LEAVES: Learning Views for Time-Series Biobehavioral Data in Contrastive Learning

Han Yu HAN.YU@RICE.EDU
Rice University, USA

Huiyuan Yang HYANG@MST.EDU
Missouri University of Science & Technology, USA

Akane Sano AKANE.SANO@RICE.EDU
Rice University, USA

Abstract

Contrastive learning has been utilized as a promising self-supervised learning approach to extract meaningful representations from unlabeled data. The majority of these methods take advantage of data-augmentation techniques to create diverse views from the original input. However, optimizing augmentations and their parameters for generating more effective views in contrastive learning frameworks is often resource-intensive and time-consuming. While several strategies have been proposed for automatically generating new views in computer vision (Tamkin et al., 2020; Rusak et al., 2020), research in other domains, such as time-series biobehavioral data, remains limited. In this paper, we introduce a simple yet powerful module for automatic view generation in contrastive learning frameworks applied to time-series biobehavioral data, which is essential for modern health care, termed **lea**rning **vie**ws for time-**s**eries data (LEAVES). This proposed module employs adversarial training to learn augmentation hyperparameters within contrastive learning frameworks. We assess the efficacy of our method on multiple time-series datasets using two well-known contrastive learning frameworks, namely *SimCLR* and *BYOL*. Across four diverse biobehavioral datasets, LEAVES requires only 20 learnable parameters—dramatically fewer than the 580,000 parameters demanded by frameworks like ViewMaker, previously proposed adversarially trained convolutional module in contrastive learning, while achieving competitive and often superior performance to existing baseline methods. Crucially, these efficiency gains are obtained without extensive manual hyperparameter tuning, which makes LEAVES particularly suitable for large-scale or real-time healthcare applications that demand both accuracy and practicality. The code of this work is available at: https://github.com/comp-well-org/LEAVES.

1. Introduction

Modern sensing devices collect continuous time-series data from the human body and provide opportunities for monitoring health and behavior. Researchers have demonstrated that such time-series data are crucial for the development of innovative methods for patient monitoring, diagnosis, and treatment (Goldberger et al., 2000). For instance, with machine learning algorithms, electrocardiogram (ECG) has been used in disease diagnosis such as arrhythmia (Śmigiel et al., 2021; Li et al., 2021; Zhang et al., 2021) and sleep apnea (Fang et al., 2022; Chang et al., 2020; Singh and Majumder, 2019); electroencephalograph (EEG)

has been leveraged for the sleep stage recognition (Perslev et al., 2019; Mousavi et al., 2019; Phan et al., 2022).

While the aforementioned methods generally require decent amounts of high-quality annotations to construct reliable and robust machine learning models, acquiring such labels for training deep learning models on biobehavioral data is often challenging. This shortage has encouraged researchers to leverage unsupervised pre-training approaches. Consequently, self-supervised learning techniques, including contrastive learning, have been widely used to improve the robustness of the model (Yuan et al., 2022; Mehari and Strodthoff, 2022; Sarkar and Etemad, 2020; Shah et al., 2021). For instance, Shah et al. (Shah et al., 2021) employ contrastive learning frameworks such as SimCLR (Chen et al., 2020a) and BYOL (Grill et al., 2020) to maximize the agreement between the original and augmented electrocardiogram (ECG) samples. This pre-training model outperforms supervised baselines in several downstream health-related tasks, including the detection of sleep apnea, hypertension, and diabetes. Data augmentation is a key element of the contrastive learning methods, which aims to create diverse modifications of the original input for pre-text task. However, selecting effective data augmentation methods for the diverse types of time series biobehavioral data remains an open challenge (Yang et al., 2022).

To address the challenge, different approaches have been proposed to find more efficient ways to search for more effective data augmentation methods, such as (Tamkin et al., 2020; Hu et al., 2021; Qiu et al., 2021; Rusak et al., 2020). These methods generate reasonably corrupted views for image datasets that improve model performance. For example, (Tamkin et al., 2020) introduced ViewMaker, an adversarially trained convolutional module in contrastive learning, to generate augmentations for images. However, methods such as ViewMaker may not be suitable for time-series biobehavioral data. First, image-based methods introduce uninterpretable noise to the original signal and result in unfaithful augmented views, where the signal waveform becomes overcorrupted. Second, previous methods primarily focus mainly on distortions to the signal's magnitude, which neglects critical temporal and frequency domain information (Um et al., 2017; Yang et al., 2022; Mehari and Strodthoff, 2022; Raghu et al., 2022).

In this work, we introduce LEAVES (Learning Views for Time-Series Data), a lightweight module that automates augmentation policy tuning for time-series biobehavioral data within contrastive learning frameworks. LEAVES is adversarially trained to generate challenging yet faithful augmentations, which enhances the ability of encoders to learn robust representations. Further, to aid in the extraction of meaningful representation from both the temporal and frequency domains, we introduce two novel differentiable augmentation approaches, Time Modulation *(TimeM)* and Frequency Suppression *(FreqSup)*, that can provide appropriate and smooth distortions in the temporal and frequency domains respectively. Further, comprehensive experiments demonstrate that LEAVES achieves competitive performance compared to manually fine-tuned augmentation policies within SimCLR and BYOL frameworks while significantly reducing the search cost for optimal augmentations. Our contributions can be summarized as follows:

- We introduce LEAVES, a novel method for automatically learning effective views in contrastive learning frameworks for time-series biobehavioral data. To our knowledge, this is the first study to explore automatic data augmentation for contrastive learning using time-series biobehavioral data.

- We propose two differentiable data augmentation methods including *TimeM* and *FreqSup* for adding subtle time-domain and frequency-domain transformation to the data.
- Our comprehensive experiments demonstrate that LEAVES achieves competitive performance compared to manually fine-tuned augmentation policies within SimCLR and BYOL frameworks while significantly reducing the search cost for optimal augmentations.

Generalizable Insights about Machine Learning in the Context of Healthcare

Time-series biobehavioral signals—including ECG, EEG, and Inertial Measurement Unit (IMU) data—are increasingly pivotal in modern, machine-learning-driven healthcare systems. Nevertheless, the scarcity of high-quality labels often impedes fully supervised approaches, which causes a shift toward self-supervised learning for more robust representation learning. In practice, many contrastive learning methods still adopt suboptimal data augmentation policies, primarily because tuning these policies can be both time- and resource-intensive. LEAVES addresses this challenge by integrating data augmentation policy tuning directly into the neural network and optimizing these policies via adversarial training—all with minimal overhead. This simple and computationally efficient design can be applied to a wide array of time-series biobehavioral signals for various downstream tasks. Moreover, we show that LEAVES consistently achieves competitive or even outperforms manually tuned augmentation policies and state-of-the-art contrastive learning baselines, while introducing only a small number of extra parameters.

2. Related Work

2.1. Augmentation-Based Contrastive Learning

Contrastive learning algorithms have been leveraged in cutting-edge self-supervised deep learning methods, with data augmentation serving as a core element in generating diverse views from original inputs to create contrastive pairs. A series of contrastive learning frameworks developed for image transformation within computer vision applications have been introduced (He et al., 2020; Chen et al., 2020a; Grill et al., 2020; Chen and He, 2021; Tamkin et al., 2020; Zbontar et al., 2021; Wang and Qi, 2022; Zhang and Ma, 2022). Among them, SimCLR (Chen et al., 2020a) and BYOL (Grill et al., 2020) are the two most widely used frameworks. For instance, SimCLR (Chen et al., 2020a) aims to enhance the agreement between two distinct augmented views of a single image, while BYOL (Grill et al., 2020) employs a cooperative learning environment between a target and an online network using two transformed views of an image. Other methods, such as Barlow Twins (Zbontar et al., 2021), VICReg (Bardes et al., 2021), MoCo-v2 (Chen et al., 2020b), MoCo-v3 (Chen et al., 2021), and SimSiam (Chen and He, 2021), have further refined the principles of contrastive and redundancy-reduction learning. Although these newer approaches have shown promise, their adaptation to non-visual modalities, particularly time-series biobehavioral data, remains limited.

