# An Unsupervised Approach for Periodic Source Detection in Time Series

**Berken Utku Demirel** [1] **Christian Holz** [1]

## Abstract

Detection of periodic patterns of interest within noisy time series data plays a critical role in various tasks, spanning from health monitoring to behavior analysis. Existing learning techniques often rely on labels or clean versions of signals for detecting the periodicity, and those employing self-supervised learning methods are required to apply proper augmentations, which is already challenging for time series and can result in collapse—all representations collapse to a single point due to strong augmentations. In this work, we propose a novel method to detect the periodicity in time series without the need for any labels or requiring tailored positive or negative data generation mechanisms with specific augmentations. We mitigate the collapse issue by ensuring the learned representations retain information from the original samples without imposing any random variance constraints on the batch. Our experiments in three time series tasks against state-of-the-art learning methods show that the proposed approach consistently outperforms prior works, achieving performance improvements of more than 45–50%, showing its effectiveness.

Code: https://github.com/eth-siplab/Unsupervised_Periodicity_Detection

## 1. Introduction

Periodic structure in time series data holds significant importance in monitoring individuals' behaviors and health (Sandvik et al., 1995; Hogan, 2006). A wide range of tasks related to time series, including step counting (Master et al., 2022) and heart rate monitoring (Mandsager et al., 2018), can reveal crucial information about one's health situation. Moreover, on a daily basis, people generate substantial amounts of time series data from smartphones or wearable devices (Bayoumy et al., 2021), which can be used to provide insights into their behaviors and health (Huhn et al., 2022; Hughes et al., 2023). However, labeling these huge amounts of temporal data is challenging, expensive, and resource-intensive. For instance, if the desired task is to monitor the respiration or heart rate, collecting gold-standard synchronized signals for obtaining ground truth information from a medical sensor or video is not only time-consuming but also expensive (Yang et al., 2023; Kiyasseh et al., 2020). Moreover, obtaining ground truth is not feasible in daily life, i.e., non-controlled environments outside of the lab, due to the potential discomfort it may impose on individuals (Gao et al., 2023), which can be caused by privacy concerns, such as the use of video recording (Ryoo et al., 2017), or the use of bulky electrodes with gels, as in the case for medical sensors (Stracina et al., 2022).

The self-supervised learning (SSL) paradigm provides a solution to overcome this problem by exploiting unlabelled data to formulate pretext tasks such as predicting the rotation of images (Gidaris et al., 2018), or contexts (Doersch et al., 2015) to learn the invariant representations of samples (Jing et al., 2022) to the applied transformation. The effectiveness of representations on downstream tasks directly depends on the inter-sample semantic similarity relations (Balestriero & LeCun, 2022; Wang et al., 2022b) that are created through tailored data augmentation techniques to preserve semantics depending on the task (Tian et al., 2020; Chen et al., 2020) and input characteristics (Demirel & Holz, 2023). However, even when the downstream task is known, creating samples that keep the task-relevant information intact is more challenging in noisy quasi-periodic time series due to the complexity of the dynamical data generation mechanisms (West et al., 1999), where strong augmentations can cause model collapse (Jing et al., 2022) while weak augmentations can hinder model convergence (Tian et al., 2020).

Considering these limitations, in this work, we propose a novel method that regularizes the parametric learners to detect periodic components in time series data by punishing the randomness in a sequence while capturing as much information as possible about the periodicity without requiring any specialized data augmentation technique. Since our proposed method bypasses the intricate data augmentation process required for generating positive or negative samples

[1]Department of Computer Science, ETH Zurich, Switzerland. Correspondence to: Berken Utku Demirel <berken.demirel@inf.ethz.ch>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

to learn the relations or invariant representations between samples, we prevent the model collapse issue by relating the learned representations with original samples from which they originated, rather than decorrelating the representations with each other or imposing random variance constraints on the batch. Our approach argues that every learned representation should retain a remnant of the sample from which it originates. The contributions can be summarized as follows:

- We propose a novel set of regularizers that consider the generation mechanism of time series data to detect periodic patterns of interest in the presence of noise.
- We introduce a new approach to prevent collapse in unsupervised periodicity detection by relating learned representations with samples from which they originated, resulting in relaxed assumptions for batch statistics.
- We demonstrate that the proposed regularizers help the learners extract useful representations without requiring any supervision or specialized time series augmentations for generating positive/negative samples.

## 2. Method

### 2.1. Notations

We use lowercase symbols $(x)$ to denote scalar quantities, bold lowercase symbols $(\mathbf{x})$ for vectors, e.g., time series, and capital letters $(X)$ for random variables. The parametric functions is represented as $f_\theta(.)$ where $\theta$ is the parameters. The discrete Fourier transformation is denoted as $\mathcal{F}(.)$, yielding a complex variable as $\mathcal{F}(\mathbf{x}) \in \mathbb{C}$. The detailed calculations for each operation are given in the Appendix A.1.

### 2.2. Objective

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i)\}_{i=1}^K$ where each $\mathbf{x}_i$ consists of real-valued sequences with length L, $(x_1, x_2, \ldots, x_L)$, which are sampled uniformly. The objective is to train a learner $f_\theta : X \to \mathbb{R}^l$ which seeks to learn periodic representations of interest in the data such that when it is evaluated on another set $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$, the representations $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ will mainly contain desired periodic source, which can be detected i.e., $\underset{k}{\operatorname{argmax}} |\mathcal{F}(\mathbf{z}_i)_k| = y_i \in \mathbb{R}^+$.

Since our objective is to detect the periodic source of interest in the uniformly sampled time series data without using any labels, we proposed a set of regularizers designed to help the learners extract useful representations that capture the desired periodicity. In the following sections, we describe the proposed regularizers and outline their primary objectives. Our first proposed regularizer decreases the randomness of the overall time series sequence while maximizing the periodicity by minimizing the entropy of the spectra as follows.

#### 2.2.1. Maximizing the Periodicity

**Proposition 2.1** (Maximizing periodicity)**.** *Minimizing the spectral entropy of sequential data is nothing but maximizing its periodicity in the time domain.*

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{se} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{I}(\mathbf{y}; \mathbf{x}),$$

$$\textit{where}\;\; \mathcal{L}_{se} = -\sum_w S_{f_\theta}(\omega) \log S_{f_\theta}(\omega),$$

$$S_{f_\theta}(\omega) = \frac{S_{f_\theta}(\omega)}{\sum_\omega S_{f_\theta}(\omega)},\; S_{f_\theta}(\omega) = \left| \sum_n f_\theta(\mathbf{x})_n e^{-j\omega n} \right|^2$$

*Proof.* The spectral entropy is maximum (periodicity is minimum) when the sequence generation process is a Gaussian with zero mean, $\sigma^2$ variance, i.e., $W(t) \sim \mathcal{N}(0, \sigma_w^2)$.

$$\mathbb{E}[W(t)] = 0, \quad R_{WW}(\tau) = \mathbb{E}[W(t+\tau)W^*(t)] \tag{1}$$

$$R_{WW}(\tau) = \sigma^2 \delta(\tau) \xrightarrow{\mathcal{F}(.)} S_W(\omega) = \sigma_\omega^2 \tag{2}$$

$$H(S_W) \geq H(S_X), \forall \mathbf{x} \in X \tag{3}$$

The spectral entropy is minimum if the sequence generation process is periodic, i.e., $R_{PP}(\tau) = R_{PP}(\tau + T)$.

$$R_{PP}(\tau) = \sum_{n=-\infty}^{\infty} \alpha_n e^{j\omega_0 n\tau}, \;\; \omega_0 = \frac{2\pi}{T} \tag{4}$$

$$\alpha_n = \frac{1}{T} \int_{-T/2}^{T/2} R_{PP}(\tau) e^{-j\omega_0 n\tau} d\tau \tag{5}$$

$$S_P(w) = 2\pi \sum_{n-\infty}^{\infty} \alpha_n \delta(\omega - n\omega_0) \tag{6}$$

$$H(S_P) \leq H(S_X), \forall \mathbf{x} \in X \tag{7}$$

Therefore, $H(S_P) \leq \mathcal{L}_{se} \leq H(S_W)$. □

The last inequality completes the proof by showing that minimizing the spectral entropy of sequential data is equivalent

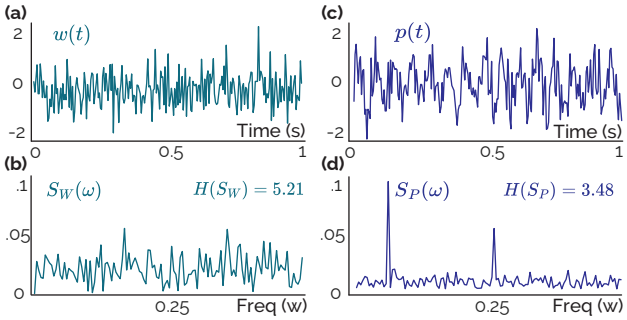



*Figure 1.* **(a)** A time sequence sample $w(t)$ from a Gaussian process **(b)** with its normalized spectra $S_W(\omega)$, **(c)** a periodic sequence in the presence of added random noise, denoted as $p(t)$ **(d)** the corresponding spectra of the sequence $p(t)$, denoted as $S_P(\omega)$.

to maximizing its periodicity. A detailed proof, with the bandlimited discrete case, can be found in the Appendix A. We also provide an intuitive demonstration of samples from data generation processes and corresponding spectra in Figure 1 where it can be observed that the periodic data has lower entropy compared to the sample from Gaussian distribution.

Although this regularizer can induce periodicity in sequences, it will cause the model to collapse to a single point and produce a constant output. The existing solutions for preventing collapse generally incorporate the notion of class balance across the dataset (Krause et al., 2010) or batch (Bardes et al., 2022; Speth et al., 2023). Such approaches, however, are overly optimistic since during SSL training, where there is no label information, this regularizer may punish the models even for learning correct representations. Alternatively, the models can learn task-unrelated information to introduce diversity in representations. Therefore, to prevent the degenerate solution, we have incorporated a secondary regularizer that ensures the extracted representations retain information from the original samples.

### 2.2.2. Preventing the Collapse

**Proposition 2.2** (Degenerate solution)**.** *The relative entropy between the spectral distributions of samples and extracted representations is upper-bounded by a degenerate solution.*

$$\begin{aligned} D_{KL}(\mathbb{P} \parallel \mathbb{Q}) > \mathcal{L}_{ds} \geq 0, \quad \textit{where} \quad \textit{Var}(\mathbb{Q}) = 0, \\ \textit{and} \quad \mathcal{L}_{ds} = \sum_{\omega} S_X(\omega) \log \frac{S_X(\omega)}{S_{f_\theta}(\omega)}, \\ S_X(\omega) = \frac{S_X(\omega)}{\sum_\omega S_X(\omega)}, \; S_X(\omega) = \left| \sum_n x_n e^{-j\omega n} \right|^2 \end{aligned} \tag{8}$$

*Proof.* There exists a sample with a spectrum $S_X(\omega)$ for which a collapsed learner, trained to maximize periodicity (minimize spectral entropy), results in an infinite divergence of spectral distributions between samples and embeddings.

$$\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{se} \implies \lim_{\mathcal{L}_{se} \to 0} S_{f_\theta}(\omega) \to 0, \; \exists \omega \in \Omega \tag{9}$$

$$\exists \omega \in (-\infty, \infty), \quad S_{f_\theta}(\omega) = 0 \wedge S_X(\omega) \neq 0, \tag{10}$$

yields $D_{KL}(\mathbb{P} \parallel \mathbb{Q}) \to \infty, \quad S_X(\omega) \sim \mathbb{P}, S_{f_\theta}(\omega) \sim \mathbb{Q}$ □

We provide an intuitive illustration of the proposition with a sample in Figure 2 and the detailed proof in Appendix A.

The proposition 2.2 suggests that degenerate solutions can be prevented if the learner seeks to minimize the relative entropy between the spectra of samples and extracted representations. In other words, this proposed regularization technique helps the learner to generate diverse representations according to the input distribution, preventing the collapse where inputs are mapped to the same vector, while avoiding the imposition of random variances within a batch.

The usage of these two regularizers together also eliminates the requirement for generating positive or negative samples to construct embedding space with specialized augmentations which is known to be particularly challenging for non-stationary time series data. Moreover, most of the unsupervised learning methods prevent model collapse by repelling the negative pairs which are randomly chosen from the batch (Chen et al., 2020), selected from memory banks according to the learned relevance (He et al., 2019), or enforcing distinct embeddings within a batch (Bardes et al., 2022). However, since there is no label information during training, these methods can repel/decorrelate representations of samples from the same category which can lead to suboptimal embedding space. Therefore, to mitigate this problem, we take a different approach and govern the representation space through two opposing regularizers, seeking to find the periodicity of interest in the noisy time sequence data without causing significant distortions to the features.

When we calculate these two loss functions, we specify a bandlimited frequency range of interest $f^*$ instead of considering the entire continuous spectrum while minimizing the overall power in frequencies outside of this defined range $f^{'}$. Minimizing frequencies outside of the specified range, essentially applying a bandpass filter, is a common approach in the literature (Speth et al., 2023; Gideon & Stent, 2021). We observe marginal improvements with this addition while the improvement is diminished with better-designed filters. The overall loss for training models is given in Equation 11.

$$\begin{aligned} \mathcal{L} = -\sum_{f^*} S(\omega) \log S(\omega) + \sum_{f^*} S_X(\omega) \log \frac{S_X(\omega)}{S(\omega)} \\ + \sum_{f^{'}} S(\omega), \quad \omega = 2\pi f \text{ and } f^{'} = \mathbb{U} \setminus f^* \end{aligned} \tag{11}$$

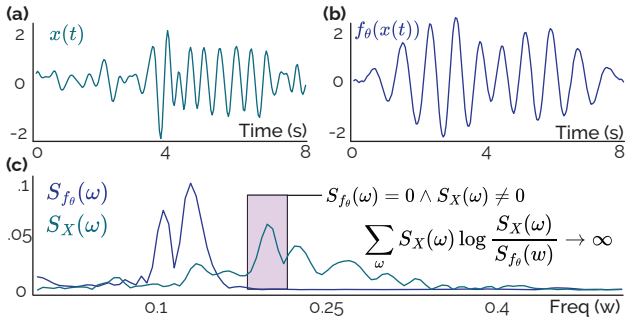



*Figure 2.* **(a)** 8-second time sequence $x(t)$, **(b)** its obtained representation from a model that is trained to maximize periodicity of the signal, **(c)** the normalized spectra of the sample and representation. As the model is trained to minimize the spectral entropy, the values of $S_{f_\theta}(\omega)$ diminish to zero, making the divergence infinity.

