we use a set $E = \{e_1, \ldots, e_i, \ldots, e_n\}$ to represent the relationship between the patient and the instance, where e_i refers to other instances of the same patient as instance i. For example, $e_i = \{j\}$, $1 < j < N$ means that instance i and j come from the same patient, the diagonal elements (q_i^a, p_i^d), (q_i^d, p_i^a) and nondiagonal elements (q_i^a, p_j^d), (q_i^d, p_j^a) of similarity matrix are defined as positive pairs for instance i. The similarity of these positive pairs is quantified as (16). It is worth mentioning that the stop gradient $sg(\bullet)$ of projection plays a crucial role in preventing collapsing caused by abandoning negative pairs. This operation is applied to p_i^σ for the $\mathcal{L}_{\mathrm{diag}}$ and $\mathcal{L}_{\mathrm{nondiag}}$

calculation. We maximize the similarity of all positive pairs by minimizing the $\mathcal{L}_C$, which is defined in the following

Algorithm 1 ASTCL

Require: dataset $X = \{x_1, \cdots x_i, \cdots x_n\}$, batchsize B, past segment length τ, set $E = \{e_1, \cdots e_i, \cdots e_n\}$, set $U = \{u_1, \cdots u_i, \cdots u_n\}$

for sampled minibatch $x_{i\{i=1\}}^{B} \in X$ **do**
  for x_i in $x_{i\{i=1\}}^{B}$ **do**
    $\widetilde{x}_i^a, \widetilde{x}_i^d \leftarrow A_a(x_i), A_d(x_i)$
    $\widetilde{z}_i^a, \widetilde{z}_i^d \leftarrow f_{\widehat{E}}(\widetilde{x}_i^a), f_{\widehat{E}}(\widetilde{x}_i^d)$
    Get $h_D(\widetilde{z}_i^a), h_D(\widetilde{z}_i^d)$
    $z_i^a, z_i^d \leftarrow f_E(\widetilde{x}_i^a), f_E(\widetilde{x}_i^d)$
    Get $h_D(z_i^a), h_D(z_i^d)$
    Recombine $RC(z_i^a, z_i^d), RC(z_i^d, z_i^a) \rightarrow \widetilde{z}_i^a, \widetilde{z}_i^d$
    $sv_i^a, sv_i^d \leftarrow f_T(\widetilde{z}_{i,\le\tau}^a), f_T(\widetilde{z}_{i,\le\tau}^d)$
    $p_i^a, p_i^d \leftarrow h_P(sv_i^a), h_P(sv_i^d)$
    $q_i^a, q_i^d \leftarrow h_Q(p_i^a), h_Q(p_i^d)$
    Stop-gradient $p_i^a, p_i^d \rightarrow sg(p_i^a), sg(p_i^d)$
  end for
  Calculate $\mathcal{L}_D$, $\mathcal{L}_G$, $\mathcal{L}_T$, $\mathcal{L}_C$ and $\mathcal{L}_F$ by Eq.7, Eq.8, Eq.15, Eq.19 and Eq.20
  Update $h_D(\bullet)$ to minimize $\mathcal{L}_D$
  Update $f_E(\bullet)$, $f_T(\bullet)$, $h_P(\bullet)$ and $h_Q(\bullet)$ to minimize $\mathcal{L}_F$
  Share weight $f_E(\bullet) \rightarrow f_{\widehat{E}}(\bullet)$
end for

Ensure: encoder $f_E(\bullet)$.

$$\mathrm{Sim}\left(q_i^a, p_i^d\right) = -\frac{q_i^a}{\|q_i^a\|_2} \cdot \frac{p_i^d}{\|p_i^d\|_2} \tag{16}$$

$$\mathcal{L}_{\text{diag}} = \frac{1}{2}\left[\mathrm{Sim}\left(q_i^a, sg\left(p_i^d\right)\right) + \mathrm{Sim}\left(q_i^d, sg\left(p_i^a\right)\right)\right] \tag{17}$$

$$\mathcal{L}_{\text{nondiag}} = \frac{1}{u_i}\sum_{j\in e_i}\left[\mathrm{Sim}\left(q_i^a, sg\left(p_j^d\right)\right) + \mathrm{Sim}\left(q_i^d, sg\left(p_j^a\right)\right)\right] \tag{18}$$

$$\mathcal{L}_C = \frac{1}{2N}\sum_{i=1}^{N}\left(\mathcal{L}_{\text{diag}} + \mathcal{L}_{\text{nondiag}}\right) \tag{19}$$

where $\mathrm{Sim}(\bullet)$ is the cosine similarity, and u_i is the number of elements in e_i.

Trained by the self-supervised tasks, such as an adversarial game, spatiotemporal prediction, and patient discrimination, the encoder is updated via using $\mathcal{L}_G$, $\mathcal{L}_T$ and $\mathcal{L}_C$ simultaneously. The overall loss function of ASTCL is shown as follows:

$$\mathcal{L}_F = w_1\mathcal{L}_G + w_2\mathcal{L}_T + w_3\mathcal{L}_C \tag{20}$$

where w_1, w_2, and w_3 are the weight of $\mathcal{L}_G$, $\mathcal{L}_T$, and $\mathcal{L}_C$ in $\mathcal{L}_F$, respectively. The main pseudocode of ASTCL is in Algorithm 1.

E. Complexity Analysis

The complexity of the main components of ASTCL is analyzed using the complexity theory, including ECG augmentations, encoder, discriminator, transformer, projector, and predictor. In the pretraining stage, the time complexity of the core steps of these components is $\mathcal{O}(T \times C)$, $\mathcal{O}(ks \times K \times D^2)$, $\mathcal{O}(D^2)$, $\mathcal{O}(K \times M^2 + K^2 \times M)$ and $\mathcal{O}(H^2)$, and their space complexity are $\mathcal{S}(T \times C)$, $\mathcal{S}(ks \times D^2 + K \times D)$, $\mathcal{S}(D^2+1)$, $\mathcal{S}(K^2 + K)$, and $\mathcal{S}(H^2 + H)$, where ks is the kernel size of the encoder, and the complexity of the projector and predictor can be regarded as equal. However, in the abovementioned components of ASTCL, only the encoder works in the fine-tuning stage and the testing stage. Thus, the time complexity is $\mathcal{O}(ks \times K \times D^2)$, and the space complexity is $\mathcal{S}(ks \times D^2 + K \times D)$. The detailed complexity analysis of ASTCL is presented in Supplementary Material.

Because the encoder of ASTCL can be replaced when the pretrained encoder is used for clinical diagnosis, only the architecture of the encoder and the size of the ECG signal will affect the diagnosis efficiency, while ASTCL will not increase time cost and memory cost in the diagnosis process.

IV. Experimental Setup

The datasets and baseline methods used in the article are specifically introduced in this section. In addition, the evaluation of downstream tasks and experimental details are also introduced.

A. Datasets

ECG classification studies are mainly divided into multiclass and multilabel tasks [57]. As shown in Table I, we use two multiclass benchmark ECG datasets, two multilabel benchmark ECG datasets, and the clinical ECG dataset to verify the performance of the ASTCL. The description of these datasets is as follows (see supplementary material for details).

Chapman [58] contains 12-lead ECG data of 10 646 patients, which is a multiclass dataset. According to the suggestions of the literature [58], we use four categories (i.e., AF, GSVT, SB, and SR) instead of the original 11 categories.

PTB-XL [59] includes 21 837 12-lead ECG records from 18 885 patients, which is a multilabel dataset with five categories. To enrich the multiclass classification, we select single-label data in five categories and reduce normal records.