2.2. Contrastive Learning in Time-Series Biobehavioral Data

The advances in self-supervised learning have inspired the adaptation of contrastive learning techniques to time-series biobehavioral data (Yue et al., 2022; Gopal et al., 2021; Mehari and Strodthoff, 2022; Wickstrøm et al., 2022; Eldele et al., 2021a; Yue et al., 2022; Luo et al., 2023; Hallgarten et al., 2023). For example, Gopal *et al.* (Gopal et al., 2021) proposed a domain-knowledge-infused augmentation for ECG data, formulating views conducive to contrastive learning. Mehar *et al.* (Mehari and Strodthoff, 2022) extended established methods including SimCLR, BYOL, and CPC (Oord et al., 2018) to time-series ECG data to improve clinical task performance. Hallgarten et al. (2023) adapted MoCo in EEG data for human activities recognition. Despite these advancements, the empirical selection of data augmentations for view generation is not always optimal, especially for new or less studied datasets, which posts the exploration of augmentation policy as a costly problem.

Although carefully selected augmentations can improve model performance(Yue et al., 2022; Mehari and Strodthoff, 2022), inappropriate augmentation policies, on the other hand, can negatively impact model performance, particularly with time series biobehavioral data (Yang et al., 2022). A robust approach to selecting data augmentation policies is essential in contrastive learning applications, especially with sensitive time-series biobehavioral data. Nevertheless, to the best of our knowledge, there is no existing study that focuses on augmentation policy exploration for time-series biobehavioral data in contrastive learning.

2.3. Automatic Augmentation

Several methods have been proposed to improve augmentation strategies, providing alternatives to traditional empirical approaches (Cubuk et al., 2019; Ho et al., 2019; Lim et al., 2019; Li et al., 2020; Cubuk et al., 2020; Liu et al., 2021). AutoAugment (Cubuk et al., 2019), for instance, employs a reinforcement learning approach to traverse through augmentation policies and optimize the weights and orders of diverse augmentation techniques. DADA (Li et al., 2020) uses a gradient-based optimization approach to identify the most effective augmentation policy during training, significantly reducing training time compared to earlier methods.

In addition to exploring the augmentation policy space, several studies have investigated auto-generating views, where data transformations are generated by neural networks (Tian et al., 2020; Rusak et al., 2020; Tamkin et al., 2020). Specifically, Rusak et al. (2020) utilized a CNN model to introduce noise into the input data and adversarially optimize the perturbation generator with respect to a supervised loss. Similarly, Tamkin et al. (2020) implemented a ResNet-based ViewMaker module to generate views for contrastive learning. However, these methods are primarily capable of altering the amplitude of the original signals and are less effective in modulating crucial temporal and frequency domain information in time-series data. In contrast, our proposed LEAVES module not only modulates the original signals in both time and frequency domains but also ensures that the augmented views remain faithful to the underlying biobehavioral patterns—a critical requirement for clinical applications.

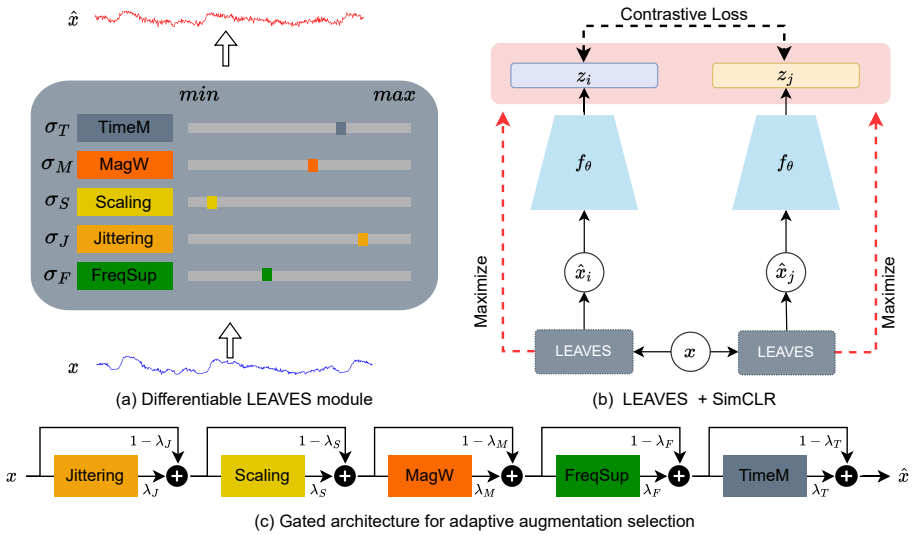

Figure 1: Overview of LEAVES integrated within the SimCLR contrastive learning framework. (a) LEAVES generates augmented views ($\hat{x}$) from input x using differentiable transformations with learnable intensity (σ) parameters. (b) Augmented views are encoded to representations (z) via encoder f_θ. The framework is trained adversarially with a contrastive loss. (c) Gating network enables adaptive augmentation selection by activating/deactivating transformations based on learned λ values.

3. Methodology

In this study, we develop our method on two well-known contrastive learning algorithms, including SimCLR (Chen et al., 2020a) and BYOL (Grill et al., 2020). The overall architecture for the pre-training method demonstrated with SimCLR is illustrated in Figure 1. Both SimCLR and BYOL utilize 1D ResNet18 as encoders.

In this section, we first introduce a differentiable LEAVES module designed to generate challenging yet faithful views for time-series biobehavioral inputs. Following this, we detail how the LEAVES module is seamlessly integrated into the contrastive learning framework to enable efficient view generation. Then, we describe the adversarial training method to optimize LEAVES with contrastive learning frameworks.

3.1. LEAVES

This section introduces LEAVES, a lightweight module designed for seamless integration into existing contrastive learning frameworks. LEAVES generates challenging views while maintaining the original input's essential waveforms. It employs various differentiable data augmentation methods, including *Jittering* ($\mathcal{T}_J$), *Scaling* ($\mathcal{T}_S$), magnitude warping (*MagW*, $\mathcal{T}_M$), time modulation (*TimeM*, $\mathcal{T}_T$), and frequency suppression (*FreqSup*, $\mathcal{T}_F$). These augmentations are sequentially applied as indicated by the symbol $\odot$. For instance, $\mathcal{T}_J \odot \mathcal{T}_P$ indicates the application of jittering noise followed by data permutation.

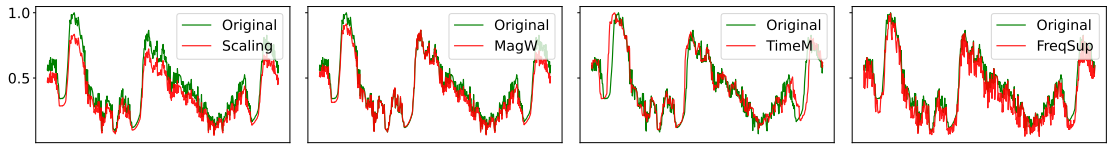

Figure 2: Examples of different data augmentation methods for time-series biobehavioral data (magnitude $\sigma = 0.03$).