The learner which is regularized by the proposed loss function is similar to an adaptive sinusoidal selector based on the Fourier transform. In uniformly sampled time sequences, periodicity is typically identified by the highest value in the Fourier transform (Li et al., 2010). However, this approach fails when there exist multiple periodicities within the frequency range of interest, or when the desired periodic pattern is obscured by noise. The proposed training regularizers help the model learn the desired pattern in the time domain such that it can detect the signal even under the noise. The characteristic of this loss is provided by the minimization of the spectral entropy. This is evident when considering that, if the spectral entropy term is excluded from the overall loss function, the optimization converges to a bandpass-filtered Fourier transform of the original signal.

## 3. Experiments

We compare our method with several techniques, from traditional heuristic-based methods such as autocorrelation and Fourier transform-based approaches, as well as learning-based methods including supervised and self-supervised.

### 3.1. Datasets

We conducted experiments on eight datasets across three tasks, including heart rate (HR) estimation from electrocardiogram (ECG) and photoplethysmography (PPG) signals, step counting using inertial measurements (IMUs), and respiratory rate (breathing) estimation from PPG signals. We provided short descriptions of each dataset below, and further detailed information can be found in Appendix B.

**Heart rate** For ECG, we used PTB-XL (Wagner et al., 2020) to investigate performance in a dataset with cardiovascular diseases, WESAD (Schmidt et al., 2018), and DaLiA (Reiss et al., 2019) to study the real-life settings, where the signals are corrupted with different level of noise as they are collected during activities like cycling. We used the IEEE SPC12 with 22 (Zhang et al., 2015), and DaLiA (Reiss et al., 2019) for PPG-based HR prediction.

**Respiratory rate** We used CapnoBase (Karlen et al., 2013) with two different window sizes and BIDMC (Pimentel et al., 2017) for estimating respiratory rate (RR) from the PPG, where both datasets provide true RR using gold standard respiratory signals and capnography.

**Step counting** We used the Clemson dataset (Mattfeld et al., 2017), which is proposed to improve pedometer evaluation. We conducted experiments using inertial measurements from the wrist with both regular and semi-regular walking settings where labels are available through videos.

### 3.2. Baselines

**Self-supervised** We compared our method with self-supervised learning techniques within the linear evaluation setting (Chen et al., 2020), including SimCLR (Chen et al., 2020), NNCLR (Dwibedi et al., 2021), BYOL (Grill et al., 2020), TS-TCC (Eldele et al., 2021), TS2Vec (Yue et al., 2022), VICReg (Bardes et al., 2022), Barlow Twins (Zbontar et al., 2021), and SimPer (Yang et al., 2023).

**No supervision** We conducted experiments with a learning-based method that requires no labels for fine-tuning. Specifically, we compared our work with SiNC (Speth et al., 2023) which is a non-contrastive unsupervised learning framework for periodicity detection without using any labels while imposing random variance constraints on the batch.

**Supervised** We included two supervised models: a fully convolutional network and a convolutional network with LSTM (Hochreiter & Schmidhuber, 1997), both commonly used in time series tasks previously (Qian et al., 2022).

**Traditional** We also included traditional heuristic-based methods, i.e., the Fourier transformation and autocorrelation function, where both are widely used before (Saul & Allen, 2000) to detect periodicity in time series. Additionally, we compared our method with RobustPeriod (Wen et al., 2021) which is a rule-based periodicity detection approach by integrating information from both time and frequency domains. More details for each baseline are given in Appendix C.

### 3.3. Implementation

**Self-supervised** We follow the same implementation setup with previous works (Qian et al., 2022; Demirel & Holz, 2023) for self-supervised learning in time series. Specifically, we use a combination of convolutional with LSTM-based network, which shows superior performance in many time series tasks (Qian et al., 2022; Biswas et al., 2019), as backbones for the encoder $f_\theta(.)$ where the projector is two fully-connected layers. During pre-training, we use InfoNCE (for contrastive learning-based methods) as the loss function, which is optimized using Adam (Kingma & Ba, 2015) with a learning rate of $0.003$. We train the models with a batch size of 256 for 120 epochs and decay the learning rate using the cosine decay. After pre-training, we train a single linear layer classifier on features extracted from the frozen pre-trained network, i.e., linear probing. Although we aim to enhance performance by experimenting with various architectures and hyperparameters, we consistently observed the best average performance across datasets with the specified implementation details, similar to previous studies. Detailed hyperparameters and augmentations for each SSL technique can be found in Appendix C.3.

**No supervision** Unlike the self-supervised methods, we employ the backbone as a representation learner without having a non-linear projection layer. In our initial experiments, we observed that LSTM or temporal dilations cause loss of frequency information similar to (Speth et al., 2023), we, therefore, use U-Net (Ronneberger et al., 2015) as the encoder architecture with single dimensional layers for time series. We use the proposed loss function in Equation 11, which is optimized using Adam with a learning rate of $0.001$. We train the model with a batch size of 512 and reduce the learning rate by half when the training loss stops decreasing for 15 consecutive epochs. We used the same implementation for all tasks and datasets while giving the same importance to each loss term in Equation 11 without tuning, i.e., no specific task or dataset optimization is performed.

For both learning paradigms, model selection is performed on the training sets with the lowest loss value similar to (Gideon & Stent, 2021; Demirel & Holz, 2023). The reported results are mean and standard deviation values across three independent runs with different random seeds.

**Supervised** We follow the same implementation as previous works (Qian et al., 2022; Demirel & Holz, 2023) for supervised learning in time series. Specifically, we use two different network architectures: the first is a combination of convolutional and LSTM-based networks (Qian et al., 2022; Biswas et al., 2019), similar to the self-supervised implementation of the encoder architecture, and the second is a fully convolutional neural network that includes a final linear layer in the end instead of a projection block. For the training of both architectures, we use categorical cross entropy as the loss function, which is optimized using Adam with a learning rate of $0.001$. We train the models with a batch size of 64 for 120 epochs. The model selection is performed on the validation sets with the lowest loss, where the validation set is created by randomly splitting 10% of remaining data after excluding the test subjects. Details of the architectures and parameter comparisons are provided in Appendix D.1. The performance of various architectures with different sizes is discussed in Appendix E.3.

**Traditional** We also compare our method with the rule-based periodicity detection methods in addition to the learning-based techniques. In our evaluation, we consider the frequency domain approach, which detects underlying periodic patterns using Fourier transforms, and the time domain, where signals are correlated with themselves through autocorrelation function (ACF) (Wen et al., 2023). Additionally, we compare our method with an approach that combines the time and frequency domains using the periodogram and the ACF, integrating the Fisher method for periodicity detection (Wen et al., 2021). Since these methods are not learning-based, they fail to detect periodicities when evaluated on datasets containing periodic noise in the same band, limiting generalization across various conditions. Since traditional methods lack randomness, we conducted the experiments only once and reported the results, i.e., no standard deviation is calculated for rule-based methods.

## 4. Results and Discussion

We present the main results of our proposed approach compared to state-of-the-art methods across the three time series tasks in eight datasets in Tables 1 to 4. Overall, the proposed set of regularizers has demonstrated substantial performance improvements, reaching up to 45–50% in some tasks, compared to other supervised and self-supervised learning methods with rule-based techniques. More importantly, our method outperforms other supervised learning-based techniques, which require label information, while eliminating

*Table 1.* Performance comparison of ours with other methods in *ECG* datasets for HR detection

| Method | PTB-XL | | | $\text{DaLiA}_{ECG}$ | | | WESAD | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| *Heuristic* | | | | | | | | | |
| Fourier | 10.51 | 28.65 | 49.12 | 4.06 | 15.46 | 73.73 | 4.88 | 18.57 | 56.69 |
| Autocorrelation | 8.64 | 19.93 | 59.63 | 7.07 | 14.07 | 76.34 | 4.19 | 8.046 | 89.90 |
| RobustPeriod | 72.79 | 80.87 | -2.93 | 12.79 | 23.21 | 37.49 | 25.16 | 30.18 | 10.23 |
| *Supervised* | | | | | | | | | |
| DCL | 6.08±0.78 | 14.23± 1.43 | 76.40±6.02 | 3.91±0.37 | 6.96±1.06 | 91.42±2.5 | 9.14±0.78 | 7.23±0.75 | 85.44±6.32 |
| CNN | 9.09±0.27 | 16.37±0.33 | 66.11±1.75 | 5.39±0.16 | 8.36±0.15 | 89.44±0.72 | 10.69±0.67 | 13.57±1.01 | 50.10±12.5 |
| *Self-Supervised* | | | | | | | | | |
| SimCLR | 9.13±0.73 | 18.78±0.12 | 70.67±0.55 | 6.23±1.10 | 14.38±1.67 | 68.46±5.25 | 7.09±0.24 | 11.05±0.51 | 65.80±2.96 |
| NNCLR | 10.05±2.06 | 18.74±1.96 | 61.50±7.71 | 8.33±1.85 | 10.89±2.08 | 79.06±8.01 | 8.84±1.14 | 11.64±1.33 | 59.48±9.39 |
| BYOL | 11.92±3.44 | 24.07±8.94 | 50.67±19.7 | 7.39±0.27 | 10.55±0.39 | 81.57±0.93 | 12.21±0.35 | 15.93±0.45 | 28.99±2.55 |
| TS-TCC | 10.13±0.52 | 18.79±0.65 | 54.23±4.62 | 5.13±0.08 | 7.75±0.13 | 90.65±0.09 | 6.16±0.29 | 8.27±0.40 | 83.30±1.68 |
| TS2Vec | 9.52±3.54 | 24.26±6.46 | 52.98±10.08 | 4.78±0.23 | 7.14±0.18 | 88.95±0.13 | 6.24±0.17 | 7.33±0.37 | 84.12±2.10 |
| VICReg | 15.09±0.23 | 23.51±0.34 | 01.35±0.53 | 13.12±4.66 | 18.82±4.30 | 66.18±10.15 | 10.29±0.62 | 13.93±0.84 | 45.27±5.33 |
| Barlow Twins | 13.78±3.73 | 22.96±5.30 | 43.90±10.03 | 13.47±0.50 | 18.02±0.58 | 56.97±3.23 | 11.61±0.34 | 15.09±0.50 | 31.19±3.37 |
| SimPer | — | — | — | 10.11±2.53 | 16.12±1.57 | 60.62±7.23 | 8.16±0.27 | 12.13±0.62 | 52.17±3.21 |
| *No Supervision* | | | | | | | | | |
| SiNC | 3.97±0.01 | 13.71±0.03 | 79.27±0.13 | 1.72±0.013 | 2.65±0.10 | 98.96±0.03 | 2.41±0.01 | 4.13±0.13 | 93.21±0.14 |
| Ours | **3.75**±0.02 | **13.58**±0.03 | **79.30**±0.17 | **1.29**±0.001 | **2.08**±0.01 | **99.12**±0.01 | **2.13**±0.01 | **3.88**±0.06 | **93.52**±0.61 |

*Table 2.* Comparison of methods in *Step counting*

| Method | Clemson (Regular) | | Clemson (Semi-regular) | |
|---|---|---|---|---|
| | MAPE↓ | MAE↓ | MAPE↓ | MAE↓ |
| *Heuristic* | | | | |
| Fourier | 12.16 | 7.60 | 44.94 | 13.23 |
| Autocorrelation | 39.96 | 24.45 | 47.64 | 13.76 |
| RobustPeriod | 14.87 | 24.20 | 47.39 | 13.91 |
| *Supervised* | | | | |
| DCL | 5.99±0.34 | 3.45±0.32 | 19.59±1.54 | 8.98±0.87 |
| FCN | 6.15±0.60 | 3.53±0.33 | **17.22**±0.31 | **6.98**±0.23 |
| *Self-Supervised* | | | | |
| SimCLR | 6.22±0.17 | 5.52±0.14 | 24.41±1.80 | 14.22±1.52 |
| NNCLR | 6.26±0.81 | 5.58±0.72 | 25.12±0.74 | 15.38±0.58 |
| BYOL | 6.69±0.54 | 5.95±0.48 | 25.48±0.46 | 15.22±0.30 |
| TS-TCC | 6.01±0.33 | 3.80±0.38 | 20.08±0.33 | 10.95±0.33 |
| TS2Vec | 6.54±0.76 | 5.17±0.64 | 26.80±0.60 | 16.62±0.46 |
| VICReg | 10.93±3.96 | 9.81±3.52 | 26.90±1.02 | 15.92±0.84 |
| Barlow Twins | 6.87±0.16 | 5.46±0.13 | 19.97±0.70 | 11.14±0.46 |
| SimPer | 7.03±0.57 | 6.75±1.43 | 23.43±0.95 | 15.12±0.41 |
| *No Supervision* | | | | |
| SiNC | 22.92±6.61 | 13.83±4.02 | 37.12±0.01 | 12.31±0.28 |
| Ours | **5.95**±0.21 | **3.42**±0.17 | 35.18±0.02 | 14.21±0.97 |

*Table 3.* Comparison of methods in *Respiration*

| Method | CapnoBase (64-second) | | CapnoBase (32-second) | |
|---|---|---|---|---|
| | MAE↓ | RMSE↓ | MAE↓ | RMSE↓ |
| *Heuristic* | | | | |
| Fourier | 4.08 | 4.96 | 4.66 | 5.47 |
| Autocorrelation | 23.08 | 26.65 | 36.77 | 40.45 |
| RobustPeriod | 7.94 | 8.50 | 8.03 | 8.55 |
| *Supervised* | | | | |
| DCL | 5.76±0.28 | 7.45±0.27 | 5.74±0.08 | 7.68±0.07 |
| FCN | 6.00±0.20 | 8.15±0.28 | 5.41±0.24 | 7.57±0.40 |
| *Self-Supervised* | | | | |
| SimCLR | 3.72±0.27 | 4.94±0.20 | 3.93±0.70 | 5.48±1.00 |
| NNCLR | 3.70±0.52 | 4.83±0.45 | 3.86±0.69 | 5.58±1.06 |
| BYOL | 4.09±0.40 | 5.45±0.42 | 4.40±1.19 | 6.05±1.58 |
| TS-TCC | 4.37±0.45 | 5.74±0.41 | 4.74±1.60 | 6.38±1.84 |
| TS2Vec | 7.00±3.30 | 8.44±4.62 | 5.57±2.76 | 7.93±3.83 |
| VICReg | 3.99±0.43 | 5.21±0.42 | 4.05±0.83 | 5.77±1.19 |
| Barlow Twins | 3.56±0.43 | 4.76±0.50 | 4.21±1.01 | 5.85±1.40 |
| SimPer | 3.89±0.64 | 5.03±1.26 | 5.01±0.90 | 7.96±1.75 |
| *No Supervision* | | | | |
| SiNC | 4.11±0.10 | 5.44±0.31 | 4.09±0.04 | 5.29±0.06 |
| Ours | **3.40**±0.20 | **4.41**±0.43 | **3.74**±0.03 | **4.77**±0.12 |

the necessity for implementing specialized data augmentation methods to learn periodic representations from data.