CODE [60] consists of 2 322 513 12-lead ECG records from 1 676 384 patients. This dataset includes seven categories for multilabel classification. We select the experimental data from exams_part0 to exams_part3 in CODE.

CPSC2018 [61] includes 6877 patients with 12-lead records. This dataset is a multilabel arrhythmias dataset with nine categories. To standardize the length of the records, we cut this dataset according to 10 s.

Clinical myocardial infarction (CMI) dataset is a multilabel clinical dataset, collected under the Cooperative Innovation Center for Internet Healthcare, Zhengzhou University, Zhengzhou, China. This dataset includes 10 336 12-lead records from 7317 patients, and it consists of five categories.

B. Baselines

The state-of-the-art data augmentation methods and contrastive learning frameworks are chosen as the baseline

TABLE I
DESCRIPTION OF DATASET IN EXPERIMENTS

Dataset	Train	Valid	Test	Category	Task
Chapman	6,376	2,126	2,126	4	Multi-class
PTB-XL	8,006	2,669	2,669	5	Multi-class
CODE	7,535	2,512	2,512	7	Multi-label
CPSC2018	5,618	1,873	1,873	9	Multi-label
CMI	6,202	2,067	2,067	5	Multi-label

TABLE II
AUC OF LINEAR EVALUATION

Method	Chapman	PTB-XL	CODE	CPSC2018
Rand-Init	79.75±2.33	71.45±1.59	78.27±1.70	72.10±0.55
CPC [19]	83.71±1.71	70.28±0.93	81.56±1.27	74.58±0.36
SimCLR [21]	87.68±0.42	67.50±0.59	81.96±0.83	73.21±1.01
BOYL [30]	90.61±0.40	76.12±1.05	88.28±0.17	77.38±0.73
SimSiam [15]	89.58±1.39	71.35±1.85	87.17±1.76	75.49±2.59
TS-TCC [13]	92.42±0.64	**81.76±0.49**	**91.40±0.14**	**83.48±0.37**
CLOCS [16]	**92.55±0.35**	81.74±0.53	90.25±0.58	82.03±0.21
ASTCL (ours)	**93.05±0.46**	**82.03±0.65**	**92.18±0.27**	**84.24±0.46**

methods. The following is a detailed introduction from two aspects: augmentation and framework.

In data augmentation, CLOCS [16] utilizes adding Gaussian noise, the flipping x-axis or y-axis, and mask wave for ECG transformation. TS-TCC [13] applies jitter-and-scale in weak augmentation and permutation-and-jitter in strong augmentation to augment EEG, respectively. The above-mentioned augmentation methods are considered baseline methods, which are implemented to compare with the proposed ECG augmentations.

Regarding the framework, we reproduced the state-of-the-art contrastive learning frameworks to compare with the proposed ASTCL, including contrastive predictive coding (CPC) [19], SimCLR [21], Bootstrap Your Own Latent (BYOL) [30], and SimSiam [15] of computer vision or natural language processing field, and TS-TCC [13] and CLOCS [16] of PTS field. In addition, we also take randomly initialized training (Rand-Init), supervised pretraining (presupervised), and supervised training as the baseline in the experiments.

C. Verification Scenarios

We first employ contrastive learning frameworks to perform unsupervised pretraining on the source domain dataset. The pretrained encoder is used to implement downstream tasks, including multiclass classification and multilabel classification. The linear classifier acts as the downstream classifier, which has a single-linear layer. To comprehensively examine the effect of our proposed ECG augmentations and ASTCL, we set up a series of scenarios to verify the augmented effect, pretraining effect, antiperturbation ability, category learning ability, transferability, semisupervised ability, and main components effect.

Aiming to test whether the proposed augmentation methods are useful, ECG augmentations and other baseline methods are applied to contrastive learning frameworks to quantify their effect, respectively. Meanwhile, the pretrained parameters are used as the initialization parameters, which are fixed in linear classifier training. The performance of the linear classifier is used to evaluate the pretraining ability. To verify the antiperturbation ability, the pretrained encoder is trained on the noise enhancement dataset to test the robustness of the encoder to noise. In category learning, the pretraining parameters are fine-tuned by labeled data of the source domain dataset. The performance of ASTCL in each category is shown by the category learning ability. For transferability evaluation, we fine-tune pretrained parameters when training labeled data of the target domain dataset, then compare results of ASTCL and baseline frameworks on the target domain dataset. Moreover, the pretrained encoder is fine-tuned by the different percentages labeled data of the source domain dataset to test semisupervised ability. To evaluate the effect of the main components of ASTCL, we conducted an ablation study to verify the ECG augmentations, adversarial game, and only using patient-level positive pairs, respectively. Besides, we also test the computational cost of our method in the pretraining stage (see supplementary material).

D. Details

In data preprocessing, the amplitudes of these five datasets are normalized between 0 and 1. We resample the data to 250 Hz and extract 10 s of data as experimental data (due to the mechanism, CLOCS uses 500-Hz data). Each training set, validation set, and testing set are randomly divided from each dataset according to the proportion of 60%, 20%, and 20%, respectively. We repeat these experiments five times with five different seeds, respectively, and analyze the mean and standard deviation of each experiment. Whether in the pretraining stage, fine-tuning stage, or testing stage, the batch size and epochs are set to 128 and 100, respectively. Meanwhile, the dimension of the encoder, transformer, and projector is set to $D = 128$, $M = 100$, and $H = 32$, respectively. We employ the Adam optimizer to update the parameters and set the learning rate to $\eta =$ 3e-4 and weight decay to $\beta 1 = 0.9$ and $\beta 2 = 0.99$. The SNR used in ECG augmentations is defined as $\mu = 5$ dB. The past segment length τ is set to $0.6K$, which is the same as in the study of [13]. About the weight of $\mathcal{L}_F$, we set $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$, which achieves the best performance in the ablation study. In SimCLR, CLOCS, and TS-TCC, the temperature parameter of the NT-Xent loss function is set as 0.2. For BYOL, we set the decay rate to 0.90. To ensure fairness, the input, data augmentations, encoder, optimizer, and classifier used by each framework are consistent. Finally, we built ASTCL using PyTorch 1.4 on Ubuntu 16.4 and trained it on NVIDIA GeForce RTX 2080Ti GPU. The implementation details are presented in supplementary material.

V. EXPERIMENTAL RESULT

This section shows the results of augmentation evaluation, linear evaluation, noise evaluation, category evaluation, transferability evaluation, semisupervised evaluation, and ablation study. We select the macro-F1 score and AUC as the metrics to evaluate the results. In Section VI, the best experiment results are marked in **black** and the second-best in **red**.