For a given dataset $x \in \mathbb{R}^{C \times L}$, where L represents the length of time series data and C the number of channels, the transformed view $\hat{x}$ is represented as:

$$\begin{aligned}\hat{\mathbf{x}} = \mathbf{x} \odot (\lambda_J \mathcal{T}_J(\sigma_J)) \odot (\lambda_S \mathcal{T}_S(\sigma_S)) \odot (\lambda_M \mathcal{T}_M(\sigma_M)) \\ \odot (\lambda_F \mathcal{T}_F(\sigma_F)) \odot (\lambda_T \mathcal{T}_T(\sigma_T))\end{aligned} \tag{1}$$

Here, σs are the hyperparameters controlling the intensity of augmentations applied to the original sample with varying ranges across augmentations. For instance, σs $\in [0, 0.10]$ specifies the standard deviation for noise generation with *Jittering*, *Scaling*, *MagW*, and *TimeM*; whereas the $\sigma_F \in [0, 1]$ is the suppression rate that controls the intensity of *FreqSup*. LEAVES not only seeks to fine-tune the optimal values of σs but also introduces a gated network architecture with gating parameters λs that enables adaptive augmentation selection via sigmoid activation. This activation produces values always close to 0 or 1 by introducing the steepness-controlling value into the sigmoid, which enables differentiable "soft" gating. We initially set λs to 1, which initially enables all the augmentations initially. This is achieved by initializing the learnable pre-sigmoid inputs, denoted as σ_k for each gate k, to a sufficiently large positive value. Figure 5 (lower row) illustrates the evolution of these gate statuses, starting from the 'on' state for all augmentations, which corresponds to λ_k close to 1. These gates can then be learned as "on" or "off" during training. This enhances LEAVES' ability to autonomously determine the most effective augmentation combinations for any given input. LEAVES aims to fine-tune the optimal values of σs and λs, which optimizes both the intensity and presence of each augmentation method. This approach encourages the creation of various views by learning combinations of different augmentation techniques. The order of augmentations in Equation 1 is fixed in this study, as the model performance remains robust within the change of order, as indicated in Supplemental Material A.1.

The LEAVES module, along with the representation encoder, is trained in an adversarial manner within the SimCLR framework as depicted in Figure 1. LEAVES aims to generate views that minimize agreement between representation pairs, whereas the encoder is learning to maximize agreement among different views. This adversarial process encourages the encoder to learn robust representations that capture the underlying signal despite the transformations introduced by LEAVES.

3.1.1. Differentiable Data Augmentations for Time-Series Data

LEAVES incorporates a set of data augmentation approaches, with each providing different distortions from the original signal. For example, *Jittering, Scaling*, and *MagW* intro-

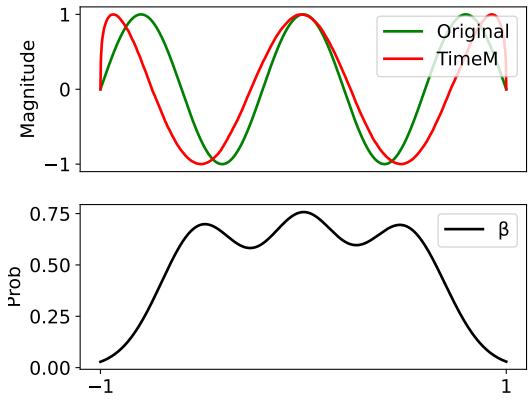

Figure 3: An illustration of time modulation augmentation deploying three Gaussian components. Areas of higher probability in the Gaussian mixture models are up-sampled in the transformation, and conversely, those of lower probability are down-sampled.

duce variations in the magnitude of the original signals; while *TimeM* and *Perm* alter the temporal structure, and *FreqSup* alters information in the frequency domain.

To efficiently tune the hyperparameters of augmentation methods, we design to make these hyperparameters tune within the training process. Nevertheless, a challenge arises from the non-differentiable nature of operations such as random value sampling and indexing in these methods. We address this by applying reparameterization techniques to make these operations differentiable. Further, to introduce the temporal distortion, we introduce the *TimeM* method for introducing temporal distortion, as described in the following section and demonstrated in Figure 2. Additionally, we propose a frequency-based augmentation method named frequency suppression (*FreqSup*) to modify the signal by changing frequency domain information. To avoid the generated views being over-corrupted from the original signals, we set constraints on the augmentation methods: a limit of $\eta = 0.10$ on the σ values for magnitude-based methods, a maximum value of standard deviation from 0 - 1.0 in each component of GMM *TimeW*, and a minimum σ value of 0 in *FreqSup*.

Jittering infuses the original signal with randomly generated noise with a Normal distribution:

$$\hat{\mathbf{x}} = \mathbf{x} + \epsilon_J, \quad \epsilon_J \sim \mathcal{N}(0, \sigma_J^2) \tag{2}$$

Here, $\epsilon_J \in \mathbb{R}^{C \times L}$ represents the noise introduced, with σ_J as the adjustable hyperparameter.

Scaling manipulates the signal's amplitude, entailing multiplication by randomly derived factors specific to each channel:

$$\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_C] = [\epsilon_S^1 x_1, \epsilon_S^2 x_2, ..., \epsilon_S^C x_C] \tag{3}$$

$\epsilon_S \sim \mathcal{N}(1, \sigma_S^2) \in \mathbb{R}^C$ denotes the sampled factors, with σ_S as the essential hyperparameter.

MagW (Magnitude Warping) (Um et al., 2017) involves distorting the magnitude of the original signal with a randomly generated smooth curve. First, we sample k nodes

from $\mathcal{N}(1, \sigma_M^2)$, which yields $knot \in \mathbb{R}^{C \times k}$. Then, we interpolate $knot$s evenly with a linear function producing ϵ_M. The modified view can be denoted as:

$$\hat{\mathbf{x}} = \mathbf{x} + \epsilon_M \tag{4}$$

We keep k to 8 and concentrate on the hyperparameter σ_M in this study.

TimeM modifies the temporal location within the original sequences by generating probabilities to determine which locations in the original signals should be sampled. By using a reparameterized Gaussian mixture model with M components, represented as $\sum_i^M \phi_i \mathcal{N}(\mu_i, \sigma_i^2)$, *TimeM* generates probabilities $\beta \in \mathbb{R}^{C \times L}$ ranging from 0 to 1 to warp the temporal positions. μ controls the center of each warping region, and σ defines its width, which affects the intensity of time warping around μ. Regions with higher β are upsampled, while those with lower β are downsampled. An example of *TimeM* is shown in Figure 3, -1 corresponds to the first time step (position 1) in the original signal, while 1 is the last time step (position L).

In this study, we empirically fix $M = 7$ and set the μ values at [-0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75]. This setting is based on preliminary hyperparameter tuning that suggests that evenly distributed seven components between -1 to 1, which offers a balanced trade-off between model complexity and the ability to capture variations in the signal. On the other hand, we make the standard deviation σ a target parameter for LEAVES to learn.

FreqSup is a frequency-based augmentation method that modifies the signal by manipulating its frequency components. This method employs a soft mask to selectively suppress certain frequencies. The mask is designed to preserve frequencies bands with high amplitudes to prevent excessive distortion of the original signal. First, a suppression mask ($MASK_S$) is defined to introduce the probability of suppression.

$$\text{MASK}_S = sigmoid\left((U(0,1) - \sigma_F)\right) \tag{5}$$

where $U(0,1)$ is a uniform random variable between 0 and 1, σ_F is a learnable parameter that controls the suppression rate. In addition, we introduce a projection mask ($MASK_P$) to preserve high-amplitude frequency components.

$$\text{MASK}_P = 0.5 \cdot \left(\tanh\left(\frac{\text{Amp}}{\max(\text{Amp})}\right) + 1\right) \tag{6}$$

where 'Amp' represents the amplitude of the frequency spectrum of the signal

$$\text{MASK}_F = (1 - \text{MASK}_P) \cdot \text{MASK}_S \tag{7}$$

MASK_F is the combination of MASK_S and MASK_P.

$$\hat{\mathbf{x}} = \text{IFFT}\left(\text{FFT}(\mathbf{x}) \cdot \text{MASK}_F\right) \tag{8}$$

FFT and IFFT denote the Fast Fourier Transform and its inverse, respectively.

Thus, the *FreqSup* method corrupts the signal by suppressing low-amplitude components in the frequency domain, which are determined by σ_F. Smaller values of σ_F lead to a boarder range of affected frequency bands, which then results in a stronger augmentation.

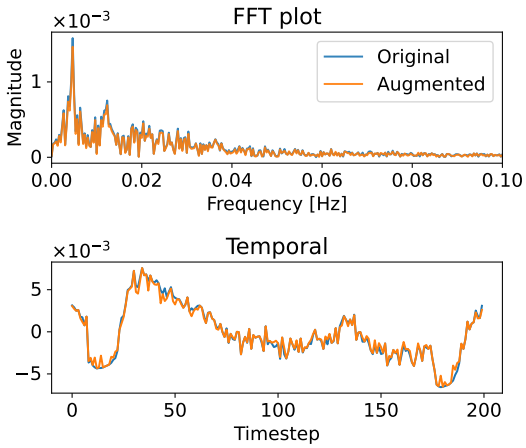

Figure 4: The upper plot shows the original signal and the augmented signal in the frequency domain. The augmentation selectively reduces the magnitude of specific frequency components. The impact of frequency suppression in the time domain is shown in the lower plot.