Tables 3 and 4 show that the rule-based methods fail to detect periodicity when applied to datasets containing multiple periodic components with noise. For example, autocorrelation, one of the most common methods for periodicity detection (Wu et al., 2021; Wang et al., 2006), performs poorly in various datasets that contain multiple periodicities with a high noise where the periodicity of interest is hidden.

Another interesting observation from the results is that the SimPer, designed for extracting representations related to periodic information from data similar to this work, performs worse when applied to noisy time series compared to other self-supervised techniques, despite sharing the same downstream task. This might be attributed to the fact that SimPer designed specialized augmentations that alter the periodicity of the sample within and contrast through that for representation learning. However, when dealing with time series containing multiple periodicities under noisy conditions, the specific augmentation can lead to spectral overlapping, resulting in suboptimal performance. Moreover, since there is no label information during self-supervised training, this method can also learn the features related to periodic noise, which is impossible to filter out as it lies on the same frequency band with the interested periodicity.

From the results, we can see that our proposed method significantly outperforms SiNC and VICReg, which impose a hinge loss on the variance over a batch of representations to enforce diverse outputs to prevent collapse, by a large margin (up to 73.9% with a 10.1% on average in tasks).

These empirical findings suggest that the assumption of having diverse samples in a batch is quite arguable, especially depending on the downstream task, where the pattern of interest can be severely hidden by noise such that the model can learn random features from the samples to have diversity in a batch of representations. The empirical results align with our initial proposition and the motivation behind introducing a novel regularizer term, which prevents the model collapse by ensuring that the extracted representations retain traces from the samples from which they originated rather than relying on random variance constraints in the batch.

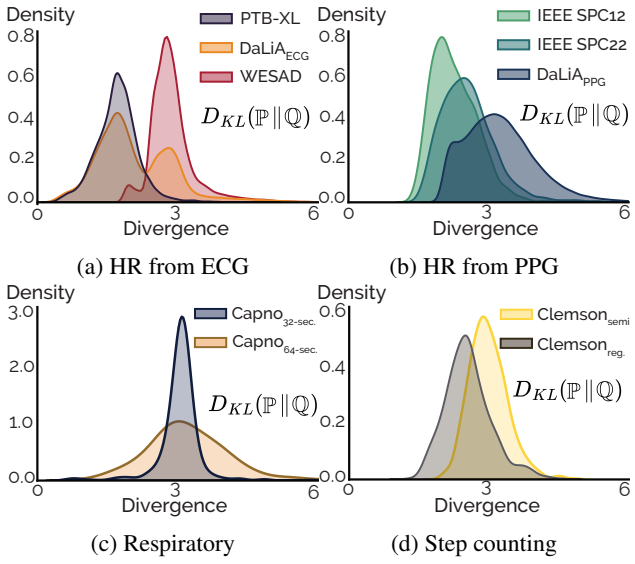
Another interesting outcome of the results is when our method is outperformed by supervised and self-supervised methods while having a minor performance gap with unsupervised techniques, which is the case for DaLiA$_{PPG}$ and Clemson semi-regular. We conjecture that the performance difference between supervised and unsupervised methods in these datasets is due to the average level of noise in the samples, making the periodic pattern of interest completely hidden and hard to extract without supervision. Similarly, in the case of semi-regular walking, the periodic patterns in the training decrease significantly compared to regular, leading to poor performance by unsupervised methods due to their inability to effectively learn desired patterns. Since our approach learns the function $f_\theta$ by minimizing the expectation of the proposed loss, which is known as the empirical risk minimization (ERM), the model converges to a point where the risk is minimal in the training set. When the process is unsupervised, the model might inadvertently learn the periodic noise or random features from the data, which leads to failures during the evaluation. Contrarily, in self-supervised and supervised scenarios, models may learn dataset statistics with label information, such as mean values. The model can then use this during evaluation, contributing to improved performance for imbalanced datasets (Yang & Xu, 2020).

These results suggest that the methods should be evaluated with several tasks in multiple datasets to investigate if they

*Table 4.* Performance Comparison of ours with other methods in *PPG* datasets for HR estimation

| Method | IEEE SPC12 | | | IEEE SPC22 | | | DaLiA$_{PPG}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$ ↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| *Heuristic* | | | | | | | | | |
| Fourier | 12.12 | 20.09 | 72.17 | 14.83 | 25.81 | 35.15 | 34.98 | 47.13 | 2.72 |
| Autocorrelation | 56.65 | 64.58 | 24.89 | 41.16 | 49.81 | 3.945 | 30.59 | 39.09 | 9.18 |
| RobustPeriod | 19.56 | 31.99 | 41.10 | 22.34 | 35.47 | 33.50 | 23.43 | 30.12 | 21.53 |
| *Supervised* | | | | | | | | | |
| DCL | 19.90±1.10 | 26.34±1.10 | 25.53±4.2 | 22.43±0.62 | 27.17±0.60 | 11.95±5.1 | **5.97**±0.19 | **11.83**±0.36 | **80.41**±0.9 |
| CNN | 10.05±0.18 | 18.12±0.40 | 67.20±0.9 | 17.97±0.33 | 23.06±0.36 | 21.91±1.76 | 7.35±0.14 | 13.74±0.26 | 74.22±0.55 |
| *Self-Supervised* | | | | | | | | | |
| SimCLR | 12.42±0.05 | 20.96±0.30 | 60.41±0.52 | 21.08±1.79 | 27.94±3.17 | 13.20±9.97 | 12.01±0.14 | 19.46±0.14 | 58.31±0.39 |
| NNCLR | 13.14±0.49 | 18.86±0.49 | 69.82±0.06 | 20.79±0.90 | 26.39±1.45 | 14.73±6.34 | 12.94±0.31 | 20.02±0.49 | 51.12±2.54 |
| BYOL | 18.71±0.93 | 25.01±1.50 | 48.82±4.36 | 20.01±0.80 | 25.58±1.19 | 15.25±1.45 | 11.67±0.32 | 17.57±0.23 | 63.96±0.97 |
| TS-TCC | 11.56±0.41 | 18.04±0.66 | 68.38±1.41 | 18.77±0.23 | 23.90±0.31 | 28.73±1.40 | 8.12±0.30 | 14.89±0.21 | 67.13±0.53 |
| TS2Vec | 9.75±0.08 | 17.82±0.43 | 75.43±0.33 | 25.77±0.17 | 31.90±0.16 | 06.40±2.31 | 10.83±0.13 | 17.89±0.19 | 60.10±0.62 |
| VICReg | 13.17±0.82 | 20.38±1.27 | 59.76±4.16 | 24.17±0.62 | 30.53±0.41 | 15.28±3.45 | 14.90±0.16 | 21.94±0.11 | 45.38±0.07 |
| Barlow Twins | 13.22±0.34 | 20.42±0.88 | 64.51±4.01 | 25.14±0.89 | 31.35±0.82 | 10.15±2.04 | 18.26±0.57 | 23.41±0.29 | 23.42±7.30 |
| SimPer | 15.10±0.20 | 21.20±0.26 | 52.75±0.84 | 26.30±0.35 | 30.90±0.63 | 05.30±3.38 | 16.35±0.36 | 23.00±0.63 | 38.11±1.38 |
| *No Supervision* | | | | | | | | | |
| SiNC | 19.34±5.38 | 28.44±5.41 | 38.35±7.60 | 21.93±5.46 | 25.52±5.18 | 24.61±6.20 | 14.20±1.30 | 25.66±1.39 | 27.91±4.65 |
| Ours | **9.30**±0.10 | **16.50**±0.20 | **77.60**±0.43 | **10.27**±0.37 | **19.62**±0.71 | **44.10**±0.89 | 27.41±4.73 | 31.26±4.55 | 18.12±3.86 |

generalize and perform well under different characteristics and conditions, i.e., noise types and levels, narrow/wide bandwidth for the periodicity of interest. Otherwise, the hyperparameters (specific augmentations with thresholds) can be optimized for a single task while performing poorly for the rest. It is important to emphasize that our implementation (the model architecture and weights for loss terms) remained consistent across all experiments, ensuring a standardized evaluation. We conduct detailed ablation experiments to investigate further the impact of components of our method on the performance across the datasets.




(a) HR from ECG (b) HR from PPG

(c) Respiratory (d) Step counting

*Figure 3.* The distribution of the spectral relative entropies between samples $\mathbb{P}$ and optimal waveforms $\mathbb{Q}$ which are obtained for each sample using the labels. In other words, $S_X(\omega) \sim \mathbb{P}$, and $S_{f_\theta^*}(\omega) \sim \mathbb{Q}$, where $f_\theta^*$ is the optimal model that extracts representations that only contain the desired periodicity.

### 4.1. Ablation Study

In this section, we present a comprehensive examination of our proposed method and the effect of its components on performance of the models for detecting the periodicity of interest. Mainly, we investigate the effect of each proposed loss term on performance across datasets while establishing connections between dataset characteristics in relation to the average noise levels and learning paradigms. Therefore, first, we extend our investigation for the case when the supervised algorithms significantly outperform our approach. Figure 3 illustrates the distribution of relative entropies computed between the spectra of original samples and optimal waveforms using a continuous probability density curve (Waskom, 2021), where the optimal samples only contain a pure sinusoidal at the frequency of the interest.

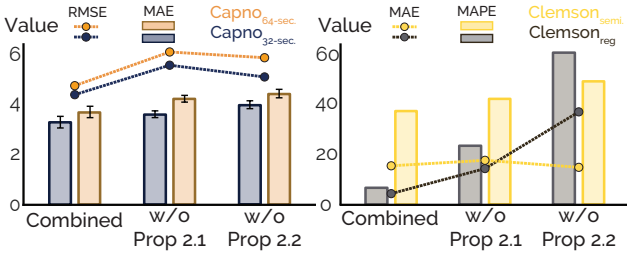
Figure 3 gives an explanation for the performance variation in supervised and unsupervised learning across datasets by demonstrating and comparing the level to which the pattern of interest is hidden by noise. Heuristic-based approaches typically require the SNR of a sample to exceed a certain threshold to detect the pattern, which can vary depending on the specific method, as these methods are not learning-based. In contrast, our method is a learning-based approach designed to train a model that seeks to learn the patterns of interest using the training set and detect those patterns even when they are hidden by noise or other periodicities in unseen samples. One significant drawback of relying on complete unsupervision for a learning-based method is that if the training set lacks sufficient diverse samples with a relatively high SNR for the model to learn the desired periodic patterns, the model may end up learning noise or random features from samples instead of the desired periodic pattern. We, therefore, believe that the average SNR of a dataset is particularly important during the training

*Table 5.* Ablation on proposed regularizers in *PPG* datasets for HR estimation

| Method | IEEE SPC12 | | | IEEE SPC22 | | | DaLiA$_{PPG}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| Combined ($\mathcal{L}$) | **9.30**±0.10 | **16.50**±0.20 | **77.60**±0.43 | **10.27**±0.37 | **19.62**±0.71 | **44.10**±0.89 | 27.41±4.73 | 31.26±4.55 | 18.12±3.86 |
| w/o Prop. 2.1 | 12.69 (-3.39) | 20.92 (-4.42) | 71.01 (-6.59) | 14.51 (-4.24) | 25.30 (-5.68) | 38.03 (-6.07) | 18.71 (+4.7) | 28.63 (+2.63) | 31.13 (+10.0) |
| w/o Prop. 2.2 | 39.14 (-29.84) | 46.01 (-29.51) | 15.95 (-61.65) | 39.71 (-29.56) | 42.66 (-23.04) | 0.771 (-43.32) | 35.26 (-11.85) | 40.13 (-8.87) | 10.03 (-10.09) |
| Fourier | 12.12 (-2.82) | 20.09 (-4.59) | 72.17 (-5.43) | 14.83 (-4.56) | 25.81 (-6.19) | 35.15 (-8.95) | 34.98 (-11.57) | 47.13 (-15.87) | 2.72 (-18.40) |

of unsupervised and self-supervised learning methods (when there is no label information), as a higher average SNR indicates that relevant patterns, those related to downstream tasks, are distinguishable from noise.

Second, we analyze the contribution and impact of each proposed regularizer on model performance across various datasets under different levels and types of noise. We, therefore, trained the neural network models with the same original implementation settings, i.e., the same architecture and training hyperparameters, using the different combinations of the loss components to analyze their contributions separately. Specifically, we first train the models without




(a) Respiratory (b) Step counting

*Figure 4.* The ablation experiments on the effect of each loss term on the *respiratory rate estimation* Figure 4a and *step counting* Figure 4b tasks. The x and y axes of both figures represent the loss combinations and corresponding error values for each task specific metric, respectively. The error decreases significantly when the model is trained using the combination of the proposed loss terms. The error bars for the second part are ignored as the deviation between random seeds is lower than 1–2% of the overall error.

maximizing periodicity and while ensuring the extracted representations retain information to the original samples in the desired frequency band, i.e., $\mathcal{L}_1 = \mathcal{L}_{ds} + \mathcal{L}_{bw}$, without maximizing the periodicity (w/o Proposition 2.1, the second rows in Tables 5 and 6). Then, we train the models to maximize the periodicity without forcing the extracted representations to retain information from the original samples, i.e., $\mathcal{L}_2 = \mathcal{L}_{se} + \mathcal{L}_{bw}$, without preventing the collapse (w/o Proposition 2.2, the third rows in Tables 5 and 6). We also add the bandwidth loss, a common practice in the literature, to the loss variants for each case. The experimental results for each dataset are presented in Tables 5 and 6 for heart rate detection task from PPG and ECG signals, respectively, and in Figure 4 for respiration and step counting tasks.