TABLE III
F1 SCORE OF NOISE STRESS EVALUATION

Dataset	Chapman			PTB-XL		
SNR	2dB	5dB	10dB	2dB	5dB	10dB
Supervised	80.83±2.60	81.44±0.59	82.09±1.96	51.50±0.76	52.03±0.94	52.27±1.93
CPC [19]	83.01±1.36	84.36±1.91	84.97±1.29	51.94±0.78	52.66±1.26	54.00±0.84
SimCLR [21]	85.73±1.03	86.12±1.44	81.72±3.69	54.20±0.67	54.76±0.93	55.81±0.87
BOYL [30]	85.92±0.74	88.09±0.41	87.17±0.62	**55.60±0.52**	55.73±0.91	55.97±1.10
SimSiam [15]	86.46±1.98	86.59±1.72	86.70±1.83	54.84±0.64	**56.24±0.79**	**56.32±0.88**
TS-TCC [13]	**86.85±1.05**	**87.76±0.96**	**88.27±0.94**	55.29±1.47	56.23±1.51	56.27±1.91
CLOCS [16]	86.49±1.13	87.41±0.82	87.63±0.99	53.56±0.83	55.31±0.88	55.44±0.61
ASTCL (ours)	**87.77±0.75**	**88.78±0.71**	**89.25±1.02**	**56.58±1.04**	**56.99±0.56**	**57.36±0.92**
Dataset	CODE			CPSC2018		
SNR	2dB	5dB	10dB	2dB	5dB	10dB
Supervised	75.32±1.37	76.44±1.14	76.53±1.52	48.40±1.51	52.83±1.11	53.78±2.68
CPC [19]	75.86±1.17	76.47±1.27	76.67±1.46	49.15±2.45	50.27±1.57	52.64±0.87
SimCLR [21]	79.48±0.91	81.71±1.48	82.14±0.89	56.93±0.96	57.74±1.06	58.25±1.61
BOYL [30]	81.37±0.84	82.58±0.81	82.42±1.28	55.18±1.28	57.80±1.08	59.45±1.19
SimSiam [15]	**81.80±0.74**	**83.76±0.76**	**83.74±1.21**	55.12±0.95	56.74±1.68	58.42±1.16
TS-TCC [13]	79.77±0.66	81.47±0.75	81.89±0.60	**58.63±1.19**	**59.81±1.02**	**60.48±0.91**
CLOCS [16]	79.46±0.38	80.29±0.83	80.65±1.05	54.76±1.35	57.15±0.56	58.39±0.86
ASTCL (ours)	**81.67±0.70**	**83.87±0.51**	**84.13±0.58**	**61.56±1.17**	**62.62±0.83**	**63.34±1.26**

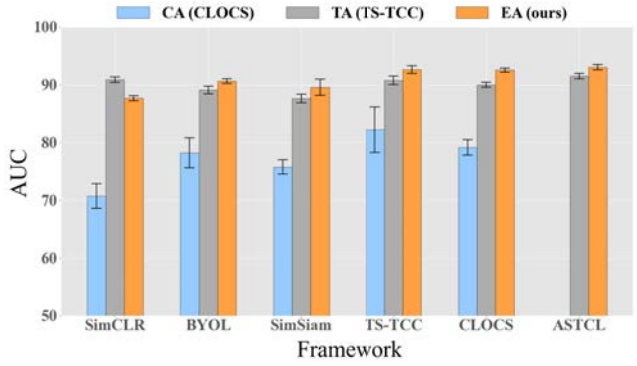


Fig. 6. Effects of different data augmentations applied to contrastive learning frameworks on Chapman dataset.

A. Augmentation Evaluation

The proposed ECG augmentations (EAs), data augmentations of CLOCS (CA), and data augmentations of TS-TCC (TA) are used in a series of contrastive learning frameworks, respectively (except CPC, because it does not need data augmentations). The two augmented views of ASTCL must be different. Thus, ASTCL does not participate in the evaluation of CA. These frameworks are pretrained on the Chapman dataset and performed linear evaluation using 50% labeled data.

Fig. 6 shows the AUCs of these augmentation methods in the linear evaluation. It is not difficult to find that EA is better than CA and TA in the pretraining of contrastive learning frameworks. For example, the results of CLOCS on three augmentation methods are $\text{AUC}_{\text{EA}} = 92.55\%$, $\text{AUC}_{\text{TA}} = 90.01\%$, and $\text{AUC}_{\text{CA}} = 79.17\%$. Overall, the average AUCs of EA, TA, and CA are 91.02%, 89.98%, and 77.23%, respectively. The proposed EA performs better than TA and CA, which verifies that the two augmented views generated by ECG noise enhancement and ECG noise denoising are more conducive to representation learning of contrastive learning frameworks.

B. Linear Evaluation

To test whether the proposed ASTCL is effective, the parameters of the pretrained encoder are used as the initialization parameters of the encoder in the downstream classification task. In Rand-Init, the initialization parameters are random parameters. During training, the parameters of the encoder are fixed, and the linear classifier is updated with the half-labeled data to perform the classification task, in order to compare the performance of our ASTCL with the baseline frameworks.

Table II shows the AUCs of the ASTCL and baseline frameworks on four ECG datasets. The AUCs of TS-TCC and CLOCS are better than other baselines, which verify the importance of spatiotemporal and semantic representations. The performance of BYOL and SimSiam is better than SimCLR, which also shows that discarding negative pairs play a useful role in representation learning. Compared with the above-mentioned state-of-the-art frameworks, ASTCL performs better than them on all datasets. Especially on the CODE dataset, the AUC of ASTCL reaches 92.18%, which improves by 0.78% over the second-best framework. To further demonstrate the influence of pretrained parameters on the encoder, we employ the t-SNE [62] to visualize learned representations in 2-D space. Fig. 7 shows visualization results of ASTCL and baseline frameworks on the Chapman dataset. Obviously, the results of ASTCL are more instrumental in distinguishing each category, and ASTCL provides a better parameter space for the linear classifier. These experiments mean that the combination of adversarial game and spatiotemporal comparison of patient-level positive pairs is effective.

C. Noise Evaluation

To verify the antiperturbation ability of the encoder pretrained by ASTCL, we designed a series of experiments to test the robustness of the encoder to noise. According to the general noise stress evaluation [63], we add BD, MA, and PF noises to the original signal using the SNR of 2, 5, and 10 dB.