3.2. Adversarial Training Approach

To optimize both the LEAVES augmentation module and the encoder, we employ an adversarial training strategy. We use SimCLR as an example to illustrate this approach. We define the output of the encoder in representation learning as z. Within the framework, we consider N pairs of representations, denoted as (z_i, z_j), $\{i, j\} \in [1, N]$. The objective is to maximize the agreement between these pairs of views, and the corresponding loss function is defined as:

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)], \tag{9}$$

$$\ell_{i,j} = -\log \frac{\exp(s(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp(s(z_i, z_k)/\tau)} \tag{10}$$

The cosine similarity between z_i and z_j is represented by $s(z_i, z_j)$, and $1_{k \neq i}$ is an indicator function that is equal to 1 when $k \neq i$. The temperature parameter τ is set to 0.05 in this study. As illustrated in Figure 1, the LEAVES and encoder are optimized in contrary directions. The encoder attempts to minimize $\mathcal{L}$, while LEAVES attempts to maximize it. During training, the gradients for the encoder parameters (θ_E) are computed to minimize the contrastive loss L (Equation 9), while the gradients for the LEAVES module parameters (θ_L, which include σs and the pre-sigmoid inputs α_k for λ_ks) are computed to maximize the same loss L. Maximizing L for LEAVES is equivalent to minimizing $-L$ from the perspective of the LEAVES module's parameters. This adversarial process forces the encoder to learn robust representations that capture the underlying signal while being robust to the diverse transformations introduced by LEAVES. After training the SimCLR framework, the learned model weights in the encoder structure are used to initialize the model weights for supervised

learning in downstream tasks. The LEAVES framework can also be integrated into other frameworks (e.g., BYOL) for generating effective augmented views.

4. Evaluation

In this section, we evaluate the effectiveness of our proposed approach using various public time-series biobehavioral datasets. These include Apnea-ECG (Penzel et al., 2000) for sleep apnea detection, Sleep-EDFE (Kemp et al., 2018) for sleep stages classification, PTB-XL (Wagner et al., 2020) for arrhythmias detection, and PAMAP2 (Reiss and Stricker, 2012) for human activity recognition.

To benchmark our approach, we plug LEAVES into two well-known contrastive learning frameworks, SimCLR (Chen et al., 2020a) and BYOL (Grill et al., 2020), and compare its performance to that of other popular time-series self-supervised approaches. Also, a supervised 1D ResNet-18 model serves as a reference for the performance achievable without pre-training. The following sub-sections introduce the experimental settings and the results of application domains.

4.1. Experimental Settings

Datasets: We evaluate LEAVES on four diverse public biobehavioral datasets with each addressing distinct tasks.

- **Apnea-ECG (Sleep Apnea Detection) (Penzel et al., 2000):** We utilize the Apnea-ECG dataset to explore the correlation between sleep apnea diagnosis and cardiac activities captured by ECG signals, which are accessible through Physionet (Goldberger et al., 2000). We follow the same settings as the original data (Penzel et al., 2000), which utilize a 100Hz ECG one-minute scale to detect the occurrence of apnea. The training set comprises 17233 samples and the testing set had 17010 samples.

- **Sleep-EDFE (Sleep Stage Classification) (Kemp et al., 2018):** Electroencephalography (EEG) plays a crucial role in monitoring human brain activities. To evaluate our methods, we conduct evaluations using the multimodal Sleep-EDF (expanded) dataset, which comprises whole-night sleep recordings of 100Hz Fpz-Cz EEG and electrooculography (EOG) signals. Following (Supratak and Guo, 2020), we extract 42308 30-second samples that are annotated in 5 sleep stages. During the pre-training, we apply two LEAVES modules for EEG and EOG individually to optimize the corresponding views for each modality. Although 10 or 20-fold cross-validations with various settings are commonly used in most prior studies (Perslev et al., 2019; Mousavi et al., 2019; Perslev et al., 2021; Phan et al., 2022), we divide the validation sets based on the subject IDs to avoid information leakage during training.

- **PTB-XL (Arrhythmia Diagnosis) (Wagner et al., 2020)**: Cardiac arrhythmias are a major contributor to the prevalence of cardiovascular diseases, thus necessitating the development of precise and dependable detection methods for clinical use. We employ the PTB-XL dataset (Wagner et al., 2020), which consists of 21,837 12-lead, 10-second ECG recordings at a 100 Hz frequency, is divided into five categories of

arrhythmias. We follow the split of train, evaluation, and test guidelines outlined in the original publication Wagner et al. (2020), and report the results on the test set.

- **PAMAP2 (Human Activity Recognition) (Reiss and Stricker, 2012):** The PAMAP2 research demonstrates the ability to detect human activities through data collected from biobehavioral sensors. This dataset includes 3 inertial measurement units (IMU) and a heart rate monitor, which are sampled at 100 Hz and upsampled heart rate data. 12 of the 18 physical activities are used in the experiments, as suggested by (Moya Rueda et al., 2018; Tamkin et al., 2020).

Baselines: We benchmark LEAVES against a range of established approaches to comprehensively assess its effectiveness in representation learning for time-series biobehavioral data:

- **Supervised ResNet-18 (He et al., 2016):** We adapt the supervised 1D ResNet-18 architecture as a reference for the performance achievable without pre-training. The encoder output is passed to a 2-layer MLP: 1025, 512, and C, where C is the number of task classes. Each linear layer is followed by BatchNorm, ReLU, and Dropout rate of 0.2. Notably, we also use the ResNet-18 architecture as the backbone for our contrastive learning models.

- **Contrastive Learning Models:** We compare LEAVES with SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), MoCo-v3 (Chen et al., 2021), and VICReg (Bardes et al., 2021). These methods learn data representations by contrasting similar and dissimilar augmented views of the same sample. To ensure a fair comparison, these methods are implemented based on the same augmentation set as the proposed LEAVES. We further explore the challenges of hyperparameter tuning of augmentations in contrastive learning by exploring the σs of specific augmentation techniques within SimCLR and BYOL.

- **Other SSL methods:** We include TS2Vec (Yue et al., 2022) and TS-TCC (Eldele et al., 2021b) in our evaluation. These methods are specifically designed for time-series data and achieve SOTA performances in corresponding domains.

4.2. Results

Table 1 summarizes the performance in terms of macro F1-score (F1) and Area Under the Receiver Operating Characteristic curve (AUROC) , with red bold indicating the best performance per dataset and metric, and blue underline highlights the second-best. The proposed LEAVES method is compared with BYOL, SimCLR, other time-series SSL methods (TS2Vec, TS-TCC), and a supervised ResNet-18-1D model.

Notably, LEAVES-enhanced models outperform or are highly competitive with all other methods. On the Apnea-ECG, Sleep-EDFE, and PTB-XL datasets, LEAVES (BYOL) achieves the highest AUROC. On the PAMAP2 dataset, LEAVES (SimCLR) secures the top performance in both F1-score and AUROC. This demonstrates the effectiveness of LEAVES in learning robust representations for diverse biobehavioral signals and downstream tasks.

To validate the significance of the observed improvements, we conducted independent t-tests comparing the AUROC of both LEAVES(SimCLR) and LEAVES(BYOL) against key

Table 1: Performance comparison using macro F1-score and AUROC. Results are shown as mean ± standard deviation. The **best** result in each column is bolded. Statistical tests for the AUROC of LEAVES variants are denoted by: *vs. Supervised, †vs. best manually-tuned corresponding baseline (e.g., LEAVES(SimCLR) vs. best SimCLR$_\sigma$). The tuning results of SimCLR and BYOL are also included in this table, with sub-optimal results grayed out.