When the models are trained without maximizing the periodicity in the extracted representations (w/o Proposition 2.1), the performance of the models is similar to periodicity detection using the Fourier transformation where the strongest harmonic is detected from the frequency spectrum of samples. This behavior is expected since the applied regularizers transform the model into a non-adaptive bandpass filter with cut-off frequencies set according to the task. The combination of the proposed loss terms consistently performs well in all tasks, especially when the models have sufficient examples with a high SNR to learn the underlying periodic patterns of interest in the training set. However, when the training data is noisy, i.e., the majority of the training set includes samples with low SNR, where the desired patterns are completely hidden, the Fourier transformation method with bandpass performs well as can be seen in Table 5.

The results obtained from the ablation studies also support the previous theoretical propositions and our motivation for presenting a novel regularizer for preventing the degenerate solution, i.e., the model collapse. For example, a closer inspection of these results shows that when we train the models without relating representations with samples from which they originated, i.e., models that are trained only using $\mathcal{L}_1$ (w/o Proposition 2.2), the models generally performed the worst among the three variants of the combinations, except for the ECG signals where the noise is less significant compared to other signal types, i.e., the periodic pattern of interest is more obvious. This observation constitutes an empirical evidence for our proposed proposition, where a learner, which is only trained to maximize peri-

*Table 6.* Ablation on proposed regularizers in *ECG* datasets for HR detection

| Method | PTB-XL | | | DaLiA$_{ECG}$ | | | WESAD | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| Combined ($\mathcal{L}$) | **3.75**±0.02 | **13.58**±0.03 | **79.30**±0.17 | **1.29**±0.001 | **2.08**±0.01 | **99.12**±0.01 | **2.13**±0.01 | **3.88**±0.06 | **93.52**±0.61 |
| w/o Prop. 2.1 | 14.29 (-10.54) | 33.26 (-19.6) | 42.04 (-37.26) | 5.23 (-3.94) | 16.84 (-14.76) | 71.29 (-27.83) | 5.09 (-2.96) | 16.10 (-12.2) | 62.15 (-31.3) |
| w/o Prop. 2.2 | 7.74 (-4.42) | 21.82 (-8.24) | 46.73 (-32.57) | 2.93 (-1.64) | 5.89 (-3.81) | 92.90 (-6.22) | 3.44 (-1.31) | 5.39 (-1.51) | 89.93 (-4.59) |
| Fourier | 10.51 (-6.76) | 28.65 (-15.07) | 49.12 (-30.18) | 4.06 (-2.77) | 15.46 (-13.38) | 73.73 (-25.39) | 4.88 (-2.75) | 18.57 (-14.69) | 56.69 (-36.93) |

odicity in uniformly sampled time series within a specific frequency band, can converge to a degenerate solution or learn the random waveforms in the signal. Notably, this degenerated model even fails to identify obvious and trivial periodic patterns that can be detected by traditional rule based Fourier transformation or autocorrelation methods.

More ablations about the sensitivity of loss terms can be found in Appendix E.2. Investigations regarding the effect of architectures in supervised learning methods are given in Appendix E.3. Comparative results regarding the performance of our approach with traditional methods that are specifically tailored for the applications are given in Appendix E.4. Examples that visually demonstrate the learned representations from samples can be found in Appendix F. Appendix G provides a detailed evaluation and comparison of the computational efficiency of our proposed method.

## 5. Related Work

**Periodicity Detection** Numerous studies have investigated periodicity detection for several time series applications (Saul & Allen, 2000; Donoghue et al., 2020). Detecting the periodicity of physiological signals, such as heart and respiratory rate, plays a pivotal role in understanding individuals' health conditions (Penzel et al., 2003). Periodicity detection also finds application in workload forecasting (Yu et al., 2024), anomaly detection (Wen et al., 2021), and voice/speech analysis (Saul & Allen, 2000). For instance, it helps identify features like pitch and intonation, which convey essential aspects of human communication, including recognizing and understanding speech (Stevens, 2000). Although current methods are valid in addressing specific problems, a common characteristic is the development of specialized tools tailored to particular issues with optimized hyperparameters, i.e., several thresholds (Wen et al., 2021), often lacking generalized solutions under noisy data (Saul & Allen, 2000). Moreover, if there are two periodicities in the interested frequency bandwidth (i.e., noise and pattern of interest), these methods will fail to detect the desired one as they lack the adaptability to learn to distinguish between multiple periodic patterns (Wen et al., 2023). In contrast, our proposed method, which is adaptive and learns the desired pattern from the data in an unsupervised way, can detect periodicity across various conditions in three time series tasks without requiring labels for training or task-specific data augmentations with parameter optimizations.

**Unsupervised Learning** Self-supervised learning methods have gained significant attention as they enable the discovery of useful representations from data without labels by designing pretext tasks that change the unsupervised learning problem into a supervised one such as predicting the rotation of images (Gidaris et al., 2018), or contexts (Doersch et al., 2015; Pathak et al., 2016). Recently, SSL techniques, particularly in the vision domain, have been explored for periodicity detection from the sequence of frames (Yang et al., 2023; Wang et al., 2022a). Most of these methods (Yang et al., 2023; Gideon & Stent, 2021) generate similar and dissimilar (positive/negative) pairs by applying tailored augmentations and learn periodic representations with slight modifications to the InfoNCE loss (van den Oord et al., 2018). However, these approaches do not consider the challenges posed by noisy non-stationary time series data where signals may exhibit multiple periodicities in the interested bandwidth under significant noise (Temko, 2017). Moreover, the usage of augmentations, in such cases, can be prone to issues like spectrum overlap, i.e., aliasing, resulting in the loss of information (Oppenheim & Schafer, 2009).

Recent approaches have used regularizers to prevent collapse issue by penalizing the model when it produces similar embeddings within a batch (Speth et al., 2023) instead of using similar/dissimilar samples as in contrastive learning. For example, VICReg applies the hinge loss on the variance over a batch of predictions to enforce diverse representations/outputs (Bardes et al., 2022). However, having diverse samples in a batch is an overly optimistic claim, because during self-supervised training, where there is no label information available to the learners, this regularizer may punish the models for learning correct representations. Even worse, the model might learn the noisy patterns to satisfy the arbitrarily set variance constraint in a batch. Therefore, we take a different approach and introduce a new method to prevent the model collapse by ensuring the representations retain information from the original samples. In other words, our approach hypothesizes that each representation should maintain a trace of the sample from which it is derived.

## 6. Conclusion

In this paper, we proposed a set of regularizers for detecting desired periodic patterns in time series sequences without requiring any label information or specialized data augmentation techniques. Theoretically, we proved that an optimized learner trained with the proposed loss will detect the periodic pattern in the sequence while preventing degenerate solutions. In contrast to previous methods that rely on optimistic assumptions on batch statistics to prevent the collapse of unsupervised models, we presented a novel approach that guarantees representations to preserve information from the original samples, thus relaxing assumptions on the batch. Empirically, we showed that our method outperforms the prior works, achieving significant performance gains of up to 40–45%, in three real-world tasks that involve diverse noisy conditions. We believe that the methods introduced in this work have the potential to significantly improve learning solutions for a wide variety of time series tasks.

## Impact Statement

This paper aims to advance machine learning by introducing a new method for detecting periodic patterns in noisy time series data without relying on labeled data. Since our method operates in an unsupervised manner, it can leverage vast amounts of unlabeled data commonly generated in everyday life. This characteristic underscores its potential for real-world deployment, tapping into the rich resource of naturally occurring time series data for applications such as health monitoring, behavior analysis, and beyond.

## References


Balestriero, R. and LeCun, Y. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=jQgsZDspz5h.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Barlow, H. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12(3):241, aug 2001. doi: 10.1088/0954-898X/12/3/301. URL https://dx.doi.org/10.1088/0954-898X/12/3/301.

Bayoumy, K., Gaber, M., Elshafeey, A., Mhaimeed, O., Dineen, E. H., Marvel, F. A., Martin, S. S., Muse, E. D., Turakhia, M. P., Tarakji, K. G., and Elshazly, M. B. Smart wearable devices in cardiovascular care: where we are and how to move forward. *Nature Reviews Cardiology*, 18(8):581–599, August 2021. ISSN 1759-5010. doi: 10.1038/s41569-021-00522-7. URL https://www.nature.com/articles/s41569-021-00522-7.

Biswas, D., Everson, L., Liu, M., Panwar, M., Verhoef, B.-E., Patki, S., Kim, C. H., Acharyya, A., Van Hoof, C., Konijnenburg, M., and Van Helleputte, N. Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment. *IEEE Transactions on Biomedical Circuits and Systems*, 13(2):282–291, 2019. doi: 10.1109/TBCAS.2019.2892297.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Demirel, B. U. and Holz, C. Finding order in chaos: A novel data augmentation method for time series in contrastive learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dbVRDk2wt7.

Doersch, C., Gupta, A. K., and Efros, A. A. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1422–1430, 2015.

Donoghue, T., Haller, M., Peterson, E. J., Varma, P., Sebastian, P., Gao, R., Noto, T., Lara, A. H., Wallis, J. D., Knight, R. T., Shestyuk, A., and Voytek, B. Parameterizing neural power spectra into periodic and aperiodic components. *Nature Neuroscience*, 23(12):1655–1665, December 2020. ISSN 1546-1726. doi: 10.1038/s41593-020-00744-x. URL https://www.nature.com/articles/s41593-020-00744-x.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9568–9577, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00945. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00945.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., and Guan, C. Time-series representation learning via temporal and contextual contrasting. In Zhou, Z.-H. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2352–2359. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/324. URL https://doi.org/10.24963/ijcai.2021/324. Main Track.

Femiano, R., Werner, C., Wilhelm, M., and Eser, P. Validation of open-source step-counting algorithms for wrist-worn tri-axial accelerometers in cardiovascular patients. *Gait & Posture*, 92:206–211, 2022. ISSN 0966-6362. doi: https://doi.org/10.1016/j.gaitpost.2021.11.035. URL https://www.sciencedirect.com/science/article/pii/S0966636221006196.

Gao, F., Liu, C., Zhang, L., Liu, T., Wang, Z., Song, Z., Cai, H., Fang, Z., Chen, J., Wang, J., Han, M., Wang, J., Lin, K., Wang, R., Li, M., Mei, Q., Ma, X., Liang, S., Gou, G., and Xue, N. Wearable and flexible electrochemical sensors for sweat analysis: a review. *Microsystems & Nanoengineering*, 9(1):1–21, January 2023. ISSN 2055-7434. doi: 10.1038/s41378-022-00443-6. URL https://www.nature.com/articles/s41378-022-00443-6.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

Gideon, J. and Stent, S. The way to my heart is through contrastive learning: Remote photoplethysmography from unlabelled video. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3975–3984, 2021. URL https://api.semanticscholar.org/CorpusID:244347889.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

He, H., Queen, O., Koker, T., Cuevas, C., Tsiligkaridis, T., and Zitnik, M. Domain adaptation for time series under feature and label shifts. In *International Conference on Machine Learning, ICML*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2019. URL https://api.semanticscholar.org/CorpusID:207930212.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Hogan, J. Why don't nurses monitor the respiratory rates of patients? *British Journal of Nursing*, 15(9):489–492, 2006. doi: 10.12968/bjon.2006.15.9.21087. URL https://doi.org/10.12968/bjon.2006.15.9.21087. PMID: 16723921.

Hong, S., Xu, Y., Khare, A., Priambada, S., Maher, K., Aljiffry, A., Sun, J., and Tumanov, A. Holmes: Health online model ensemble serving for deep learning models in intensive care units. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1614–1624, 2020.

Hughes, A., Shandhi, M. M. H., Master, H., Dunn, J., and Brittain, E. Wearable devices in cardiovascular medicine. *Circulation Research*, 132(5): 652–670, 2023. doi: 10.1161/CIRCRESAHA.122.322389. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCRESAHA.122.322389.

Huhn, S., Axt, M., Gunga, H.-C., Maggioni, M. A., Munga, S., Obor, D., Sié, A., Boudo, V., Bunker, A., Sauerborn, R., Bärnighausen, T., and Barteit, S. The impact of wearable technologies in health research: Scoping review. *JMIR Mhealth Uhealth*, 10(1):e34384, Jan 2022. ISSN 2291-5222. doi: 10.2196/34384.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7.

Karlen, W., Raman, S., Ansermino, J. M., and Dumont, G. A. Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE Transactions on Biomedical Engineering*, 60(7):1946–1953, 2013. doi: 10.1109/TBME.2013.2246160.

Khintchine, A. Korrelationstheorie der stationären stochastischen Prozesse. *Mathematische Annalen*, 109 (1):604–615, December 1934. ISSN 1432-1807. doi: 10.1007/BF01449156. URL https://doi.org/10.1007/BF01449156.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kiyasseh, D., Zhu, T., and Clifton, D. A. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*, 2020.

Krause, A., Perona, P., and Gomes, R. Discriminative clustering by regularized information maximization. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information*

*Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/42998cf32d552343bc8e460416382dca-Paper.pdf.

Li, Z., Ding, B., Han, J., Kays, R., and Nye, P. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pp. 1099–1108, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300551. doi: 10.1145/1835804.1835942. URL https://doi.org/10.1145/1835804.1835942.

Mandsager, K., Harb, S., Cremer, P., Phelan, D., Nissen, S. E., and Jaber, W. Association of Cardiorespiratory Fitness With Long-term Mortality Among Adults Undergoing Exercise Treadmill Testing. *JAMA Network Open*, 1(6):e183605–e183605, 10 2018. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2018.3605. URL https://doi.org/10.1001/jamanetworkopen.2018.3605.

Martinez, J., Almeida, R., Olmos, S., Rocha, A., and Laguna, P. A wavelet-based ecg delineator: evaluation on standard databases. *IEEE Transactions on Biomedical Engineering*, 51(4):570–581, 2004. doi: 10.1109/TBME.2003.821031.

Master, H., Annis, J., Huang, S., Beckman, J. A., Ratsimbazafy, F., Marginean, K., Carroll, R., Natarajan, K., Harrell, F. E., Roden, D. M., Harris, P., and Brittain, E. L. Association of step counts over time with the risk of chronic disease in the All of Us Research Program. *Nature Medicine*, 28(11):2301–2308, November 2022. ISSN 1546-170X. doi: 10.1038/s41591-022-02012-w. URL https://www.nature.com/articles/s41591-022-02012-w.

Mattfeld, R., Jesch, E., and Hoover, A. A new dataset for evaluating pedometer performance. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017.