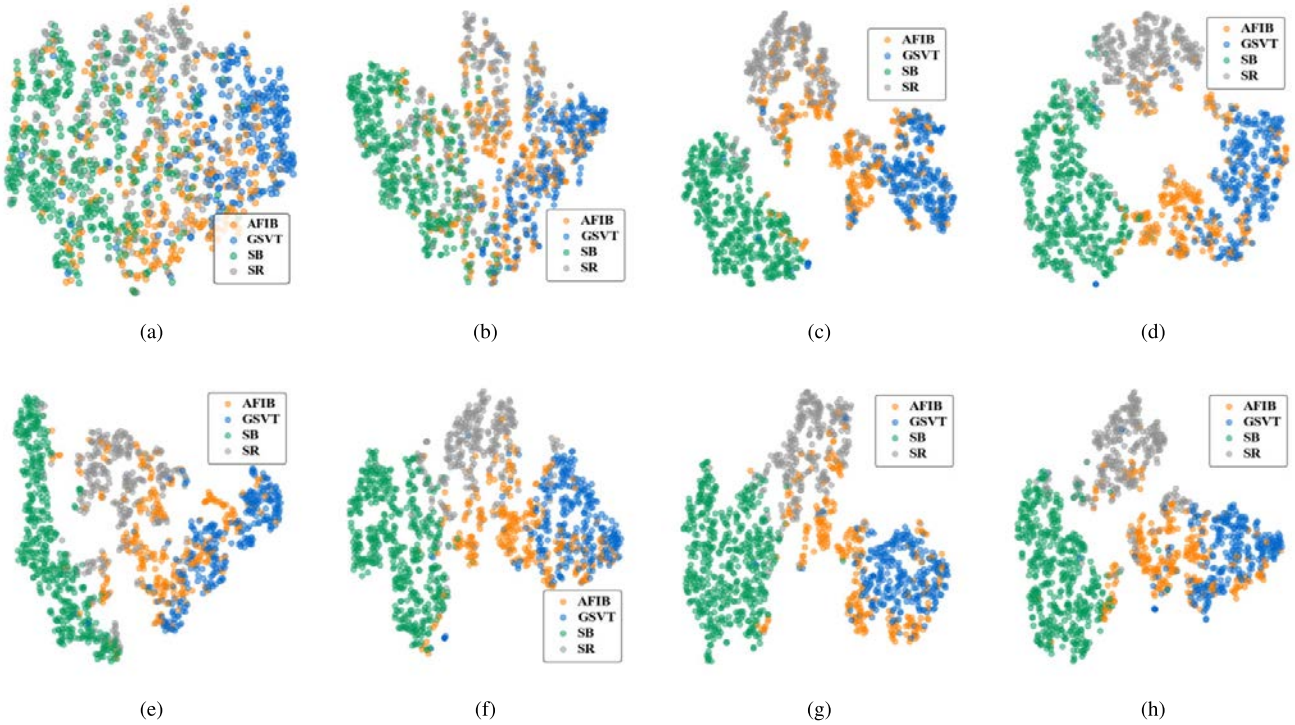


Fig. 7. t-SNE visualization of the learned representations on Chapman dataset. Different colors represent different categories. (a) Rand Init. (b) CPC [19]. (c) SimCLR [21]. (d) BYOL [30]. (e) SimSiam [15]. (f) TS-TCC [13]. (g) CLOCS [16]. (h) ASTCL (ours).

After the pretraining of all contrastive learning frameworks, the pretrained encoder is fine-tuned by 50% labeled noised ECG data to perform the classification task.

As shown in Table III, when BD, MA, and PF noises are integrated into the original signal, the performance of the model in supervised training is degraded significantly on all datasets, which indicates that ECG noises seriously disturb the model to learn representations. Despite noise perturbation, ASTCL still outperforms other baseline frameworks. In the twelve evaluation scenarios on four datasets, the F1 score of ASTCL is almost the top 1. Especially on the CPSC2018 dataset, the F1 scores of ASTCL are 2.93%, 2.81%, and 2.86% higher than the top 2. Compared with supervised training, when the SNR is 2 dB, the F1 score of ASTCL increases by 13.16%. The experimental results show that the encoder pretrained by ASTCL has been strengthened in antiperturbation ability. Even if there is more serious noise, ASTCL can still maintain its robustness to noise.

D. Category Evaluation

For the ECG classification task, it is significant to effectively learn the category representations and improve the classification precision of each category. We further examine the category representation learning ability of ASTCL by a series of category evaluation experiments. We first pretrain these contrastive learning frameworks on five datasets, respectively, and then use 50% labeled data of pretraining datasets for fine-tuning. In supervised training, we use the same labeled data to train for comparison.

The experimental F1 scores of each category are shown in Table IV. We show that the proposed ASTCL is superior to other frameworks in category representation learning. On the Chapman dataset, the F1 scores of ASTCL in four categories achieve 82.99%, 87.43%, 96.96%, and 90.80%, which are higher than the second-best framework. On the other datasets, the experimental results of ASTCL are stable in the top 1 or top 2. To qualitatively show the learned category representations, the confusion matrices of ASTCL fine-tuning results and supervised results are illustrated in Fig. 8. Since most datasets have more than four categories and duplicate categories, we only retain categories different from other datasets in the confusion matrix. The number of confusing matrix elements is n_s/n_t, where n_s is the predicted number of the category samples, and n_t is the total number of the category samples. As shown in Fig. 8, ASTCL effectively improves the classification performance of each category. Especially in the MI category of the PTB-XL dataset, ASTCL can achieve 20.9% improvement compared with supervised training. These experimental results prove that compared with other frameworks, ASTCL can better learn category representations.

E. Transferability Evaluation

In this section, we test the transferability of contrastive learning frameworks through transfer learning experiments. We utilize ASTCL and baseline frameworks to pretrain the source domain dataset. The pretrained parameters are the initialization parameters of the encoder. In presupervised, we take supervised pretraining on the source domain dataset. During fine-tuning, the partially labeled data of the target domain dataset is used to fine-tune the parameters of the pretrained encoder and update the linear classifier.

Table V shows the F1 scores of ASTCL and baseline frameworks in transfer learning experiments. The 50% labeled data of the target domain dataset is used for fine-tuning. Compared with all baseline frameworks, ASTCL is the best in transferability. It can be seen from the F1 score of ASTCL

TABLE IV
F1 SCORE OF CATEGORY EVALUATION ON ALL EXPERIMENTAL DATASETS

(a)

Category	Supervised	CPC [19]	SimCLR [21]	BOYL [30]	SimSiam [15]	TS-TCC [13]	CLOCS [16]	ASTCL (ours)
AFIB	73.83±1.42	76.15±1.94	74.34±2.19	80.06±1.37	79.05±2.89	**81.97±2.05**	80.22±1.74	**82.99±2.45**
GSVT	79.87±2.40	85.72±1.33	80.87±2.58	85.99±1.48	85.88±2.29	**86.12±1.47**	86.00±0.43	**87.43±1.01**
SB	95.75±0.13	95.80±0.26	95.75±0.05	96.19±0.35	96.35±0.53	**96.91±0.22**	96.52±0.28	**96.96±0.33**
SR	83.11±2.51	85.50±1.12	82.87±3.30	88.29±1.30	88.31±1.87	**90.29±0.75**	88.82±1.05	**90.80±0.94**
Total	83.14±1.36	85.79±1.02	83.46±2.14	87.63±0.99	87.40±1.41	**88.82±0.95**	87.89±0.84	**89.54±1.11**

(b)

Category	Supervised	CPC [19]	SimCLR [21]	BOYL [30]	SimSiam [15]	TS-TCC [13]	CLOCS [16]	ASTCL (ours)
CD	66.19±2.45	66.80±2.36	69.80±1.72	**72.69±1.37**	71.89±1.09	71.58±0.84	69.37±1.22	**72.61±0.61**
HYP	0.64±0.79	4.22±1.35	3.82±1.75	3.39±1.71	5.63±1.33	**6.50±1.34**	4.40±1.67	**6.20±1.20**
MI	51.99±5.14	55.97±1.00	58.99±2.57	**61.83±0.96**	60.29±0.84	58.04±1.39	57.33±2.49	**61.39±1.48**
NORM	81.06±0.82	81.53±0.48	81.60±0.84	82.50±0.33	82.56±0.21	82.26±0.66	**82.79±0.29**	**83.06±0.54**
STTC	61.36±0.91	62.27±1.21	62.29±0.93	**63.79±0.92**	63.57±1.42	63.61±1.25	**64.95±0.61**	63.76±1.05
Total	52.25±1.36	54.16±0.39	55.51±0.67	**56.84±0.57**	56.79±0.37	56.42±0.87	55.77±0.88	**57.40±0.88**

(c)