	Apnea-ECG		Sleep-EDFE		PTB-XL		PAMAP2	
Methods	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC
Supervised	$.755_{\pm.008}$	$.793_{\pm.010}$	$.773_{\pm.007}$	$.895_{\pm.005}$	$.665_{\pm.012}$	$.858_{\pm.010}$	$.908_{\pm.005}$	$.941_{\pm.004}$
ViewMaker	$.760_{\pm.011}$	$.801_{\pm.013}$	$.770_{\pm.008}$	$.888_{\pm.006}$	$.670_{\pm.011}$	$.864_{\pm.011}$	$.913_{\pm.006}$	$.944_{\pm.005}$
TS2Vec	$.765_{\pm.008}$	$.806_{\pm.010}$	$.777_{\pm.006}$	$.902_{\pm.005}$	$.669_{\pm.012}$	$.867_{\pm.010}$	$\mathbf{.928}_{\pm.005}$	$.955_{\pm.004}$
TS-TCC	$.749_{\pm.011}$	$.788_{\pm.009}$	$.762_{\pm.009}$	$.885_{\pm.008}$	$.672_{\pm.010}$	$.870_{\pm.009}$	$.921_{\pm.007}$	$.944_{\pm.005}$
VICReg	$.758_{\pm.010}$	$.800_{\pm.007}$	$.782_{\pm.005}$	$.906_{\pm.006}$	$.668_{\pm.013}$	$.865_{\pm.011}$	$.915_{\pm.006}$	$.947_{\pm.006}$
MoCo-v3	$.763_{\pm.009}$	$.808_{\pm.009}$	$.774_{\pm.006}$	$.898_{\pm.004}$	$.665_{\pm.012}$	$.860_{\pm.014}$	$.911_{\pm.005}$	$.942_{\pm.004}$
SimCLR$_{\sigma=.01}$	$.758_{\pm.009}$	$.795_{\pm.010}$	$.776_{\pm.006}$	$.901_{\pm.005}$	$.667_{\pm.010}$	$.861_{\pm.012}$	$.906_{\pm.004}$	$.938_{\pm.006}$
SimCLR$_{\sigma=.02}$	$.762_{\pm.008}$	$.804_{\pm.011}$	$.780_{\pm.007}$	$.908_{\pm.007}$	$.672_{\pm.009}$	$.869_{\pm.011}$	$.913_{\pm.005}$	$.946_{\pm.005}$
SimCLR$_{\sigma=.03}$	$.768_{\pm.010}$	$.811_{\pm.010}$	$.783_{\pm.006}$	$.910_{\pm.007}$	$.672_{\pm.011}$	$.871_{\pm.012}$	$.916_{\pm.006}$	$.948_{\pm.005}$
SimCLR$_{\sigma=.04}$	$.767_{\pm.010}$	$.809_{\pm.009}$	$.781_{\pm.007}$	$.910_{\pm.008}$	$.670_{\pm.013}$	$.868_{\pm.012}$	$.918_{\pm.006}$	$.951_{\pm.004}$
SimCLR$_{\sigma=.05}$	$.767_{\pm.008}$	$.810_{\pm.008}$	$.784_{\pm.006}$	$.908_{\pm.007}$	$.669_{\pm.011}$	$.866_{\pm.013}$	$.915_{\pm.007}$	$.950_{\pm.005}$
BYOL$_{\sigma=.01}$	$.766_{\pm.007}$	$.808_{\pm.009}$	$.774_{\pm.008}$	$.897_{\pm.009}$	$.666_{\pm.010}$	$.864_{\pm.011}$	$.917_{\pm.004}$	$.950_{\pm.004}$
BYOL$_{\sigma=.02}$	$.771_{\pm.010}$	$.814_{\pm.007}$	$.777_{\pm.006}$	$.903_{\pm.007}$	$.670_{\pm.009}$	$.868_{\pm.010}$	$.921_{\pm.004}$	$.952_{\pm.005}$
BYOL$_{\sigma=.03}$	$.775_{\pm.009}$	$.818_{\pm.008}$	$.779_{\pm.006}$	$.906_{\pm.006}$	$.671_{\pm.010}$	$.869_{\pm.010}$	$.924_{\pm.005}$	$.955_{\pm.004}$
BYOL$_{\sigma=.04}$	$.777_{\pm.009}$	$.820_{\pm.007}$	$.780_{\pm.005}$	$.905_{\pm.005}$	$.674_{\pm.011}$	$.873_{\pm.010}$	$.922_{\pm.006}$	$.954_{\pm.005}$
BYOL$_{\sigma=.05}$	$.776_{\pm.011}$	$.819_{\pm.009}$	$.776_{\pm.010}$	$.902_{\pm.009}$	$.671_{\pm.008}$	$.871_{\pm.009}$	$.923_{\pm.006}$	$.954_{\pm.004}$
LEAVES$_{(SimCLR)}$	$.775^{*}_{\pm.009}$	$.822^{*\dagger}_{\pm.008}$	$\mathbf{.790}^{*\dagger}_{\pm.007}$	$\mathbf{.914}^{*\dagger}_{\pm.006}$	$.677^{*}_{\pm.009}$	$.875^{*\dagger}_{\pm.008}$	$.921^{*}_{\pm.005}$	$.950_{\pm.005}$
LEAVES$_{(BYOL)}$	$\mathbf{.784}^{*\dagger}_{\pm.008}$	$\mathbf{.830}^{*\dagger}_{\pm.006}$	$.787^{*\dagger}_{\pm.005}$	$.912^{*\dagger}_{\pm.005}$	$\mathbf{.680}^{*\dagger}_{\pm.011}$	$\mathbf{.877}^{*\dagger}_{\pm.011}$	$.926^{*}_{\pm.003}$	$\mathbf{.957}^{*\dagger}_{\pm.004}$

baselines for each dataset. The analysis confirms the robust performance of our method. On the Apnea-ECG, Sleep-EDFE, and PTB-XL datasets, both LEAVES variants demonstrated statistically significant improvements over the Supervised baseline and their corresponding best manually-tuned model. For the PAMAP2 dataset, LEAVES(BYOL) also achieved a statistically significant improvement over the Supervised and best manually-tuned BYOL baselines ($p < .05$), though its advantage over TS2Vec was not statistically significant. These results confirm that the performance gains from the LEAVES framework are not only numerically superior but also statistically meaningful across diverse biobehavioral signals and tasks.

Our experiments highlight the practical challenges of manual hyperparameter tuning for data augmentations. As shown by the grayed out entries in Table 1, empirically searching for the optimal augmentation strength (σ) for SimCLR and BYOL often leads to suboptimal outcomes. This underscores the difficulty of tuning augmentations to diverse time-series datasets and tasks. In contrast, LEAVES automates this process, learning suitable augmentation policies without requiring extensive hyperparameter searches, and and therefore simplifies the application of contrastive learning while delivering superior performance.

4.3. Learning Hyperparameters for Augmentations

In the LEAVES framework, the scalar hyperparameters that control differentiable augmentations are dynamically optimized across training epochs. This adaptability enables LEAVES to adjust augmentation strategies to the unique characteristics and challenges of each dataset. The dynamics of these hyperparameters, recorded at the end of each training

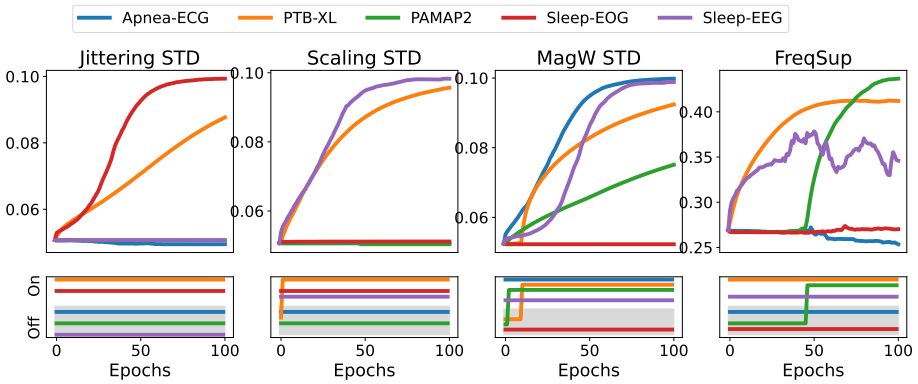

Figure 5: Visualization of the scalar hyperparameters, including intensity (σ) and gating status (λ) during LEAVES (SimCLR) training. The upper row indicates the intensity change of each augmentation; whereas the lower row shows the gating "on" or "off" status of augmentations.

epoch, are shown in Figure 5. We observe the changes in intensity with the on status of the gating parameter.