Moss, A., Lee, H., Xun, L., Min, C., Kawsar, F., and Montanari, A. Ultra-low power dnn accelerators for iot: Resource characterization of the max78000. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, SenSys '22, pp. 934–940, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450398862. doi: 10.1145/3560905.3568300. URL https://doi.org/10.1145/3560905.3568300.

Oppenheim, A. V. and Schafer, R. W. *Discrete-Time Signal Processing*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0131988425.

Oppenheim, A. V., Willsky, A. S., and Nawab, S. H. *Signals & Systems (2nd Ed.)*. Prentice-Hall, Inc., USA, 1996. ISBN 0138147574.

Pan, J. and Tompkins, W. J. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985. doi: 10.1109/TBME.1985.325532.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2536–2544. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.278. URL https://doi.org/10.1109/CVPR.2016.278.

Penzel, T., Kantelhardt, J. W., Lo, C.-C., Voigt, K., and Vogelmeier, C. Dynamics of Heart Rate and Sleep Stages in Normals and Patients with Sleep Apnea. *Neuropsychopharmacology*, 28(1), 2003.

Pimentel, M. A. F., Johnson, A. E. W., Charlton, P. H., Birrenkott, D., Watkinson, P. J., Tarassenko, L., and Clifton, D. A. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE Transactions on Biomedical Engineering*, 64(8):1914–1923, 2017. doi: 10.1109/TBME.2016.2613124.

Qian, H., Tian, T., and Miao, C. What makes good contrastive learning on small-scale wearable-based tasks? In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 3761–3771, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539134. URL https://doi.org/10.1145/3534678.3539134.

Reiss, A., Indlekofer, I., Schmidt, P., and Van Laerhoven, K. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14), 2019. ISSN 1424-8220. doi: 10.3390/s19143079. URL https://www.mdpi.com/1424-8220/19/14/3079.

Rocha, L. G., Biswas, D., Verhoef, B.-E., Bampi, S., Van Hoof, C., Konijnenburg, M., Verhelst, M., and Van Helleputte, N. Binary cornet: Accelerator for hr estimation from wrist-ppg. *IEEE Transactions on Biomedical Circuits and Systems*, 14(4):715–726, 2020. doi: 10.1109/TBCAS.2020.3001675.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In

Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing.

Ryoo, M., Rothrock, B., Fleming, C., and Yang, H. J. Privacy-preserving human activity recognition from extreme low resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.11233. URL https://ojs.aaai.org/index.php/AAAI/article/view/11233.

Sandvik, L., Erikssen, J., Ellestad, M., Erikssen, G., Thaulow, E., Mundal, R., and Rodahl, K. Heart rate increase and maximal heart rate during exercise as predictors of cardiovascular mortality: a 16-year follow-up study of 1960 healthy men. *Coronary Artery Disease*, 6(8):667, August 1995. ISSN 0954-6928. URL https://journals.lww.com/coronary-artery/abstract/1995/08000/heart_rate_increase_and_maximal_heart_rate_during.12.aspx.

Saul, L. and Allen, J. Periodic component analysis: An eigenvalue method for representing periodic structure in speech. In Leen, T., Dietterich, T., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., and Van Laerhoven, K. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, pp. 400–408, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356923. doi: 10.1145/3242969.3242985. URL https://doi.org/10.1145/3242969.3242985.

Speth, J., Vance, N., Flynn, P., and Czajka, A. Non-contrastive unsupervised learning of physiological signals from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Stevens, K. N. *Acoustic Phonetics*. The MIT Press, 07 2000. ISBN 9780262284387. doi: 10.7551/mitpress/1072.001.0001. URL https://doi.org/10.7551/mitpress/1072.001.0001.

Stracina, T., Ronzhina, M., Redina, R., and Novakova, M. Golden standard or obsolete method? review of ecg applications in clinical and experimental context. *Frontiers in Physiology*, 13, 2022. ISSN 1664-042X. doi: 10.3389/fphys.2022.867033. URL https://www.frontiersin.org/articles/10.3389/fphys.2022.867033.

Temko, A. Accurate heart rate monitoring during physical exercises using ppg. *IEEE Transactions on Biomedical Engineering*, 64(9):2016–2024, 2017. doi: 10.1109/TBME.2017.2676243.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. URL https://api.semanticscholar.org/CorpusID:49670925.

Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., and Schaeffter, T. PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, 7(1):154, May 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0495-6. URL https://www.nature.com/articles/s41597-020-0495-6.

Wang, H., Ahn, E., and Kim, J. Self-supervised representation learning framework for remote physiological measurement using spatiotemporal augmentation loss. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):2431–2439, Jun. 2022a. doi: 10.1609/aaai.v36i2.20143. URL https://ojs.aaai.org/index.php/AAAI/article/view/20143.

Wang, J., Chen, T., and Huang, B. Cyclo-period estimation for discrete-time cyclo-stationary signals. *IEEE Transactions on Signal Processing*, 54(1):83–94, 2006. doi: 10.1109/TSP.2005.859237.

Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022b. URL https://openreview.net/forum?id=ECvgmYVyeUz.

Waskom, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

Wen, Q., He, K., Sun, L., Zhang, Y., Ke, M., and Xu, H. Robustperiod: Robust time-frequency mining for multiple periodicity detection. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, pp. 2328–2337, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383431. doi: 10.1145/3448016.3452779. URL https://doi.org/10.1145/3448016.3452779.

Wen, Q., Yang, L., and Sun, L. Robust dominant periodicity detection for time series with missing data. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095241.

West, M., Prado, R., and Krystal, A. D. Evaluation and comparison of eeg traces: Latent structure in nonstationary time series. *Journal of the American Statistical Association*, 94(446):375–387, 1999. doi: 10.1080/01621459.1999.10474128. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1999.10474128.

Wiener, N. Generalized harmonic analysis. *Acta Mathematica*, 55(none):117 – 258, 1930. doi: 10.1007/BF02546511. URL https://doi.org/10.1007/BF02546511.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235623791.

Yang, R., Song, J., Huang, B., Li, W., and Qi, G. An Energy-Efficient Step-Counting Algorithm for Smartphones. *The Computer Journal*, 65(3):689–700, 08 2020. ISSN 0010-4620. doi: 10.1093/comjnl/bxaa096. URL https://doi.org/10.1093/comjnl/bxaa096.

Yang, Y. and Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Yang, Y., Liu, X., Wu, J., Borac, S., Katabi, D., Poh, M.-Z., and McDuff, D. Simper: Simple self-supervised learning of periodic targets. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=EKpMeEV0hOo.

Yu, H., Liao, C., Liu, R., Li, J., Yun, H., and Wang, X. Amortizedperiod: Attention-based amortized inference for periodicity identification. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=psEswR8Jz4.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. Ts2vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, Jun. 2022. doi: 10.1609/aaai.v36i8.20881. URL https://ojs.aaai.org/index.php/AAAI/article/view/20881.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Zhang, Z., Pi, Z., and Liu, B. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015. doi: 10.1109/TBME.2014.2359372.

# A. Proof

In this section, we present complete proofs of our theoretical study, starting with notations.

## A.1. Notations and Preliminaries

Fourier transform of a real-valued sample with a finite duration is obtained as in Equation 12.

$$\mathcal{F}(\mathbf{x}) = \sum_{n=-\infty}^{\infty} \mathbf{x}_n e^{-j2\pi kn} \tag{12}$$

The absolute frequency spectra of the samples are calculated as below.

$$S_X(w) = \left| \sum_n \mathbf{x}_n e^{-j\omega n} \right|^2, \quad \text{where} \quad \omega = 2\pi f \quad \text{and} \quad f \in [0, f_s/2] \tag{13}$$

We applied probability normalization to the obtained spectra for both original samples and the extracted representations in the desired frequency as in below before calculating the entropy and divergence.

$$\begin{aligned} S_X(\omega^*) &= \frac{S_X(\omega^*)}{\sum_{\omega^*} S_X(\omega^*)}, \quad \text{where} \quad \omega^* \in [0, 2\pi f_s/2] \\ H(S_X) &= \sum_{\omega} S_X(\omega^*) \log \frac{S_X(\omega^*)}{S_X(\omega^*)} \end{aligned} \tag{14}$$

The autocorrelation of a real-valued signal with time shift $\tau$ is calculated as in Equation 15, where the denominator is the sample variance of the time series in that segment.

$$R_{XX}(\tau) = \frac{\sum_{n=1}^{N-\tau} (x_n - \mu_x)(x_{n+\tau} - \mu_x)}{\sum_{n=1}^{N} (x_n - \mu_x)^2} \tag{15}$$

We also used the Wiener-Khintchine (Wiener, 1930; Khintchine, 1934) theorem for calculating the spectral density of the sample $\mathbf{x}$ from the autocorrelation as in Equation 16.

$$S(\omega) = \sum_{\tau} R_{XX}(\tau) e^{-j\omega\tau} \tag{16}$$

Similarly, as the time series samples are absolutely summable in our case, the autocorrelation can be obtained as in below.

$$R_{XX}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) e^{j\tau\omega} d(\omega) \tag{17}$$

Since the spectral density is periodic in the frequency domain, the summation is performed over one period.

### A.2. Proof for Proposition 2.1

**Proposition A.1** (Maximizing periodicity)**.** *Minimizing the spectral entropy of sequential data is nothing but maximizing its periodicity in the time domain.*

$$\begin{aligned} \theta^* &= \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{se} = \underset{\theta}{\operatorname{argmax}}\, \mathcal{I}(\mathbf{y};\mathbf{x}), \\ \textit{where}\ \ \mathcal{L}_{se} &= -\sum_{w} S_{f_\theta}(w) \log S_{f_\theta}(w), \\ S_{f_\theta}(w) = \frac{S_{f_\theta}(w)}{\sum_w S_{f_\theta}(w)},\ S_{f_\theta}(w) &= \left| \sum_n f_\theta(x)_n e^{-j\omega n} \right|^2 \end{aligned} \tag{18}$$

*Proof.* The spectral entropy is maximum (periodicity is minimum) when the sequence generation process is a Gaussian with zero mean, $\sigma^2$ variance, i.e., $W(t) \sim \mathcal{N}(0, \sigma_w^2)$.

$$\mathbb{E}[W(t)] = 0, \quad R_{WW}(\tau) = \mathbb{E}[W(t+\tau)W^*(t)]$$

For samples drawn from the Gaussian process, the correlation is non-zero only when $\tau = 0$ (as they are independent), and it is equal to the variance of the distribution. So;

$$R_{WW}(\tau) = \begin{cases} \sigma_w^2, & \text{if } \tau = 0 \\ 0, & \text{if } \tau \neq 0 \end{cases} \longrightarrow R_{WW}(\tau) = \sigma_w^2 \delta(\tau) \tag{19}$$

Using the Wiener-Khintchine theorem in Equation 16,

$$S_W(\omega) = \sum_{\tau} \sigma_w^2 \delta(\tau) e^{-j\omega\tau} \tag{20}$$

$$S_W(\omega) = \begin{cases} \sigma_w^2, & \text{if } \omega \in (-\pi, \pi) \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

In the last part of the above equation, the single period of the spectra in the frequency domain is considered for all samples. When we apply the probability normalization to the obtained spectra in the desired frequency band,

$$S_W(\omega) = \frac{1}{b-a} \ \text{ where } \ \omega \in [a, b] \subset (-\pi, \pi) \tag{22}$$

The obtained spectrum from the Gaussian process is nothing but the uniform distribution in the interested frequency band. It is known that the Entropy is maximized if the distribution is uniform, therefore;

$$H(S_W) \geq H(S_X), \forall \mathbf{x} \in X \tag{23}$$

The spectral entropy is minimum if the sequence generation process is periodic, i.e., $R_{PP}(\tau) = R_{PP}(\tau + T)$. Since the $R_{PP}(\tau)$ is periodic with T, it can be expressed as a linear combination of harmonically related complex exponentials (Oppenheim et al., 1996), which is also referred to as the Fourier series representation.

$$R_{PP}(\tau) = \sum_{n=-\infty}^{\infty} \alpha_n e^{j\omega_0 n\tau}, \quad \omega_0 = \frac{2\pi}{T} \tag{24}$$

$$\alpha_n = \frac{1}{T} \int_{-T/2}^{T/2} R_{PP}(\tau) e^{-j\omega_0 n\tau} d\tau \tag{25}$$

Using the Wiener-Khintchine theorem in Equation 16, we can write the spectrum density as follows;

$$S_P(w) = \sum_{\tau} R_{PP}(\tau) e^{-j2\pi\tau n} \tag{26}$$

$$S_P(w) = \sum_{\tau} \sum_{n} \alpha_n e^{j\omega_0 n\tau} e^{-j\omega\tau} \tag{27}$$

$$S_P(w) = \sum_{n} \sum_{\tau} \alpha_n e^{-j\tau(\omega - \omega_0 n)} \tag{28}$$

Using the orthogonality of complex exponentials, we can write the above equation as follows,

$$S_P(w) = \begin{cases} 2\pi \sum_n \alpha_n, & \text{for } \omega = \omega_0 n \\ 0, & \text{for } \omega \neq \omega_0 n \end{cases} \longrightarrow S_P(w) = 2\pi \sum_{n-\infty}^{\infty} \alpha_n \delta(\omega - n\omega_0), \tag{29}$$

where the $2\pi$ comes from the single period $(-\pi, \pi)$ summation of the autocorrelation. When we apply the probability normalization to the obtained spectra in the desired frequency band while converting the discrete-time Fourier transform to the discrete Fourier transform*, we can obtain the following.

$$S_P(w) = \delta(\omega - \omega_0) \;\text{ where }\; \omega \in [a, b] \subset (0, \pi) \tag{30}$$

When we consider the positive frequencies, the normalized spectra will contain an impulse at the frequency of the periodicity, which is the case for minimum entropy.

$$H(S_P) \leq H(S_X), \forall \mathbf{x} \in X \tag{31}$$

Combining Equation 23 and the above,

$$H(S_P) \leq H(S_X) \leq H(S_W) \tag{32}$$

Therefore, $H(S_P) \leq \mathcal{L}_{se} \leq H(S_W)$. □

The last inequality concludes the proof by showing that minimizing the spectral entropy of uniformly sampled time series data is equivalent to maximizing its periodicity.