Category	Supervised	CPC [19]	SimCLR [21]	BOYL [30]	SimSiam [15]	TS-TCC [13]	CLOCS [16]	ASTCL (ours)
1dAVb	48.96±7.10	44.35±4.88	59.14±6.43	64.57±3.91	**68.34±2.26**	61.29±1.67	57.44±4.05	**68.33±2.34**
RBBB	91.79±1.11	93.22±0.45	93.43±0.20	93.62±0.16	**93.49±0.36**	92.92±0.69	92.52±0.39	**93.75±0.31**
LBBB	88.01±1.33	88.45±0.65	88.74±0.63	89.23±0.43	**90.55±0.36**	89.35±0.68	89.24±0.59	**90.38±1.32**
SB	81.35±1.74	80.89±1.44	81.91±1.25	83.65±1.43	82.79±1.36	**84.19±1.77**	82.55±0.53	**85.41±1.33**
ST	88.13±1.43	88.70±1.02	89.98±1.24	89.99±0.53	**90.80±0.90**	89.77±0.41	88.27±0.72	**90.76±0.72**
AF	51.23±2.82	51.26±5.04	65.45±3.79	65.89±1.69	**70.36±2.99**	63.19±2.11	57.92±2.69	**72.27±1.55**
Norm	92.32±0.64	92.42±0.62	94.12±0.36	**94.34±0.22**	**94.63±0.31**	93.76±0.40	93.03±0.49	94.32±0.15
Total	77.39±1.87	77.04±1.50	81.82±1.09	83.04±0.65	**84.42±0.69**	82.07±0.55	80.14±1.00	**85.03±0.42**

(d)

Category	Supervised	CPC [19]	SimCLR [21]	BOYL [30]	SimSiam [15]	TS-TCC [13]	CLOCS [16]	ASTCL (ours)
PVC	51.52±3.62	54.08±2.61	53.85±2.89	55.63±1.35	57.64±1.07	**58.73±1.81**	57.48±1.15	**59.62±0.96**
AF	67.70±2.14	62.72±5.50	74.81±2.55	73.16±1.70	70.37±3.46	**75.81±2.09**	70.56±1.42	**76.80±1.31**
LBBB	88.42±1.79	88.89±0.61	89.61±0.68	**90.41±1.11**	89.15±1.13	90.32±1.01	90.19±1.99	**90.69±0.44**
STE	37.40±4.39	27.37±5.34	44.68±3.29	31.71±7.63	30.09±6.73	**40.34±2.34**	38.13±5.57	**43.24±3.62**
IAVB	53.36±6.13	40.27±4.13	**68.59±5.99**	65.23±1.30	65.16±4.70	64.19±2.73	60.94±3.42	**71.71±2.98**
PAC	20.83±2.83	14.36±2.52	13.08±1.80	19.36±1.64	20.55±4.81	**22.79±2.41**	20.15±2.13	**22.29±2.11**
NSR	51.79±3.14	54.46±2.15	49.05±4.83	57.34±1.92	57.99±2.83	**60.73±3.94**	57.07±3.69	**60.96±1.19**
STD	48.76±4.50	51.58±3.56	55.59±1.82	55.08±2.17	53.63±4.44	**58.92±1.35**	56.04±3.30	**58.76±2.31**
RBBB	86.63±0.62	87.25±1.62	87.90±0.66	87.86±0.97	88.13±0.67	**88.76±0.91**	87.51±0.53	**89.11±0.64**
Total	56.27±1.71	53.44±1.29	59.68±1.80	59.53±1.24	84.42±0.69	**62.29±1.07**	59.79±1.12	**63.69±0.69**

(e)

Category	Supervised	CPC [19]	SimCLR [21]	BOYL [30]	SimSiam [15]	TS-TCC [13]	CLOCS [16]	ASTCL (ours)
AMI	93.14±1.37	94.24±0.86	95.20±0.65	95.03±0.62	95.21±0.66	**95.33±0.88**	94.64±0.51	**95.75±0.31**
IMI	86.65±0.99	86.84±0.51	86.86±1.34	88.32±0.86	**88.46±0.91**	**88.23±0.59**	88.22±0.69	88.19±1.00
LMI	35.13±3.95	19.14±2.80	28.90±8.13	37.86±4.99	29.65±3.82	**41.01±4.72**	26.82±5.02	**40.61±4.03**
PMI	30.27±1.50	21.84±7.67	17.39±12.42	30.81±4.25	23.83±6.54	**31.23±3.36**	23.58±4.62	**32.30±4.01**
Norm	92.98±2.92	95.21±0.94	95.42±0.79	95.76±1.11	95.85±0.46	96.29±0.16	**96.37±0.24**	**96.52±0.33**
Total	66.23±0.59	63.69±1.47	65.87±4.32	68.78±1.62	66.74±2.08	**70.26±1.28**	66.62±1.78	**70.61±1.90**

that it ranks the top 1 in the nine transfer scenarios and the top 2 in the rest scenarios. In particular, when pretraining on the PTB-XL dataset and fine-tuning on four target domain datasets, the F1 scores of ASTCL reach 90.09%, 82.69%, 62.84%, and 70.01% respectively, all ranking first. Above all, our proposed ASTCL can improve the transferability of learned representations over the presupervised by about 1.37% on average in terms of F1 score, which consistently outperforms other baseline frameworks.

F. Semisupervised Evaluation

A series of semisupervised evaluation experiments are set up to test the semisupervised ability of ASTCL. The 5%, 10%, 20%, 50%, and 100% labeled data of the training dataset are randomly selected to fine-tune pretraining parameters. Fig. 9 shows the F1 scores of ASTCL and supervised training when training with different proportions of labeled data. The orange line indicates ASTCL fine-tuning and the blue line indicates supervised training.

We find that when the number of labeled training data is the same, ASTCL fine-tuning on any dataset performs better than supervised training. Especially in the case of a few labels, the effect of ASTCL improvement is significant. When training with 5% labeled data, compared with the supervised training, ASTCL is improved by 44.02%, 24.06%, 13.51%, 16.05%, and 23.37% on five datasets, respectively. In addition,

TABLE V
F1 SCORE OF TRANSFER LEARNING CROSS DOMAIN

Source domain Dataset	Chapman				PTB-XL			
Target domain Dataset	PTB-XL	CODE	CPSC2018	CMI	Chapman	CODE	CPSC2018	CMI
Pre-Supervised	53.55±1.59	81.49±1.22	61.87±0.76	64.69±4.34	88.14±0.55	79.31±0.89	59.94±1.27	67.85±0.49
CPC [19]	52.27±1.97	78.29±2.45	54.97±2.05	61.75±3.82	84.93±0.58	76.23±0.95	53.57±1.76	62.87±1.85
SimCLR [21]	54.05±1.37	80.42±0.67	55.28±1.53	62.13±3.18	81.26±2.56	81.01±2.05	56.79±2.31	62.97±2.74
BOYL [30]	56.01±0.49	80.85±0.64	59.70±2.16	67.07±1.25	88.52±0.74	**82.09±1.06**	61.33±0.95	**68.58±1.27**
SimSiam [15]	56.60±0.78	**83.12±0.73**	61.28±1.73	**67.21±1.33**	87.97±0.82	81.87±0.96	62.02±1.50	66.96±3.21
TS-TCC [13]	**57.52±0.95**	81.38±0.59	**62.41±1.12**	63.50±2.26	**89.18±1.06**	81.38±0.99	**62.22±1.25**	63.60±2.75
CLOCS [16]	56.12±0.34	80.20±0.17	61.03±1.27	67.05±2.07	87.19±1.29	79.90±0.53	61.29±1.49	66.90±1.61
ASTCL (ours)	**57.26±0.77**	**82.98±0.42**	**63.12±0.67**	**68.65±1.93**	**90.09±0.45**	**82.69±0.68**	**62.84±0.91**	**70.01±1.63**
Source domain Dataset	CODE				CPSC2018			
Target domain Dataset	Chapman	PTB-XL	CPSC2018	CMI	Chapman	PTB-XL	CODE	CMI
Pre-Supervised	91.13±0.56	55.00±1.27	65.57±0.80	69.08±1.36	89.59±0.73	56.36±0.51	82.26±0.45	66.76±2.89
CPC [19]	85.19±0.68	54.16±0.97	56.92±1.58	65.26±2.08	84.85±0.88	53.71±2.43	77.09±1.22	64.84±1.24
SimCLR [21]	83.25±2.63	56.67±1.24	58.47±3.28	67.31±1.32	84.02±1.42	55.30±2.52	81.02±0.68	62.61±3.02
BOYL [30]	88.58±0.66	56.27±0.49	61.34±1.86	66.05±3.39	87.66±0.51	55.74±0.54	81.25±0.55	**67.57±1.30**
SimSiam [15]	**90.31±0.82**	**57.49±1.26**	**64.28±0.58**	67.34±2.40	87.45±1.16	55.42±0.71	**82.29±1.67**	65.67±2.17
TS-TCC [13]	89.61±0.52	56.17±1.57	63.16±0.57	67.62±1.64	**89.03±0.35**	**56.23±0.72**	81.90±0.70	67.49±1.38
CLOCS [16]	87.80±1.06	55.70±0.46	60.09±1.92	**67.86±2.36**	87.85±0.95	54.73±0.55	79.42±0.65	65.93±1.98
ASTCL (ours)	**90.89±0.49**	**58.76±1.49**	**65.01±0.97**	**69.79±0.61**	**89.49±0.33**	**56.13±1.45**	**82.54±0.28**	**68.30±1.02**