For *Jittering*, *Scaling*, *MagW*, and *FreqSup*, the *sigma* values exhibit an increasing trend, which indicates that the augmentations become progressively more aggressive. However, these values do not converge to a single threshold across datasets; rather, they vary across datasets, which reflects the adaptive nature of the framework to different signals. Similarly, the selection of augmentations based on λ values shows LEAVES's ability to adapt various augmentation policies to different signals during training.

The learned parameters of augmentations may reflect the various characteristics of different biobehavioral signals. For Apnea-ECG, *MagW* and *TimeM* are consistently activated, which suggests that altering temporal dynamics and magnitude relationships in the ECG waveform is more informative for 1-lead ECG. In contrast, the 12-lead ECG data in PTB-XL benefits from all augmentations including jittering and scaling that are not activated for Apnea ECG, potentially due to the higher dimensionality and complexity of the 12-lead signal. For PAMAP2 (IMU), the activation of *MagW*, *TimeM*, and *FreqSup* indicate that both temporal and frequency information are essential in learning representation from IMU signals. The initial deactivation of *FreqSup* suggests that LEAVES prioritizes learning temporal patterns before gradually incorporating spectral augmentation into the representations. *Jittering* and *Scaling* are not activated while pretraining on IMU data. This is potentially because jittering and scaling noise are already present in the IMU data, even without augmentation (Nirmal et al., 2016). For Sleep-EDFE, only jittering is activated for EOG while being consistently deactivated for EEG. This suggests that EEG signals are more sensitive to noise perturbations compared to EOG signals. These observations highlight the importance of selecting the proper augmentations based on the specific waveform patterns of the data and task.

5. Discussion

In this section, various ablation studies are presented. Additionally, we discuss the learned augmentation policy and computational complexity of the proposed LEAVES module.

Table 2: Performance drops of removing specific augmentation methods from LEAVES. The magnitude-based methods (*Mag*) combine the augmentations of *Jittering*, *Scaling*, and *MagW*. The results are based on the Macro-F1 scores (%) from the main experimental results.

Dataset	*Mag (M)*	*TimeM (T)*	*FreqSup (F)*	*M & T*	*M & F*	*T & F*
Apnea-ECG	2.1 ↓	1.3 ↓	0.1 ↓	2.8 ↓	2.4 ↓	1.3 ↓
Sleep-EDFE	1.5 ↓	1.7 ↓	0.8 ↓	2.4 ↓	2.0 ↓	2.0 ↓
PTB-XL	1.4 ↓	0.3 ↓	0.6 ↓	1.5 ↓	1.6 ↓	0.7 ↓
PAMAP2	1.2 ↓	2.2 ↓	1.5 ↓	3.0 ↓	1.9 ↓	2.3 ↓

5.1. Ablation Study: Impact of Different Types of Data Augmentations on LEAVES Performances

In this study, we investigate the impact of the data augmentation methods. We consider the categories of augmentation methods as magnitude-based method (*Mag*), which encompasses adjustments to signal amplitude; temporal method (*TimeM*), which manipulates the temporal positions of signals; and the frequency-based method (*FreqSup*), which modifies the signal from the frequency domain. We perform an ablation experiment using SimCLR-based LEAVES by systematically removing each category of augmentation method and evaluating the resulting model's performance on various downstream tasks. The results are summarized in Table 2, which showcases the performance drop observed when excluding a specific augmentation method from LEAVES. The table presents results for all four biobehavioral datasets using the drops in macro-F1 score as the performance metric. The results consistently demonstrate that removing any single augmentation technique leads to a decline in performance across most datasets. This indicates that each augmentation method plays a valuable role in enhancing the model's ability to learn informative representations. Notably, both the proposed *FreqSup* and *TimeM* methods consistently contribute to improving performance. Furthermore, the combined effect of removing multiple augmentations often results in more substantial performance drops compared to removing single ones. This suggests that the combinations of different types of augmentations provide more comprehensive and diverse data transformations.

5.2. Comparison of LEAVES with ViewMaker for View Generation

Our approach is inspired by ViewMaker (Tamkin et al., 2020), but there are some key differences. Instead of using "black-box" deep networks to create spatial distortions, our LEAVES module uses time-series domain knowledge-based augmentations to generate views. This makes LEAVES lightweight and can produce more diverse yet reliable views than ViewMaker. Figure 6 shows an example of ViewMaker's limitations in temporal distortion and information preservation, as ViewMaker distorts most ECG fiducials. To assess the quality of the data from the generated views, we use an ECG quality check method (Zhao and Zhang, 2018) with the NeuroKit package (Makowski et al., 2021). We find that ViewMaker corrupted almost half (49.5%) of the ECGs to be labeled as "Unacceptable," which indicates signals that are barely recognizable as ECG signals. In contrast, the views generated using our proposed method with both SimCLR and BYOL have acceptable data quality (96.0%

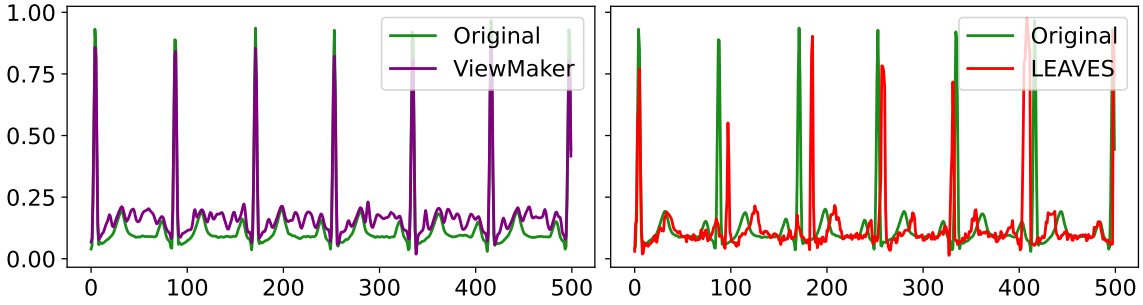

Figure 6: Visualization of the views learned using the ViewMaker and the proposed method

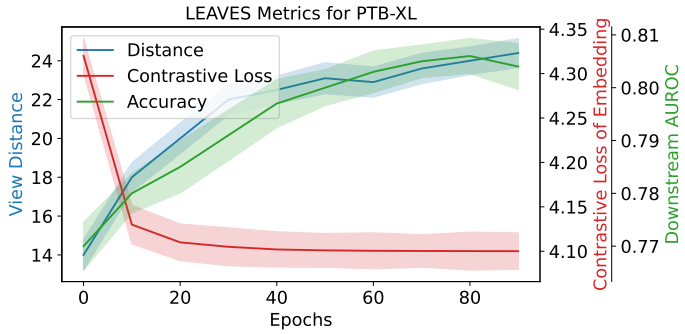

Figure 7: Learning metrics of LEAVES (SimCLR) in datasets of PTB-XL. The blue lines represent the distance between the pairs of augmented signals calculated with dynamic time warping (DTW); red indicates the contrastive loss of the learning encoders; and green indicates the AUROC with frozen encoder.

and 93.1%, respectively) compared to the original data. These limitations of ViewMaker when applied to time-series data motivate us to develop our proposed method in this study.