### A.3. Proof for Proposition 2.2

**Proposition A.2** (Degenerate solution)**.** *The relative entropy between the spectral distributions of samples and extracted representations is upper-bounded by a degenerate solution.*

$$\begin{gathered} D_{KL}(\mathbb{P} \parallel \mathbb{Q}) > \mathcal{L}_{ds} \geq 0, \;\; \textit{where} \;\; \textit{Var}(\mathbb{Q}) = 0, \\ \textit{and} \;\; \mathcal{L}_{ds} = \sum_{\omega} S_X(\omega) \log \frac{S_X(\omega)}{S_{f_\theta}(\omega)}, \\ S_X(\omega) = \frac{S_X(\omega)}{\sum_\omega S_X(\omega)}, \; S_X(\omega) = \left| \sum_n x_n e^{-j\omega n} \right|^2 \end{gathered} \tag{33}$$

*Proof.* From Equation 30, we can say that a learner, which is trained to maximize the periodicity, will output representations $(f_\theta(\mathbf{x}))$ that have spectra as in below.

$$S_{f_\theta}(\omega) = \delta(\omega - \omega_0) \tag{34}$$

Nevertheless, this leads to an informational collapse, resulting in a degenerate solution where representations become identical, despite the maximum periodicity present. In other words, if the distribution of representations is defined as $\mathbb{Q}$, the variance will be zero, i.e., $\text{Var}(\mathbb{Q}) = 0$. Therefore,

$$\exists \omega \in (-\infty, \infty), \quad S_{f_\theta}(\omega) = 0 \wedge S_X(\omega) \neq 0, \tag{35}$$

And,

$$D_{KL}(S_X(\omega) \parallel S_{f_\theta}(\omega)) = \sum_{\omega} S_X(\omega) \log \frac{S_X(\omega)}{S_{f_\theta}(\omega)} \to \infty \tag{36}$$

Thus,

$$D_{KL}(\mathbb{P} \parallel \mathbb{Q}) \geq \mathcal{L}_{se} \geq 0 \tag{37}$$

□

The proof suggests that degenerate solutions can be prevented if the learner seeks to minimize the relative entropy between the spectra of samples and extracted representations.

*The Fourier transform is sampled at N equally spaced frequencies (Oppenheim et al., 1996).

## B. Datasets

In this section, we give details about the datasets that are used during our experiments.

### B.1. Heart rate

#### B.1.1. *ECG* BASED

**PTB-XL** The PTB-XL ECG is a large dataset of 21799 clinical 12-lead ECGs, where lead-II was used during experiments, from 18869 patients of 10-second length segments. The dataset itself provides recommended splits into training and test sets. We, therefore, follow the exact recommendation. ten percent of the training training set is used as fine-tuning for self-supervised learning techniques.

**DaLia**$_{ECG}$ PPG dataset for motion compensation and heart rate estimation in Daily Life Activities (DaLia) was recorded from 15 subjects (8 females, 7 males, mean age of 30.6), where each recording was approximately two hours long. The signals were recorded while subjects went through different daily life activities, for instance sitting, walking, driving, cycling, working, and so on. ECG signals were recorded at a sampling rate of 700 Hz. We follow leave-one-out-cross-validation for each subject. For the self-supervised setting, we used the first five subjects for fine-tuning.

**WESAD** The multimodal WESAD data set includes physiological and mobility data from wrist-worn (Empatica E4) and chest-worn (RespiBAN) devices. Data were acquired from 15 subjects and contain multiple features, pulse rate, ECG, and body temperature, as extracted from a wrist-worn device, and blood volume pulse (BVP) and respiration extracted by the chest-worn devices (Schmidt et al., 2018). We evaluate the dataset using leave-one-subject-out.

ECG datasets are standardized as follows. Initially, a fourth-order Butterworth bandpass filter with a frequency range of 0.7–40 Hz is applied to the signals. In the case of DaLia and WESAD, we use an 8-second sliding window with 2-second shifts for segmentation. Differently, for PTB-XL, the ECG segments are pre-provided. After segmentation, we calculated the square of the first differentiation before feeding samples to the models, which helps to emphasize beats (Pan & Tompkins, 1985). Lastly, the signals are normalized to zero to one range.

#### B.1.2. *PPG* BASED

**IEEE SPC** This competition provided a training dataset of 12 subjects (SPC12) and a test dataset of 10 subjects (Reiss et al., 2019). The IEEE SPC dataset overall has 22 recordings of 22 subjects, ages ranging from 18 to 58 performing three different activities (Rocha et al., 2020). Each recording has sampled data from three accelerometer signals and two PPG signals along with a sampling frequency of 125 Hz. All these recordings were recorded from the wearable device placed on the wrist of each individual. All recordings were captured with a 2-channel pulse oximeter with green LEDs, a tri-axial accelerometer, and a chest ECG for the ground-truth HR estimation. During our experiments, we used PPG channels. We use leave-one-out-cross-validation for the SPC12 and SPC22 as source domains similar to the previous setup while the last six subjects of SPC22 are used for source domains to prevent overlapping subjects with SPC12, similar to (Demirel & Holz, 2023).

**DaLia**$_{PPG}$ PPG signals from the DaLia dataset were recorded at a sampling rate of 64 Hz. The first five subjects are used as target domains with leave-one-out-cross-validation, following the (Demirel & Holz, 2023).

All PPG datasets are standardized as follows. Initially, a fourth-order Butterworth bandpass filter with a frequency range of 0.5–4 Hz is applied to PPG signals. Subsequently, a sliding window of 8 seconds with 2-second shifts is employed for segmentation, followed by z-score normalization of each segment. Lastly, the signal is resampled to a frequency of 25 Hz for each segment.

### B.2. Respiratory rate

**CapnoBase** The CapnoBase dataset includes recordings of ECG and PPG along with capnometry signals from 42 subjects (13 adults, 29 children, and neonates). The dataset also includes the inhaled and exhaled carbon dioxide (CO2) signal labeled by the research assistance (Karlen et al., 2013) and used as the reference breathing rate. The dataset is split into 10 folds where each fold contains four subjects. The last five fold is used as the test set for all baselines. For the self-supervised learning, we used seven fold for pre-training and ten percent of the pre-training for fine-tuning.

**BIDMC** The BIDMC dataset consists of ECG, pulse oximetry (PPG), and impedance pneumography respiratory signals. The data was acquired by randomly selecting critically ill patients undergoing hospital care at the Beth Israel Deaconess Medical Centre (Boston, MA, USA). Two trained annotators manually annotated individual breaths in each recording using the respiratory impedance signal. The data set contains 8-minute recordings of ECG, PPG, and impedance pneumography signals from 52 adult patients aged from 19 to older than 90, including 32 females (Pimentel et al., 2017). For each set of annotations, the RR value was determined based on the average time between consecutive breaths within a given window; only those windows of data for which the agreement between both estimates was within 2 breaths per minute were retained. We used 10-fold cross-validation for the evaluation where each fold contains 5 subjects. For the self-supervised learning, we used seven fold for pre-training and ten percent of the pre-training for fine-tuning.

### B.3. Step counting

**Clemson** The Clemson dataset has 30 participants, including 15 males and 15 females. Each participant wore three Shimmer3 sensors. We use the data from the device which is positioned on the non-dominant wrist. The inertial measurement data is recorded at 15 Hz. The accelerometers were set to record from -2 to 2 gravities. We computed the overall magnitude of the accelerometer and used it as input to the models. We used the regular gait experiment, where the participant was instructed to walk two laps around a designated path at their normal walking pace. In the semi-regular gait experiment, participants were instructed to perform a scavenger hunt, locating four objects in four different rooms throughout a building (Mattfeld et al., 2017). We used 10-fold cross-validation for the evaluation where each fold contains 3 subjects. For the self-supervised learning, we used seven fold for pre-training and two fold for fine-tuning.

When we split the datasets for training and testing folds, we ensure that each person's recordings appear in only one set. We followed this procedure across all datasets to ensure that the trained model had not seen the testing data during training.

### B.4. Metrics

We used the common evaluation metrics in the literature for each task. Specifically, we used mean absolute error (MAE), root mean square error (RMSE), and Pearson correlation coefficient for *heart rate prediction* (Speth et al., 2023), and we used the mean absolute percentage error (MAPE) for step counting (Yang et al., 2020; Femiano et al., 2022).

Here, we explain how to calculate each metric for different time series tasks. The MAE, RMSE, and correlation coefficient ($\rho$) are calculated using the following equations:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{38}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{n}(\hat{y}_i - y_i)^2}{K}} \tag{39}$$

$$\rho = \frac{\sum_{i=1}^{n}(\hat{y}_i - \mu_{\hat{y}})(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \mu_{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \tag{40}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\% \tag{41}$$

where n represents the total number of segments. The variables $\hat{y}_i$ and $y_i$ denote the output of the model and ground truth values in beats (respiration)-per-minute or number of steps for the $n^{th}$ segment, respectively. We reported the percentage for the Pearson correlation coefficient to save precision in the tables.

## C. Baselines

### C.1. Traditional-Heuristic based methods

**Fourier transform** We calculated the Fourier transform of signals and found the maximum amplitude sinusoidal from it to detect the periodicity. The length of transformation is set to 2048 for ECG signals and 512 for the rest. The frequency range of interest is defined according to common physiological limits for each task: [5, 40] *rpm* for respiration rate, [40, 140] for step counts per minute, and [30, 210] *bpm* for heart rate. We increased the length of the Fourier transform for ECG signals as their sampling frequency is approximately four times higher than the other signals. During the loss calculation of our proposed method, we use the same hyperparameters, frequency range of interests with length, of Fourier transformation.

**Autocorrelation** Similar to the Fourier-based method, we calculate the autocorrelation, as in Equation 42, of the signals and find the maximum value of $y_k$ in the range of frequency of interest.

$$y_k = \frac{\sum_{t=k+1}^{n}(x_t - \mu_x)(x_{t-k} - \mu_x)}{\sum_{t=1}^{n}(x_t - \mu_x)^2} \tag{42}$$

**RobustPeriod** RobustPeriod adopts maximal overlap discrete wavelet transform (MODWT) to decompose the input time series into multiple time series at different levels, ranks by wavelet variance, and then performs Huber periodogram with autocorrelation. We used Daubechies 10 wavelet with 4 levels, the lambda in the Hodrick–Prescott (HP) filter is set to 1e6, the Huber function hyperparameter is set to 2, and the M-Periodogram is set to 1.345. Although we searched for the optimum hyperparameters, the search space was limited due to the significant time consumption arising from the wavelet computation and optimization for each sample, given the numerous hyperparameters involved in the technique. Since this method includes autocorrelation to detect periodicity at each scale after the wavelet decomposition, the performance decreases severely when the desired period of interest is hidden by other periodic patterns in the same frequency band.

### C.2. Supervised

**DCL** We used a similar implementation of DeepConvLSTM architecture which is a 4-layer convolutional neural network with a kernel size of five, and ReLU activation after each layer, followed by a Dropout (Srivastava et al., 2014) and a two-layer LSTM with a hidden size being 128. We chose this architecture for the supervised and self-supervised learning paradigms as it was widely used before in the literature (Qian et al., 2022; Demirel & Holz, 2023), especially for detecting the periodicities in the supervised paradigm from signals (Biswas et al., 2019).

**CNN** We also implemented a fully convolutional neural network with a 3-layer followed by ReLU activation and MaxPooling after each convolutional layer. Dropout with 0.5 is applied after the first convolutional layer. We set the kernel and padding size to 8 and 4, respectively for each convolutional layer. The number of kernels for each convolutional layer is set to 32 for the first one and 64 for the rest.

We also performed a grid search for the hyperparameters of the architectures where we mainly investigated the number of convolutional layers with the kernel size. However, we did not observe a performance improvement with the increased number of parameters for the architectures as shown with the additional results in Appendix E.3.

### C.3. Self-Supervised

**NNCLR** We follow a similar setup to SimCLR by applying two separate data augmentations, then we use nearest neighbors in the learned representation space as the positive in contrastive losses (Dwibedi et al., 2021). The maximum size of the support set equals 1024.

**BYOL** For the BYOL implementation, the exponential moving average parameter is set to 0.996 where the projector size is set to 128. We set the learning rate to 0.03 similar to other SSL techniques. Following the original implementation, we use a weight decay parameter of $1.5e-6$.

**TS-TCC** We follow the same architecture implementation with the losses, i.e., contextual and temporal contrasting. TS-TCC proposed applying two separate augmentations, one augmentation is weak (jitter-and-scale) and the other is strong

(permutation-and-jitter). The authors also change the strength of the permutation window from dataset to dataset. In our experiments, we first used the original augmentations for each time series task, however, we observed performance decreases depending on the signal type. We, therefore, applied the specific augmentations for each time series, where we observed a general performance improvement in other SSL techniques as well.

**TS2Vec** TS2Vec (Yue et al., 2022) is a SSL method specifically designed for time series based on contrastive (instance and temporal wise) learning in a hierarchical way over augmented context views where the context is generated by applying timestamp masking and random cropping on the input time series. Following the original framework, we use a dilated CNN architecture with a depth of 10 and hidden size of 64, which has a similar number of parameters with the architectures used by other SSL methods. The batch size is set to 256, and the number of epochs to 120, consistent with other SSL techniques.

**VICReg** We follow the original implementation and set the coefficients for each loss term to 25 ($\lambda$), 25 ($\mu$), and 1 ($\nu$), corresponding to the invariance, variance, and covariance terms, respectively. Although we conducted a search for these loss term values, no performance enhancements were detected across the tasks.

$$\ell = \lambda \left[ s(z, z') \right] + \mu \left[ v(z) + v(z') \right] + \nu \left[ c(z) + c(z') \right], \tag{43}$$

where $s$ is the mean-squared Euclidean distance, $v$ is a hinge function on the standard deviation of the embeddings along the batch dimension, $c$ is the covariance regularization term as the sum of the squared off-diagonal coefficients

**Barlow Twins** Barlow Twins (Zbontar et al., 2021; Barlow, 2001) presents an objective function that naturally avoids collapse for SSL by measuring the cross-correlation matrix between the outputs of two identical networks fed with augmented versions of a sample, and making it as close to the identity matrix as possible. This causes the embedding vectors of augmented versions of a sample to be similar, while minimizing the redundancy between the components of these vectors. Following the original implementation, we applied batch normalization to the extracted embeddings and set the hyperparameter $\lambda$ coefficient (in Equation 44) to 0.005.

$$\mathcal{L} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2, \tag{44}$$

where $C$ is the cross-correlation matrix computed between the two sets of normalized embeddings.