TABLE VI
AUC OF LINEAR EVALUATION IN ABLATION STUDY

	EA	AG	RN	Chapman	PTB-XL	CODE	CPSC2018
ASTCL (-EA)		✓	✓	91.49±0.48	77.21±0.34	**92.08±0.29**	81.91±0.39
ASTCL (-AG)	✓		✓	**92.82±0.38**	**81.92±0.31**	91.67±0.24	83.89±0.54
ASTCL (-RN)	✓	✓		92.58±0.29	81.63±0.46	91.58±0.18	**83.93±0.33**
ASTCL	✓	✓	✓	**93.05±0.46**	**82.03±0.65**	**92.18±0.27**	**84.24±0.46**

on the four-fifths semisupervised scenarios, the performance of ASTCL fine-tuning with only 50% labeled data is better than that of supervised training using 100% labeled data. This means that ASTCL can use limited labeled data to pretrain outstanding models.

G. Ablation Study

Compared with other contrastive learning studies, the main improvements of our work are ECG augmentations, adversarial games, and only using patient-level positive pairs. Although the ECG augmentations have been demonstrated in the augmentation evaluation, we replace ECG augmentations with the data augmentations of [13], namely ASTCL (-EA), to study the contribution of ECG augmentations to ASTCL. To examine the effectiveness of the adversarial game, the adversarial game is taken out from ASTCL, which is expressed as ASTCL (-AG). To test whether it is useful by only using the patient-level positive pairs, we delete the predictor and update the gradient of projection, and the NT-Xent loss function of [21] is used instead of our $\mathcal{L}_C$. This modified framework is called ASTCL (-RP). We carry out the linear evaluation, noise evaluation, and category evaluation in the ablation study to verify the three improvements.

Table VI shows linear evaluation results of the ablation study on four datasets. We observe that the performance of the complete ASTCL remains the best. However, the overall performance of ASTCL (-EA) decreases obviously. Especially on the PTB-XL dataset, the AUC of ASTCL (-EA) decreases by 4.82%. This indicates that ASTCL (-EA) is greatly affected without ECG noise enhancement and ECG noise denoising. By using spatiotemporal prediction and patient discrimination with only patient-level positive pairs, ASTCL (-AG) achieves the second-best AUC. But due to the lack of an adversarial game, ASTCL (-AG)'s performance is not robust. After using instance-level positive pairs and employing negative pairs, the performance of ASTCL (-RN) is also weaker than that of complete ASTCL on all datasets. This is obvious because negative pairs of the same category can weaken the category representation learning ability of the model. In a word, the three improvements of ASTCL are indispensable during pretraining. Meanwhile, the noise evaluation and category evaluation in the ablation study prove that the adversarial game and only using patient-level positive pairs play a crucial role in improving antiperturbation ability and increasing category representation learning ability, respectively (see supplementary material). In addition, this article also applies these improvements to other frameworks and discusses their availability in supplementary material.

To analyze the selected weight of the $\mathcal{L}_F$, this article also carries out a series of experiments on the selected weight of the $\mathcal{L}_F$. One weight is changed from 0.001 to 1000, and the other two weights are fixed to 1. Fig. 10 shows the AUCs of w_1, w_2, and w_3 under each value, respectively. When w_1, w_2, and w_3 are all set to 1, the results of ASTCL are the best.

In general, the performance of the complete ASTCL is the best, which means that our improvements in ECG augmentations, adversarial games, and only using patient-level positive pairs are effective. In the optimization of $\mathcal{L}_F$, the chosen weight is also correct.

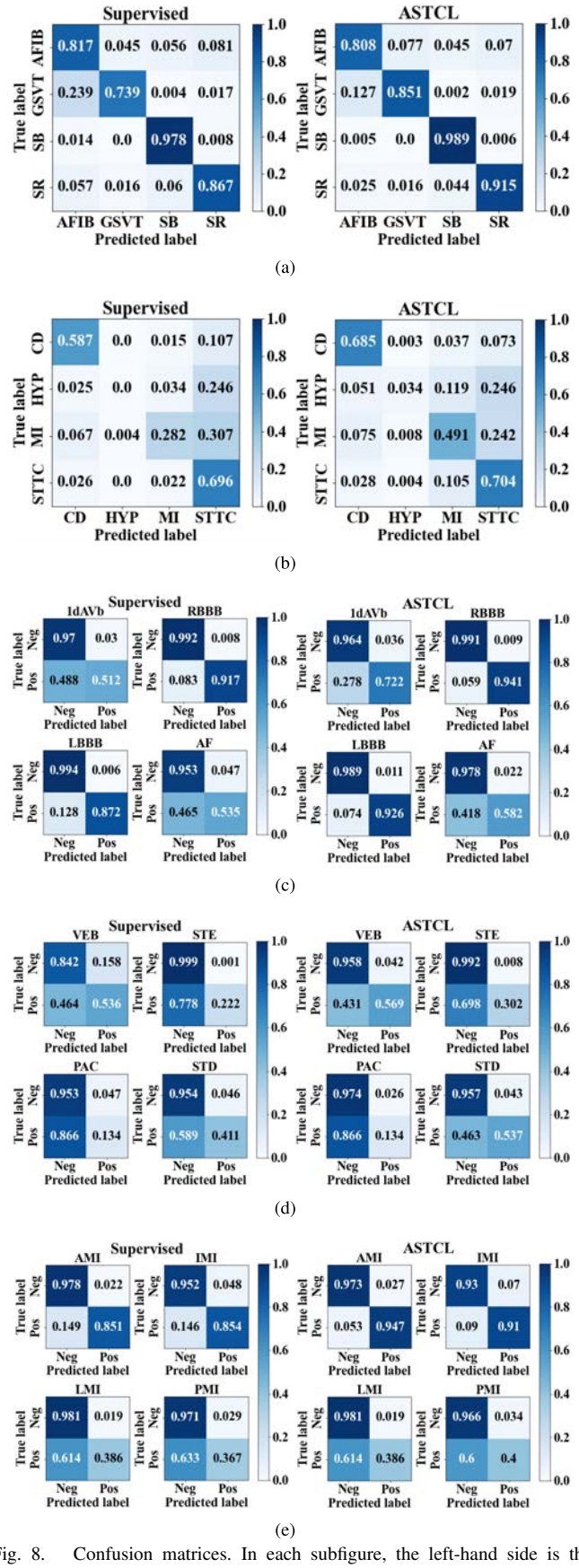


Fig. 8. Confusion matrices. In each subfigure, the left-hand side is the supervised training results, and the right-hand side is the ASTCL fine-tuning results. Among them, "Pos" represents positive and "Neg" represents negative. (a) Chapman. (b) PTB-XL. (c) CODE. (d) CPSC2018. (e) CMI.