5.3. Adaptive Learning and Augmentation Strategies in LEAVES

To further our understanding of the adaptive learning processes within the LEAVES framework, we analyze several key performance metrics throughout the training period. These metrics include the distance between paired views measured by dynamic time warping (DTW), contrastive loss, and AUROC in downstream tasks assessed using a linear probe with frozen encoders, as shown in Figure 7.

The analysis reveals an increase in the distance between paired views, which suggests that the framework generates increasingly challenging views over time. Moreover, the improvement in AUROC on downstream tasks as training progresses demonstrates that the representations learned are increasingly effective. These results highlight the effectiveness of the LEAVES framework in achieving the dual objectives of contrastive learning: diversifying augmented views and enhancing encoder performance for varied applications.

5.4. Space & Time Complexities

The proposed method, LEAVES, has demonstrated significant advantages in both model space and time complexity when compared to previous state-of-the-art methods such as ViewMaker. Notably, LEAVES optimizes approximately 20 parameters for view generation, in contrast to the roughly 580K parameters required by the 1D ViewMaker structure. This substantial reduction in parameter count not only minimizes memory overhead but also contributes to faster convergence during training.

In addition, the integration of LEAVES into contrastive learning frameworks yields considerable runtime improvements. For instance, on the Sleep-EDFE dataset—using an *AWS p3.2xlarge* instance equipped with an NVIDIA V100 GPU—the baseline SimCLR framework requires an average of 578.0 seconds per epoch (for 100 training epochs with a batch size of 128). When LEAVES is incorporated, the per-epoch time is reduced to 390.8 seconds. This efficiency gain is primarily attributed to the design of LEAVES, which integrates augmentations directly into the network. By doing so, it leverages GPU acceleration to perform complex augmentation operations concurrently with other network computations, thereby eliminating the need for separate data pre-processing and external augmentation modules that typically add computational overhead.

5.5. Limitation

Although we substantially improve the interpretability of the data augmentation process in contrastive learning by leveraging the pre-defined data transforming methods, challenges remain. The the optimization process is not easily understandable and makes it difficult for researchers to identify the best data augmentation policies. Therefore, enhancing interpretability further would be beneficial to facilitate future research.

6. Conclusion

In this work, we introduce a simple but effective LEAVES module to learn augmentations for time-series data in contrastive learning. With an adversarial training approach, our proposed method optimizes the hyperparameters for data augmentation methods in contrastive learning. We evaluate the proposed method on four datasets, and find that it outperforms the baselines. Our evaluation across four diverse datasets demonstrates that LEAVES, as a computationally efficient solution, typically outperforms or closely matches existing baselines while avoiding extensive manual tuning. We also demonstrate that LEAVES can preserve the original information in augmented views, especially on ECG time-series data, compared to the prior methods.

Although LEAVES achieves promising results in learning views for contrastive learning, there are still limitations. For example, the interpretability of the view-tuning process can be improved. Thus, future work will include enhancing interpretability to better understand data augmentation policies in contrastive learning. Also, we will expand more augmentation methods in LEAVES and apply LEAVES to a wider range of time-series data.

Acknowledgments

This work was supported by National Science Foundation (# 2047296 and # 1840167) and National Institute of Health (# R01DA059925)

References

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Hung-Yu Chang, Cheng-Yu Yeh, Chung-Te Lee, and Chun-Cheng Lin. A sleep apnea detection system based on a one-dimensional deep convolution neural network model using single-lead electrocardiogram. *Sensors*, 20(15):4157, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021a.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2352–2359, 2021b.

Hengyang Fang, Changhua Lu, Feng Hong, Weiwei Jiang, and Tao Wang. Sleep apnea detection based on multi-scale residual network. *Life*, 12(1):119, 2022.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Bryan Gopal, Ryan Han, Gautham Raghupathi, Andrew Ng, Geoff Tison, and Pranav Rajpurkar. 3kg: Contrastive learning of 12-lead electrocardiograms using physiologically-inspired augmentations. In *Machine Learning for Health*, pages 156–167. PMLR, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Philipp Hallgarten, David Bethge, Ozan Özdcnizci, Tobias Grosse-Puppendahl, and Enkelejda Kasneci. Ts-moco: Time-series momentum contrast for self-supervised physiological representation learning. In *2023 31st European Signal Processing Conference (EUSIPCO)*, pages 1030–1034. IEEE, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019.

Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2021.

Bob Kemp, A Zwinderman, B Tuk, H Kamphuisen, and J Oberyé. Sleep-edf database expanded. 2018.

Xiaoyu Li, Chen Li, Yuhua Wei, Yuyao Sun, Jishang Wei, Xiang Li, and Buyue Qian. Bat: Beat-aligned transformer for electrocardiogram classification. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 320–329. IEEE, 2021.

Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada: differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019.

Aoming Liu, Zehao Huang, Zhiwu Huang, and Naiyan Wang. Direct differentiable augmentation search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12219–12228, 2021.

Dongsheng Luo, Wei Cheng, Yingheng Wang, Dongkuan Xu, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Yanchi Liu, Yuncong Chen, Haifeng Chen, et al. Time series contrastive learning with information-aware augmentations. *arXiv preprint arXiv:2303.11911*, 2023.

Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. Neurokit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, Feb 2021. ISSN 1554-3528. doi: 10.3758/s13428-020-01516-y. URL https://doi.org/10.3758/s13428-020-01516-y.

Temesgen Mehari and Nils Strodthoff. Self-supervised representation learning from 12-lead ecg data. *Computers in Biology and Medicine*, 141:105114, 2022.

Sajad Mousavi, Fatemeh Afghah, and U Rajendra Acharya. Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach. *PloS one*, 14 (5):e0216456, 2019.

Fernando Moya Rueda, René Grzeszick, Gernot A Fink, Sascha Feldhorst, and Michael Ten Hompel. Convolutional neural networks for human activity recognition using body-worn sensors. In *Informatics*, volume 5, page 26. Multidisciplinary Digital Publishing Institute, 2018.

K Nirmal, AG Sreejith, Joice Mathew, Mayuresh Sarpotdar, Ambily Suresh, Ajin Prakash, Margarita Safonova, and Jayant Murthy. Noise modeling and analysis of an imu-based attitude sensor: improvement of performance by filtering and sensor fusion. In *Advances in optical and mechanical technologies for telescopes and instrumentation II*, volume 9912, pages 2138–2147. SPIE, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Thomas Penzel, George B Moody, Roger G Mark, Ary L Goldberger, and J Hermann Peter. The apnea-ecg database. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 255–258. IEEE, 2000.

Mathias Perslev, Michael Jensen, Sune Darkner, Poul Jørgen Jennum, and Christian Igel. U-time: A fully convolutional network for time series segmentation applied to sleep staging. *Advances in Neural Information Processing Systems*, 32, 2019.

Mathias Perslev, Sune Darkner, Lykke Kempfner, Miki Nikolic, Poul Jørgen Jennum, and Christian Igel. U-sleep: resilient high-frequency sleep staging. *NPJ digital medicine*, 4 (1):1–12, 2021.

Huy Phan, Kaare Mikkelsen, Oliver Y Chén, Philipp Koch, Alfred Mertins, and Maarten De Vos. Sleeptransformer: Automatic sleep staging with interpretability and uncertainty quantification. *IEEE Transactions on Biomedical Engineering*, 69(8):2456–2467, 2022.

Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pages 8703–8714. PMLR, 2021.

Aniruddh Raghu, Divya Shanmugam, Eugene Pomerantsev, John Guttag, and Collin M Stultz. Data augmentation for electrocardiograms. In *Conference on Health, Inference, and Learning*, pages 282–310. PMLR, 2022.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pages 108–109. IEEE, 2012.

Evgenia Rusak, Lukas Schott, Roland Zimmermann, Julian Bitterwolfb, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. Increasing the robustness of dnns against im-age corruptions by playing the game of noise. 2020.

Pritam Sarkar and Ali Etemad. Self-supervised ecg representation learning for emotion recognition. *IEEE Transactions on Affective Computing*, 13(3):1541–1554, 2020.