**SimPer** SimPer is a simple contrastive self-supervised technique for learning periodic information in data. To exploit the periodic inductive bias, SimPer introduces customized augmentations and feature similarity measures. We follow the same augmentation in the original paper, and we downsample or interpolate the signals, which the authors call speed change. The range and the number of periodicity variant frequency augmentation are searched over [0.5, 3] and [3, 20], respectively, as in the original paper. We use the MXCorr for the similarity metric. Although the objective of the SimPer is the same as our work, the performance is extremely poor in all tasks and datasets compared to our proposed method even though the SimPer requires labels for training.

### C.4. No supervision

**SiNC** The three loss terms (bandwidth, sparsity, variance) with equal weights are implemented for signal estimation via the non-contrastive unsupervised learning (SiNC) technique. However, we adapted these limits individually for each task, following the specific frequency boundaries outlined in the Fourier transform.

## D. Implementation Details

Here, we have provided the details of the searched architectures, hyperparameters, and data augmentations to push the performance of the supervised and self-supervised techniques. To begin, we considered the model topologies that were used in previous works (Demirel & Holz, 2023; Qian et al., 2022; Biswas et al., 2019) and expanded upon them. We also evaluate the 1D Res-Net (Hong et al., 2020) implementation; however, we observe no significant performance improvement across datasets. This aligns with findings from previous works indicating that complex models may perform worse for time series data (He et al., 2023).

### D.1. Architectures

Here, we present the details of architectures that are investigated for the performance of learning-based algorithms. Mainly, we consider the baselines in the literature and increase the number of parameters approximately four times to observe if the performance increases. Batch normalization (Ioffe & Szegedy, 2015) is applied after each convolutional block, except the first version of DCL, similar to (Qian et al., 2022). ReLU activation is employed following batch normalization, in line with (He et al., 2016), with the exception of the final activation function in the U-Net, which is set to hyperbolic tangent. The number of parameters for each model is calculated for an input with 200 dimensions, i.e., a segment of the PPG signal.

*Table 7.* The model topologies for the combination of convolutional with LSTM-based networks. Table 7a) is the baseline architecture from previous works. Table 7b) is a similar model with $\approx$ 2x parameters.

(a) DeepConvLSTM Architecture

| Layer | Kernel Size | Output Size |
|---|---|---|
| Input (1 channel) | - | (T, 1) |
| Conv (64 kernels) | (5, 1) | (T-4, 64) |
| Conv (64 kernels) | (5, 1) | (T-8, 64) |
| Conv (64 kernels) | (5, 1) | (T-12, 64) |
| Conv (64 kernels) | (5, 1) | (T-16, 64) |
| Permute + Reshape | - | (64, T-16) |
| Dropout (p=0.5) | - | (64, T-16) |
| LSTM (2 layers, 128 units) | - | (128, T-16) |
| Output (if backbone) | - | (128,) |
| Linear (if not backbone) | - | (n_classes,) |
| # Parameters | | $\approx$316k |

(b) DeepConvLSTM_v2 Architecture

| Layer | Kernel Size | Output Size |
|---|---|---|
| Input (1 channel) | - | (T, 1) |
| Conv (64 kernels) | (7, 1) | (T-2, 64) |
| Conv (128 kernels) | (5, 1) | (T-4, 128) |
| Conv (128 kernels) | (5, 1) | (T-8, 128) |
| Conv (256 kernels) | (5, 1) | (T-10, 256) |
| Permute + Reshape | - | (256, T-10) |
| Dropout (p=0.5) | - | (256, T-10) |
| LSTM (2 layers, 128 units) | - | (128, T-10) |
| Output (if backbone) | - | (128,) |
| Linear (if not backbone) | - | (n_classes,) |
| # Parameters | | $\approx$494k |

*Table 8.* The model topologies for the fully convolutional neural networks. Table 8a) is the baseline architecture from previous works. Table 8b) is a similar model with ≈ 1.5x parameters.

(a) FCN Architecture

| Layer | Kernel Size | Output Size |
|---|---|---|
| Input (1 channel) | - | (T, 1) |
| Conv (32 kernels) | (8, 1) | (T, 32) |
| MaxPool | (2, 1) | (T/2, 32) |
| Dropout (p=0.2) | - | (T/2, 32) |
| Conv (64 kernels) | (8, 1) | (T/2, 64) |
| MaxPool | (2, 1) | (T/4, 64) |
| Conv (128 kernels) | (8, 1) | (T/4, 128) |
| MaxPool | (2, 1) | (T/8, 128) |
| Output (if backbone) | - | (128,) |
| Linear (if not backbone) | - | (n_classes,) |
| # Parameters | | ≈700k |

(b) FCN_v2 Architecture

| Layer | Kernel Size | Output Size |
|---|---|---|
| Input (1 channel) | - | (T, 1) |
| Conv (32 kernels) | (8, 1) | (T, 32) |
| Conv (32 kernels) | (8, 1) | (T, 32) |
| MaxPool | (2,1) | (T/2, 32) |
| Dropout (p=0.2) | - | (T/2, 32) |
| Conv (64 kernels) | (6, 1) | (T/2, 64) |
| Conv (64 kernels) | (6, 1) | (T/2, 64) |
| MaxPool | (2,1) | (T/4, 64) |
| Conv (128 kernels) | (4, 1) | (T/4, 128) |
| Conv (128 kernels) | (4, 1) | (T/4, 128) |
| MaxPool | (2, 1) | (T/8, 128) |
| Output (if backbone) | - | (128,) |
| Linear (if not backbone) | - | (n_classes,) |
| # Parameters | | ≈905k |

*Table 9.* 1D U-Net Architecture

| Layer | Kernel Size | Output Size | Input to Layer |
|---|---|---|---|
| Input | - | (T, 1) | — |
| AvgPool1 | (2,1) | (T/2, 1) | Input |
| AvgPool2 | (4.1) | (T/4, 1) | Input |
| Conv1D | (3, 1) | (T, 32) | Input |
| $\text{Conv1D}_1$ (down) | (3, 1) | (T/2, 64) | Conv1D |
| $\text{Conv1D}_2$ (down) | (3, 1) | (T/4, 96) | `concat`[$\text{Conv1D}_1$, AvgPool1] |
| $\text{Conv1D}_3$ (down) | (3, 1) | (T/8, 128) | `concat`[$\text{Conv1D}_2$, AvgPool2] |
| $\text{Upsample}_1$ | (2, 1) | (T/4, 128) | $\text{Conv1D}_3$ |
| $\text{Conv1D}_4$ (up) | (3, 1) | (T/4, 96) | `concat`[$\text{Upsample}_1$, $\text{Conv1D}_2$] |
| $\text{Upsample}_2$ | (2, 1) | (T/2, 96) | $\text{Conv1D}_4$ |
| $\text{Conv1D}_5$ (up) | (3, 1) | (T/4, 64) | `concat`[$\text{Upsample}_2$, $\text{Conv1D}_1$] |
| $\text{Upsample}_3$ | (2, 1) | (T, 64) | $\text{Conv1D}_5$ |
| $\text{Conv1D}_6$ (up) | (3, 1) | (T, 32) | `concat`[$\text{Upsample}_3$, Conv1D] |
| Out Conv | (3, 1) | (T, 1) | $\text{Conv1D}_6$ |
| # Parameters | | ≈355k | |

### D.2. Augmentations

In this section, we give details about the data augmentations that are applied to the time series tasks. During our experiments, we searched for the best traditional augmentation technique for a given task. We searched over common time series augmentation methods in literature (Table 10), and applied them with self-supervised learning baselines.

Specifically, we applied *scaling* for ECG and IMU signals to detect heart rate and step counting, *permutation with noise* for HR detection from PPG signals. *scaling* and *permutation with noise* for respiratory rate from PPG. We also observed that while permutation enhances the performance of PPG signals, it significantly diminishes the performance of ECG signals.

*Table 10.* Common time series augmentations

| Augmentation | Details |
|---|---|
| Noise | Add Gaussian noise sampled from normal distribution, $\mathcal{N}(0, 0.3)$ |
| Scale | Amplify channels by a random distortion sampled from normal distribution $\mathcal{N}(2, 1.1)$ |
| Negate | Multiply the value of the signal by a factor of -1 |
| Permute | Split signals into no more than 5 segments, then permute the segments and combine them into the original shape |
| Resample | Interpolate the time series to 3 times its original sampling rate and randomly down-sample to its initial dimensions |
| Time Flip | Flip the time series in time for all channels, i.e., $\mathbf{x}_{Aug}[n] = \mathbf{x}[-n]$ |
| Random Zero Out | Randomly choose a section to zero out (at most 1/10 of the signal) |
| Permutation + Noise | Combination of Permutation and Noise with the noted parameters |
| Noise + Scale | Combination of Noise and Scaling |

SimPer applies interpolation and decimation to change the periodicity of the sample as augmentations. We limit the speed change—the term authors used for resampling the short and long samples—range to be within [0.5, 2], ensuring the augmented sequence is longer or shorter than a fixed length in the time dimension, following the same implementation in the paper.

# E. Additonal Results

## E.1. BIDMC dataset

The BIDMC dataset exhibits a significant imbalance (Pimentel et al., 2017), leading self-supervised and supervised learning techniques to essentially leverage the training data statistics, particularly the mean output, during inference. We, therefore, only report the heuristic and unsupervised learning-based methods in Table 11. As seen in Table 11, our proposed method

*Table 11.* Comparison of methods in *Respiration* for BIDMC

| Method | BIDMC (64-second) | | BIDMC (32-second) | |
|---|---|---|---|---|
| | MAE↓ | RMSE↓ | MAE↓ | RMSE↓ |
| *Heuristic* | | | | |
| Fourier | 4.081 | 4.962 | 5.71 | 7.25 |
| Autocorrelation | 14.05 | 16.23 | 14.36 | 16.64 |
| RobustPeriod | 10.84 | 13.36 | 10.02 | 12.79 |
| *No Supervision* | | | | |
| SiNC | 4.11±0.10 | 5.44±0.31 | 6.01±0.98 | 7.33±0.92 |
| Ours | **3.40**±0.20 | **4.41**±0.43 | **3.74**±0.03 | **4.77**±0.12 |

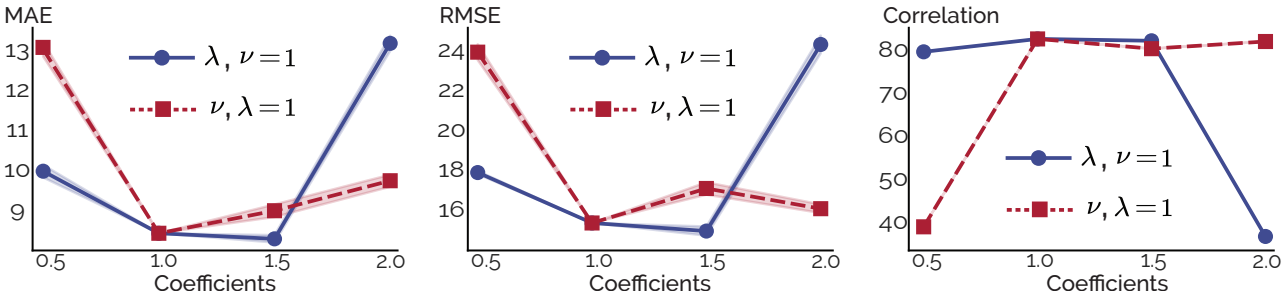
demonstrates a significant performance improvement over the baselines. Furthermore, the performance gap between our method and another deep learning solution, SiNC, exceeds 25%.

## E.2. The effect of loss weights

In this section, we investigate the effect of loss weights. During our experiments, we give the same importance to each regularizer, i.e., the coefficients ($\lambda$, $\nu$) in Equation 45 are set to 1.

$$\mathcal{L} = -\lambda \sum_{f^*} S(\omega) \log S(\omega) + \nu \sum_{f^*} S_X(\omega) \log \frac{S_X(\omega)}{S(\omega)} + \sum_{f'} S(\omega), \quad \omega = 2\pi f \text{ and } f' = \mathbb{U} \setminus f^* \tag{45}$$

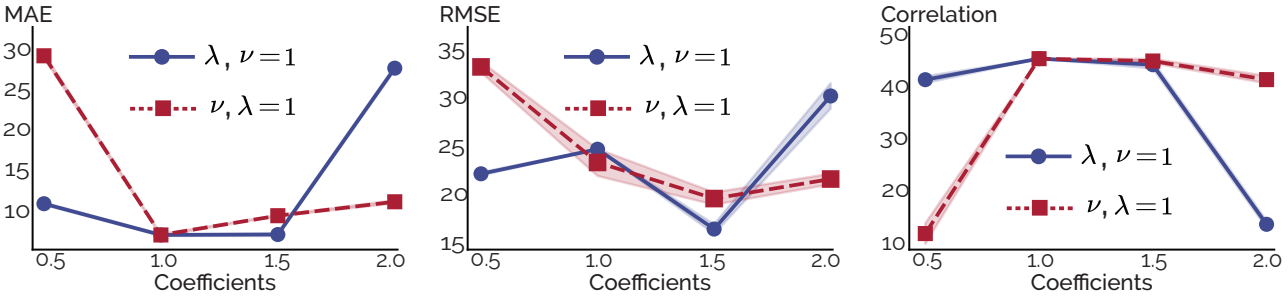
We observed the effect of two coefficients on the performance by setting them to discrete values of $\{0.5, 1.0, 1.5, 2.0\}$, independently. The below figures present the results for two different signals, including electrocardiogram and photoplethysmography. It is evident from these figures that the optimal performance is achieved when the weights for both losses are balanced, rather than assigning greater importance to one over the other. Moreover, a closer inspection of these figures reveals an important outcome regarding the losses. When the models are trained while giving more importance to decreasing spectral entropy ($\lambda$) rather than keeping the relative entropy between input and output close ($\nu$), they collapse. This is noticeable when examining the blue line—increasing the importance of periodicity with the same weight for the other loss—the model error increases significantly.