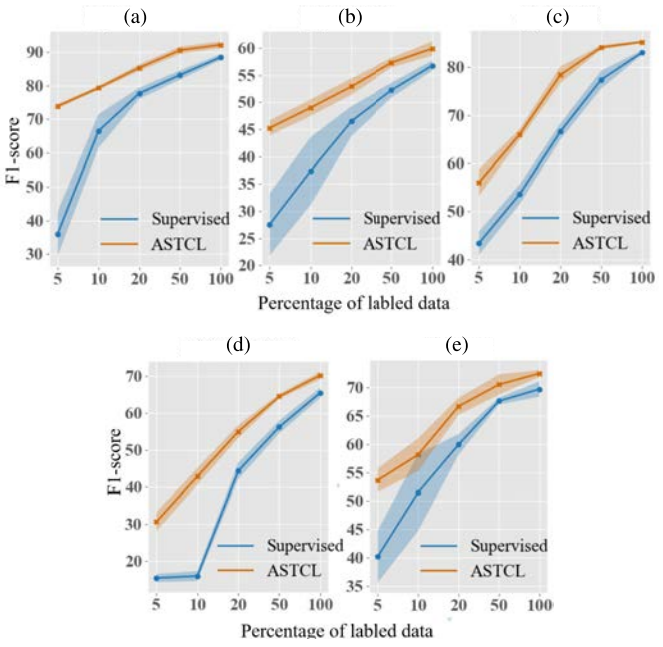


Fig. 9. F1-score of supervised training and ASTCL fine-tuning in semisupervised experiments on the same amount of labeled data. (a) Chapman. (b) PTB-XL. (c) CODE. (d) CPSC2018. (e) CMI.

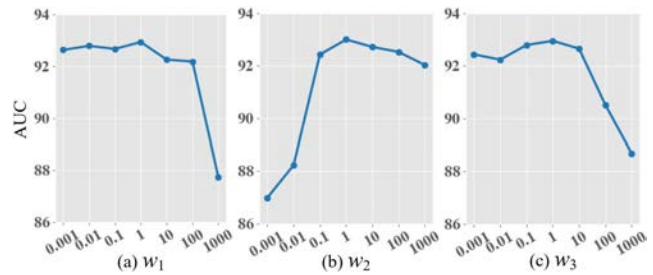


Fig. 10. Weight analysis of loss function on Chapman dataset. (a) w_1. (b) w_2. (c) w_3.

VI. Conclusion

This article proposes a patient-level ASTCL framework for unsupervised representation learning in ECG signals, which consists of ECG augmentations, an adversarial module, and a spatiotemporal contrastive module. The goal of this framework is to improve noise robustness and learn the spatiotemporal and semantic representations of categories. To reduce the impact of noise, the proposed ECG augmentations generate two distinct but effective augmented views for each data via ECG noise enhancement and ECG noise denoising. The aim of the adversarial module is to pull the representations into the shared distribution between positive pairs by an adversarial game task to discard the perturbation representations and learn the invariant representations. The spatiotemporal contrastive module learns the spatiotemporal and semantic representations of categories by spatiotemporal prediction and patient discrimination. To better learn category representations, we only employ patient-level positive pairs and alternately utilize the predictor and the stop-gradient to replace negative pairs.

Extensive experiments are presented on four ECG benchmark datasets and one clinical dataset. The experiments show that ASTCL outperforms the state-of-the-art methods in terms

of augmentation effect, pretraining ability, antiperturbation ability, category learning ability, transfer ability, and semisupervised ability. The main components of ASTCL profit to better and more stable learning representations of ECG.

Heartbeat-level classification is an important task in diagnosing arrhythmias. In the future, we will probe the self-supervised task of capturing heartbeat-level features and assembling them into the proposed ASTCL. According to the features of the heartbeat, a general augmented paradigm will be explored. In addition, the proposed ASTCL would be extended to the study of unsupervised clustering in ECG.

REFERENCES

[1] WHO. (2021). *Cardiovascular Diseases (CVDS)*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] A. Y. Hannun et al., "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[3] Z. I. Attia et al., "Screening for cardiac contractile dysfunction using an artificial intelligence-enabled electrocardiogram," *Nature Med.*, vol. 25, pp. 70–74, Jan. 2019.

[4] H. Wang, Y. Zhou, B. Zhou, X. Niu, H. Zhang, and Z. Wang, "Interactive ECG annotation: An artificial intelligence method for smart ECG manipulation," *Inf. Sci.*, vol. 581, pp. 42–59, Dec. 2021.

[5] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[6] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 69–84.

[7] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2018, pp. 1–16.

[8] M. Vo, E. Yumer, K. Sunkavalli, S. Hadap, Y. Sheikh, and S. G. Narasimhan, "Self-supervised multi-view person association and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2794–2808, Aug. 2021.

[9] N. D. Q. Bui, Y. Yu, and L. Jiang, "Self-supervised contrastive learning for code retrieval and summarization via semantic-preserving transformations," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 11–15.

[10] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[12] S. Gidaris, P. Singh, and N. Komodakis, "Golany 12-lead ECG reconstruction via Koopman operators," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 3745–3754.

[13] E. Eldele et al., "Time-series representation learning via temporal and contextual contrasting," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2352–2359.

[14] X. Liu, H. Wang, and Z. Li, "An approach for deep learning in ECG classification tasks in the presence of noisy labels," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 369–372.

[15] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[16] D. Kiyasseh, T. Zhu, and D. A. Clifton, "CLOCS: Contrastive learning of cardiac signals across space, time, and patients," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 5606–5615.

[17] A. Hyvärinen and H. Morioka, "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 3765–3773.

[18] J. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2019, pp. 4652–4663.

[19] A. Van Den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[20] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2019, pp. 1–24.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, Nov. 2020, pp. 1597–1607.

[22] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 776–794.

[23] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 1–12.

[24] C. Ho and N. Vasconcelos, "Contrastive learning with adversarial examples," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 1–17.

[25] R. Zhu, B. Zhao, J. Liu, Z. Sun, and C. W. Chen, "Improving contrastive learning by visualizing feature transformation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10286–10295.

[26] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 1–21.

[27] H. Xu et al., "Seed the views: Hierarchical semantic alignment for contrastive representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3753–3767, Mar. 2023.

[28] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2378–2392, Jun. 2022.

[29] N. Kermiche, "Contrastive Hebbian feedforward learning for neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2118–2128, Jun. 2020.

[30] J. Grill et al., "Bootstrap your own latent a new approach to self-supervised learning," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 1–35.

[31] Z. Gao et al., "EEG-based spatio-temporal convolutional neural network for driver fatigue evaluation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2755–2763, Sep. 2019.

[32] P. Sarkar and A. Etemad, "Self-supervised ECG representation learning for emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1541–1554, Jul. 2022.

[33] H. Banville, I. Albuquerque, A. Hyvärinen, G. Moffat, D. Engemann, and A. Gramfort, "Self-supervised representation learning from electroencephalography signals," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Oct. 2019, pp. 1–6.