Kevalee Shah, Dimitris Spathis, Chi Ian Tang, and Cecilia Mascolo. Evaluating contrastive learning on wearable timeseries for downstream clinical outcomes. *arXiv preprint arXiv:2111.07089*, 2021.

Sinam Ajitkumar Singh and Swanirbhar Majumder. A novel approach osa detection using single-lead ecg scalogram based on deep neural network. *Journal of Mechanics in Medicine and Biology*, 19(04):1950026, 2019.

Sandra Śmigiel, Krzysztof Pałczyński, and Damian Ledziński. Ecg signal classification using deep learning techniques based on the ptb-xl dataset. *Entropy*, 23(9):1121, 2021.

Akara Supratak and Yike Guo. Tinysleepnet: An efficient deep learning model for sleep stage scoring based on raw single-channel eeg. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 641–644. IEEE, 2020.

Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. *arXiv preprint arXiv:2010.07432*, 2020.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, pages 216–220, 2017.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):1–15, 2020.

Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Kristoffer Wickstrøm, Michael Kampffmeyer, Karl Øyvind Mikalsen, and Robert Jenssen. Mixing up contrastive learning: Self-supervised representation learning for time series. *Pattern Recognition Letters*, 155:54–61, 2022.

Huiyuan Yang, Han Yu, and Akane Sano. Empirical evaluation of data augmentations for biobehavioral time series data with deep learning. *arXiv preprint arXiv:2210.06701*, 2022.

Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *arXiv preprint arXiv:2206.02909*, 2022.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

Jing Zhang, Deng Liang, Aiping Liu, Min Gao, Xiang Chen, Xu Zhang, and Xun Chen. Mlbf-net: a multi-lead-branch fusion network for multi-class arrhythmia classification using 12-lead ecg. *IEEE Journal of Translational Engineering in Health and Medicine*, 9:1–11, 2021.

Junbo Zhang and Kaisheng Ma. Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16650–16659, 2022.

Zhidong Zhao and Yefei Zhang. Sqi quality evaluation mechanism of single-lead ecg signal based on simple heuristic fusion and fuzzy comprehensive evaluation. *Frontiers in physiology*, 9:727, 2018.

Table 3: Performance impact of different augmentation method orders in LEAVES with SimCLR on downstream tasks. Macro-F1 score is used as the evaluation metric.

Dataset	Random Order	Fixed Order
Apnea-ECG	0.779 ± 0.011	0.781 ± 0.009
Sleep-EDFE	0.796 ± 0.014	0.791 ± 0.013
PTB-XL	0.680 ± 0.010	0.678 ± 0.012
PAMAP2	0.931 ± 0.009	0.934 ± 0.008

Appendix A. Additional Abalation Study

A.1. Ablation Study: Order of Augmentation Methods

Does the order of augmentations matter? To answer this question, we conducted an ablation study to explore the effect of the order of the augmentations. For the four applications evaluated in Section 4, we randomly shuffle the order of the augmentation methods applied and evaluate the performance in the downstream tasks. As control groups, we repeat the pre-training experiments with fixed augmentation orders but with shuffled random seeds. All ablation experiments are performed with the combination of LEAVES and SimCLR. Table 3 shows the results of shuffling the order of augmentation methods. From the table, we can see that the performance variations of the shuffled order and the fixed order follow a similar pattern and that no statistically significant differences (unpaired t-test, $p > 0.05$) are observed. Therefore, we can conclude that the order of the different augmentation methods applied in the LEAVES module does not significantly affect its performance.

A.2. Ablation Study: Robustness Evaluation

We further conduct an ablation study to examine the robustness of augmentation-based contrastive learning methods under noise corruption. In this experiment, we perform supervised learning on top of the pre-trained backbones using the original and LEAVES-integrated SimCLR and BYOL. Differing from the evaluation setting in Section 4, we fine-tune the models with clean training sets. However, we evaluate the model performance on the corrupted test set with randomly generated noises from *Jittering, Scaling, MagW*, and *TimeM* with randomized σ between 0.01 and 0.05. The corruption applied to the test set thus introduces distribution shifts between the original and augmented datasets. Table 4 shows the results of the robustness evaluation. We conduct paired t-tests to compare the *Diff* between clean and corrupted samples in terms of robustness. We compare the results from 4 datasets of supervised learning to those from SimCLR and BYOL, respectively. Additionally, we conduct paired t-tests between SimCLR and BYOL, and between SimCLR (LEAVES) and BYOL (LEAVES). The test results show that all the p-values were below 0.05. However, when we compare SimCLR to BYOL and SimCLR (LEAVES) to BYOL (LEAVES), no significance can be observed ($p > 0.05$). From the table and the results of statistical tests, we observe that applying both SimCLR and BYOL pre-training methods improved the model's robustness to corrupted noises. Moreover, by integrating the LEAVES module, the negative impact of introducing noise into the test set is further alleviated. On the other

Table 4: The macro-F1 performances table of applying pre-trained backbone on the clean and corrupted test sets. Diff. representations of the performance gap between clean and corrupted sets. The bold values indicate the smallest Diff. in performance between the clean and corrupted test sets.

2*Method	Apnea-ECG			Sleep-EDF		
	Clean	Corrupted	Diff	Clean	Corrupted	Diff
Supervised	0.757	0.671	0.086	0.769	0.688	0.081
SimCLR	0.775	0.705	0.070	0.783	0.729	0.054
BYOL	0.779	0.712	0.067	0.780	0.724	0.056
SimCLR(LEAVES)	0.788	0.725	0.063	0.790	0.740	0.050
BYOL(LEAVES)	0.795	0.733	**0.062**	0.785	0.738	**0.047**

2*Method	PAMAP2			PTB-XL		
	Clean	Corrupted	Diff	Clean	Corrupted	Diff
Supervised	0.896	0.812	0.084	0.662	0.615	0.047
SimCLR	0.925	0.867	0.058	0.671	0.619	0.052
BYOL	0.931	0.868	0.063	0.671	0.621	0.050
SimCLR(LEAVES)	0.934	0.883	**0.051**	0.669	0.632	**0.037**
BYOL(LEAVES)	0.936	0.881	0.055	0.677	0.636	0.041

hand, the choice between SimCLR and BYOL does not affect the improvement in model robustness.

Appendix B. Implementation Details

B.1. Hyperparameter and Training Details

All models were implemented in PyTorch and trained on an NVIDIA V100 GPU. For both the self-supervised pre-training and the downstream supervised fine-tuning, we used the Adam optimizer. The learning rate was set to `1e-3` for pre-training and fine-tuned for each downstream task. A consistent batch size of 128 was used across all experiments. The temperature parameter (τ) for the SimCLR contrastive loss was set to 0.05.

The pre-training phase for all contrastive models was run for 100 epochs. For the downstream fine-tuning, the number of epochs varied by dataset to ensure convergence, as detailed in Table 5.

B.2. Downstream Task Evaluation: Full Finetuning

To clarify the evaluation process for the main results presented in Table 1, we employed a full fine-tuning protocol. After the self-supervised pre-training phase, the learned encoder weights were used to initialize a 1D ResNet-18 model. A linear classification head was then added on top of the encoder. Subsequently, the **entire network** (both the encoder and the new classifier head) was fine-tuned end-to-end on the labeled training data of each respective downstream task. This approach allows the model to adapt the learned representations

Hyperparameter	Apnea-ECG	Sleep-EDF	EPTB-XL	PAMAP2
Pre-training				
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	1e-3	1e-3	1e-3	1e-3
Batch Size	128	128	128	128
Epochs	100	100	100	100
Fine-tuning				
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	1e-4	1e-4	5e-5	5e-5
Batch Size	128	128	128	128
Epochs	50	50	80	80

Table 5: Hyperparameter settings for pre-training and downstream fine-tuning across all datasets.

specifically to the target task, which typically leads to superior performance compared to using a frozen encoder with a linear probe. The Supervised baseline was trained from a random initialization using the identical architecture and fine-tuning procedure for a fair comparison.

B.3. Code Availability

The source code and trained models for this research are publicly available on GitHub to ensure reproducibility and facilitate future research: https://github.com/comp-well-org/LEAVES.