[34] H. Banville, O. Chehab, A. Hyvärinen, D. A. Engemann, and A. Gramfort, "Uncovering the structure of clinical EEG signals with self-supervised learning," *J. Neural Eng.*, vol. 18, no. 4, pp. 1–23, Mar. 2021.

[35] H. Fan, F. Zhang, R. Wang, X. Huang, and Z. Li, "Semi-supervised time series classification by temporal relation prediction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3545–3549.

[36] H. Fan, F. Zhang, and Y. Gao, "Self-supervised time series representation learning by inter-intra relational reasoning," 2020, *arXiv:2011.13548*.

[37] H. Zhao, Q. Zheng, K. Ma, H. Li, and Y. Zheng, "Deep representation-based domain adaptation for nonstationary EEG classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 535–545, Feb. 2021.

[38] Q. Ma, S. Li, W. Zhuang, S. Li, J. Wang, and D. Zeng, "Self-supervised time series clustering with model-based dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3942–3955, Sep. 2021.

[39] J. Y. Cheng, H. Goh, K. Dogrusoz, O. Tuzel, and E. Azemi, "Subject-aware contrastive learning for biosignals," 2020, *arXiv:2007.04871*.

[40] E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy, and S. L. Hyland, "Contrastive representation learning for electroencephalogram classification," in *Proc. Mach. Learn. Health Workshop (ML4H)*, Dec. 2020, pp. 238–253.

[41] X. Shen, X. Liu, X. Hu, D. Zhang, and S. Song, "Contrastive learning of subject-invariant EEG representations for cross-subject emotion recognition," *IEEE Trans. Affect. Comput.*, early access, Apr. 4, 2022, doi: 10.1109/TAFFC.2022.3164516.

[42] X. Lan, D. Ng, S. Hong, and M. Feng, "Intra-inter subject self-supervised learning for multivariate cardiac signals," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2021, pp. 4532–4540.

[43] J. Zhang et al., "Automated localization of myocardial infarction of image-based multilead ECG tensor with Tucker2 decomposition," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.

[44] Z. Liu, H. Wang, Y. Gao, and S. Shi, "Automatic attention learning using neural architecture search for detection of cardiac abnormality in 12-lead ECG," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
[45] N. Seeuws, M. De Vos, and A. Bertrand, "Electrocardiogram quality assessment using unsupervised deep learning," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 882–893, Feb. 2022.
[46] T. Mehari and N. Strodthoff, "Self-supervised representation learning from 12-lead ECG data," *Comput. Biol. Med.*, vol. 141, pp. 1–11, Feb. 2022.
[47] Y. Hsu, J. Wang, W. Chiang, and C. Hung, "Automatic ECG-based emotion recognition in music listening," *IEEE Trans. Affect. Comput.*, vol. 11, no. 1, pp. 85–99, Jan. 2020.
[48] G. Moody, W. Muldrow, and R. Mark, "A noise stress test for arrhythmia detectors," in *Proc. Comput. Cardiol. (CinC)*, Feb. 1984, pp. 381–384.
[49] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 1, pp. 85–98, Jan. 1990.
[50] M. B. Hossain, S. K. Bashar, J. Lazaro, N. Reljin, Y. Noh, and K. H. Chon, "A robust ECG denoising technique using variable frequency complex demodulation," *Comput. Methods Programs Biomed.*, vol. 200, pp. 1–27, Mar. 2021.
[51] H. Li, "Research of coronary artery disease detection based on ensemble deep learning of two-modal signals," Ph.D. dissertation, Shandong Univ., Jinan, China, 2020.
[52] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2672–2680.
[53] P. Feng et al., "Unsupervised semantic-aware adaptive feature fusion network for arrhythmia detection," *Inf. Sci.*, vol. 582, pp. 509–528, Jan. 2022.
[54] T. Golany and K. Radinsky, "PGANs: Personalized generative adversarial networks for ECG synthesis to improve patient-specific deep ECG classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Jul. 2019, pp. 557–564.
[55] A. Vaswani et al., "Attention is all you need," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 5998–6008.
[56] Q. Wang et al., "Learning deep transformer models for machine translation," in *Proc. Meeting Assoc. Comput. Linguistics (ACL)*, Jul. 2019, pp. 1810–1822.
[57] Z. Ge et al., "Multi-label correlation guided feature fusion network for abnormal ECG diagnosis," *Knowl.-Based Syst.*, vol. 233, pp. 1–10, Sep. 2021.
[58] J. Zheng, J. Zhang, S. Danioko, H. Yao, H. Guo, and C. Rakovski, "A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients," *Sci. Data*, vol. 7, no. 1, pp. 1–8, Feb. 2020.
[59] P. Wagner et al., "PTB-XL, a large publicly available electrocardiography dataset," *Sci. Data*, vol. 7, no. 1, pp. 1–15, May 2020.
[60] A. H. Ribeiro et al., "Automatic diagnosis of the 12-lead ECG using a deep neural network," *Nature Commun.*, vol. 11, no. 1, pp. 1–9, Apr. 2020.
[61] F. Liu et al., "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection," *J. Med. Imag. Health Informat.*, vol. 8, no. 7, pp. 1368–1373, Sep. 2018.
[62] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[63] J. Wang et al., "Adversarial de-noising of electrocardiogram," *Neurocomputing*, vol. 349, pp. 212–224, Jul. 2019.

Ning Wang (Member, IEEE) received the B.S. and M.S. degrees in computer science from Henan University, China, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China.

He is currently a Research with the Cooperative Innovation Center for Internet Healthcare, Zhengzhou University. His research interests include representation learning and physiological signal processing.

Panpan Feng (Member, IEEE) received the B.S. degree in statistics from Zhengzhou University, Zhengzhou, China, in 2018, where she is currently pursuing the Ph.D. degree.

She is currently a Research with the Cooperative Innovation Center for Internet Healthcare, Zhengzhou University. Her research interests include artificial intelligence, intelligence healthcare, and transfer learning.

Zhaoyang Ge received the B.S. and M.S. degrees in computer technology from Zhengzhou University, Zhengzhou, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree.

He is currently a Researcher with the Cooperative Innovation Center for Internet Healthcare, Zhengzhou University. His research interests include intelligence healthcare, deep learning, and computer vision.

Yanjie Zhou (Member, IEEE) received the B.S. and M.S. degrees in computer science and computer applied technology from Zhengzhou University, Zhengzhou, China, in 2012 and 2015, respectively, and the Ph.D. degree from the Department of Industrial Engineering, Pusan National University, Busan, South Korea, in 2020.

He is currently an Associate Professor with the School of Management, Zhengzhou University. His research interests include optimization problems in industrial engineering, game theory, and intelligence healthcare.

Bing Zhou received the B.S. and M.S. degrees in computer science from Xi'an Jiaotong University, Xi'an, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2003.

He is currently a Professor with the School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, China. His current research interests include intelligent diagnosis of electrocardiograms, computer vision, Internet medical, and multimedia applications.

Zongmin Wang received the Ph.D. degree in engineering from Tsinghua University, Beijing, China, in 1996.

From 1996 to 1997, he was with The University of Hong Kong, Hong Kong, where he was involved in collaborative scientific research. From 1997 to 2017, he served as a Professor and a Ph.D. Tutor with Zhengzhou University, Zhengzhou, China, where he is currently the Director of the Cooperative Innovation Center for Internet Healthcare. His research interests include intelligent diagnosis of electrocardiograms, Internet medical, computer networks, and intelligence healthcare.