(a) Mean absolute error on IEEE SPC12 (b) Root mean square error on IEEE SPC12 (c) Correlation on IEEE SPC12

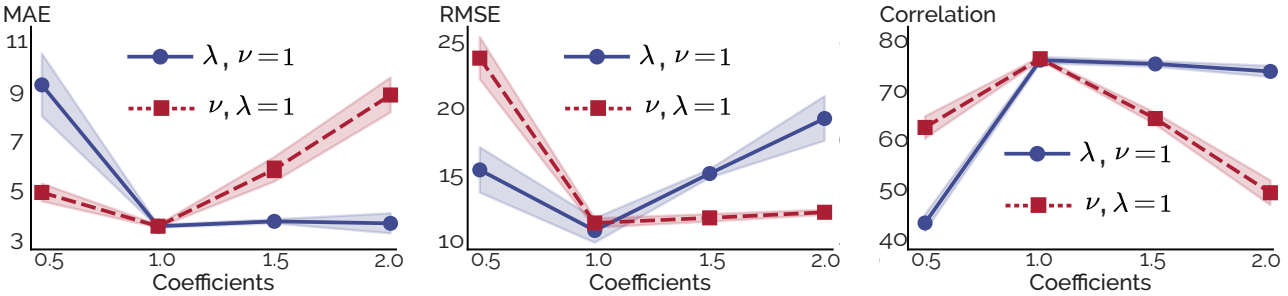
*Figure 5.* The three error metrics——mean absolute error (MAE), root mean square error (RMSE), and correlation——provided in the SPC12 dataset are used to evaluate the sensitivity of loss weights on the overall performance.

Moreover, this effect is more pronounced with the datasets that include more noisy samples. For example, the mean absolute error increases by twice when more weight is given to periodicity as in Figure 6.




(a) Mean absolute error on IEEE SPC22 (b) Root mean square error on IEEE SPC22 (c) Correlation on IEEE SPC22

*Figure 6.* The three error metrics——mean absolute error (MAE), root mean square error (RMSE), and correlation——provided in the SPC22 dataset are used to evaluate the sensitivity of loss weights on the overall performance.




(a) Mean absolute error on PTB-XL (b) Root mean square error on PTB-XL (c) Correlation on PTB-XL

*Figure 7.* The three error metrics——mean absolute error (MAE), root mean square error (RMSE), and correlation——provided in the PTB-XL dataset are used to evaluate the sensitivity of loss weights on the overall performance.

When investigating the model performance with varying loss weights on two signals, we can see the different behavior of the model. Specifically, in less noisy signal segments such as ECG, increasing the importance of the periodicity loss does not lead to substantial performance declines when compared with PPG signals. This phenomenon can be attributed to the distinct and clear periodic patterns, such as QRS complexes, present in the ECG signal. In contrast, the periodicity in blood volume changes is less apparent, hidden beneath higher noise levels, leading the model to capture more noise rather than the desired periodicity when the weights are imbalanced.

### E.3. The effect of architectures

Here, we present the performances of the baseline architectures that are explicitly given in Appendix D for each dataset to investigate the effect of model capacity on generalization. An interesting result from these tables is that even though the number of parameters increases for architectures, the overall performance decreases for some datasets. For example, the second version of the fully convolutional neural network has roughly twice the number of parameters when compared with DCL. However, it performs worse than the DCL in cases such as PTB-XL and DaLiA. Although we applied this architecture search for the self-supervised learning paradigm as well, we have not observed a performance change across datasets.

*Table 12.* Performance comparison of ours with other methods in *ECG* datasets for HR detection

| Method | PTB-XL | | | DaLiA$_{ECG}$ | | | WESAD | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| *Supervised* | | | | | | | | | |
| DCL | 6.08±0.78 | 14.23± 1.43 | 76.40±6.02 | 3.91±0.37 | 6.96±1.06 | 91.42±2.5 | 9.14±0.78 | 7.23±0.75 | 85.44±6.32 |
| DCL$_{v2}$ | 5.65±0.19 | 18.13± 1.31 | 66.22±0.79 | 2.41±0.06 | 4.91±0.10 | 95.64±0.13 | 2.80±0.09 | 4.49±0.11 | 95.13±0.59 |
| CNN | 9.09±0.27 | 16.37±0.33 | 66.11±1.75 | 5.39±0.16 | 8.36±0.15 | 89.44±0.72 | 10.69±0.67 | 13.57±1.01 | 50.10±12.5 |
| CNN$_{v2}$ | 7.50±0.45 | 17.36±1.13 | 62.13±5.90 | 4.02±0.15 | 6.95±0.37 | 91.87±1.01 | 4.98±0.06 | 7.27±0.19 | 87.98±0.66 |
| U-Net | 11.36±0.50 | 19.33±0.70 | 62.44±2.56 | 4.72±0.03 | 10.88±0.07 | 80.76±0.10 | 5.39±0.02 | 7.75±0.06 | 87.77±0.31 |
| *No Supervision* | | | | | | | | | |
| Ours | **3.75**±0.02 | **13.58**±0.03 | **79.30**±0.17 | **1.29**±0.001 | **2.08**±0.01 | **99.12**±0.01 | **2.13**±0.01 | **3.88**±0.06 | **93.52**±0.61 |

*Table 13.* Comparison of methods in *Step counting*

| Method | Clemson (Regular) | | Clemson (Semi-regular) | |
|---|---|---|---|---|
| | MAPE↓ | MAE↓ | MAPE↓ | MAE↓ |
| *Supervised* | | | | |
| DCL | 5.99±0.34 | 3.45±0.32 | 19.59±1.54 | 8.98±0.87 |
| $DCL_{v2}$ | **5.64**±0.59 | **3.23**±0.33 | 18.73±0.54 | 7.90±0.35 |
| FCN | 6.15±0.60 | 3.53±0.33 | 17.22±0.31 | 6.98±0.23 |
| $FCN_{v2}$ | 5.97±0.26 | 3.48±0.15 | **15.24**±0.57 | **6.44**±0.28 |
| U-Net | 6.24±0.53 | 3.62±0.32 | 16.83±0.45 | 7.71±0.26 |
| *No Supervision* | | | | |
| Ours | 5.95±0.21 | 3.42±0.17 | 35.18±0.02 | 14.21±0.97 |

*Table 14.* Comparison of methods in *Respiration*

| Method | CapnoBase (64-second) | | CapnoBase (32-second) | |
|---|---|---|---|---|
| | MAE↓ | RMSE↓ | MAE↓ | RMSE↓ |
| *Supervised* | | | | |
| DCL | 5.76±0.28 | 7.45±0.27 | 5.74±0.08 | 7.68±0.07 |
| $DCL_{v2}$ | 6.28±0.37 | 8.61±0.71 | 6.30±0.80 | 8.71±1.04 |
| FCN | 6.00±0.20 | 8.15±0.28 | 5.41±0.24 | 7.57±0.40 |
| $FCN_{v2}$ | 5.49±0.21 | 7.65±0.38 | 5.63±0.60 | 7.73±0.70 |
| U-Net | 5.49±0.21 | 7.65±0.38 | 5.63±0.60 | 7.73±0.70 |
| *No Supervision* | | | | |
| Ours | **3.40**±0.20 | **4.41**±0.43 | **3.74**±0.03 | **4.77**±0.12 |

*Table 15.* Performance comparison of supervised methods with different architectures in *PPG* datasets for HR estimation

| Method | IEEE SPC12 | | | IEEE SPC22 | | | $DaLiA_{PPG}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| *Supervised* | | | | | | | | | |
| DCL | 19.90±1.10 | 26.34±1.10 | 25.53±4.2 | 22.43±0.62 | 27.17±0.60 | 11.95±5.1 | 5.97±0.19 | 11.83±0.36 | 80.41±0.9 |
| $DCL_{v2}$ | 11.05±0.08 | 14.64±0.10 | 77.40±0.71 | 18.89±0.44 | 22.89±0.26 | 09.90±5.1 | **5.66**±0.06 | **11.29**±0.21 | **81.99**±0.46 |
| CNN | 12.48±0.23 | 18.19±0.26 | 68.08±1.85 | 17.97±0.33 | 23.06±0.36 | 21.91±1.76 | 7.35±0.14 | 13.74±0.26 | 74.22±0.55 |
| $CNN_{v2}$ | 12.10±0.22 | 18.19±0.26 | 68.08±1.85 | 18.12±0.37 | 22.90±0.39 | 19.83±3.14 | 6.89±0.02 | 13.02±0.12 | 76.60±0.31 |
| U-Net | 18.40±0.79 | 25.06±1.05 | 39.53±4.76 | 25.86±0.75 | 31.45±0.83 | 06.74±1.76 | 8.70±0.21 | 16.76±0.34 | 65.61±1.11 |
| *No Supervision* | | | | | | | | | |
| Ours | **9.30**±0.10 | **16.50**±0.20 | **77.60**±0.43 | **10.27**±0.37 | **19.62**±0.71 | **44.10**±0.89 | 27.41±4.73 | 31.26±4.55 | 18.12±3.86 |

### E.4. Comparison with additional traditional methods

Here, we compared our approach with a traditional wavelet-based method (Martinez et al., 2004) specifically designed for detecting the QRS complex in ECG signals, as QRS complexes represent the main source of periodicity in the ECG (Pan & Tompkins, 1985), which aligns with the periodicity of interest in our experiments for the ECG signals. The comparison results are given in Table 16.

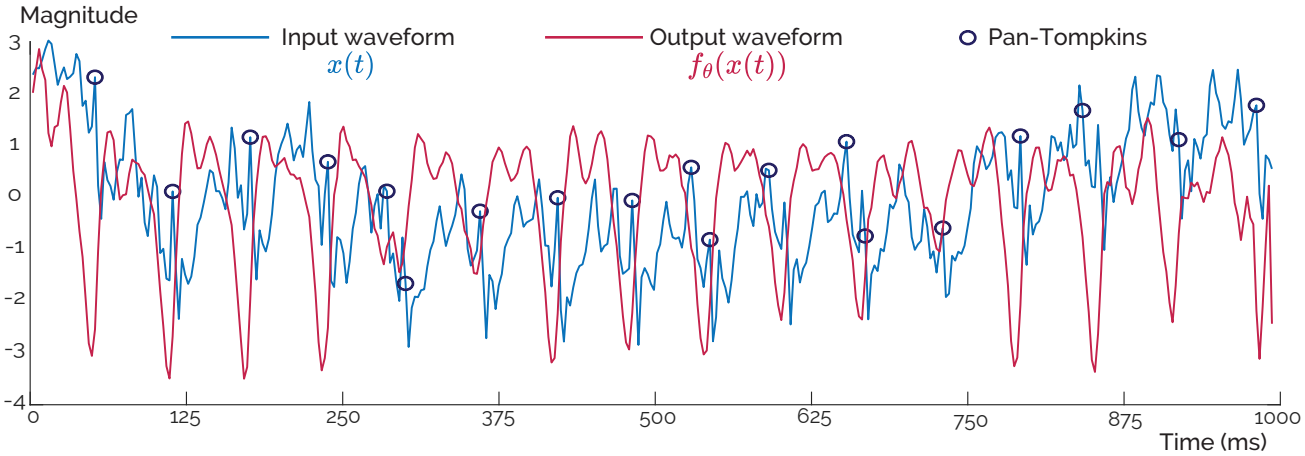
*Table 16.* Performance comparison of ours with other methods in *ECG* datasets for HR detection

| Method | PTB-XL | | | $DaLiA_{ECG}$ | | | WESAD | | |
|---|---|---|---|---|---|---|---|---|---|
| | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ | MAE↓ | RMSE↓ | $\rho$↑ |
| *Heuristic* | | | | | | | | | |
| Fourier | 10.51 | 28.65 | 49.12 | 4.06 | 15.46 | 36.86 | 4.88 | 18.57 | 56.69 |
| Autocorrelation | 8.64 | 19.93 | 59.63 | 7.17 | 14.07 | 38.17 | 4.19 | 8.046 | 89.90 |
| RobustPeriod | 72.79 | 80.87 | -2.93 | 12.79 | 23.21 | 37.49 | 4.19 | 7.130 | 62.69 |
| Wavelet-based | 10.06 | 22.85 | 50.73 | 5.41 | 10.94 | 90.52 | 9.47 | 23.19 | 70.45 |
| Ours | **3.75**±0.02 | **13.58**±0.03 | **79.30**±0.17 | **1.29**±0.001 | **2.08**±0.01 | **99.12**±0.01 | **2.13**±0.01 | **3.88**±0.06 | **93.52**±0.61 |

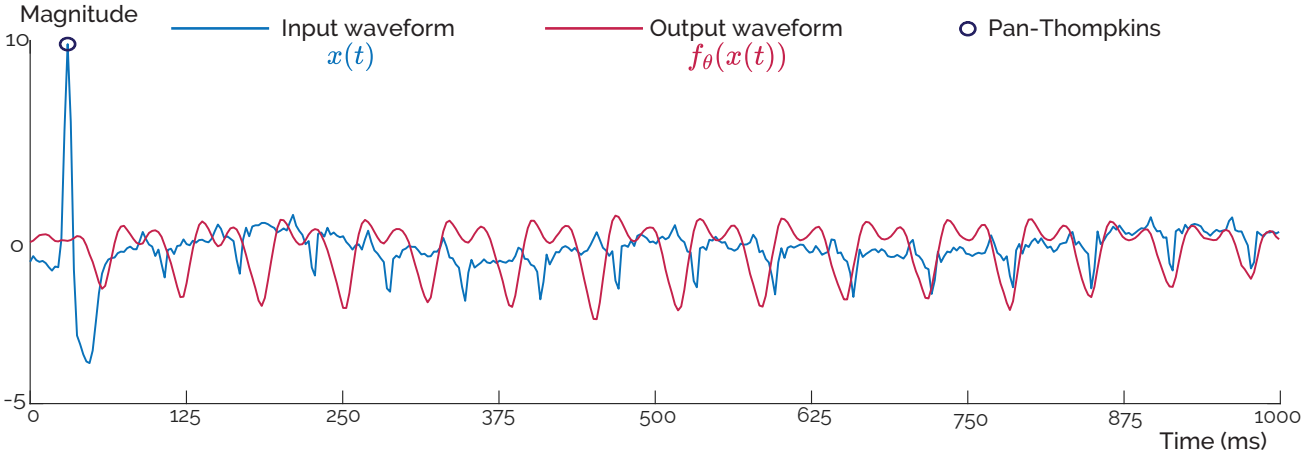
As shown in Table 16, our proposed method outperforms traditional methods by a significant margin. we believe that the main reason for this performance difference is the fact that these methods are often developed by optimizing numerous hyperparameters specific to the dataset statistics, making them less effective in a general case.

## F. Visual Examples

Here, we give some examples where the heuristic rule-based methods fail to detect periodicities in signals due to various types of noise. First, we show the cases when our method performs better than the Pan-Tompkins (Pan & Tompkins, 1985) algorithm for periodicity detection in ECG signals.




(a) A 10-second ECG recording from PTB-XL, noise is present with numerous spikes at the same amplitude as the QRS complex, leading to the Pan-Tompkins algorithm detecting multiple beats at the same location.

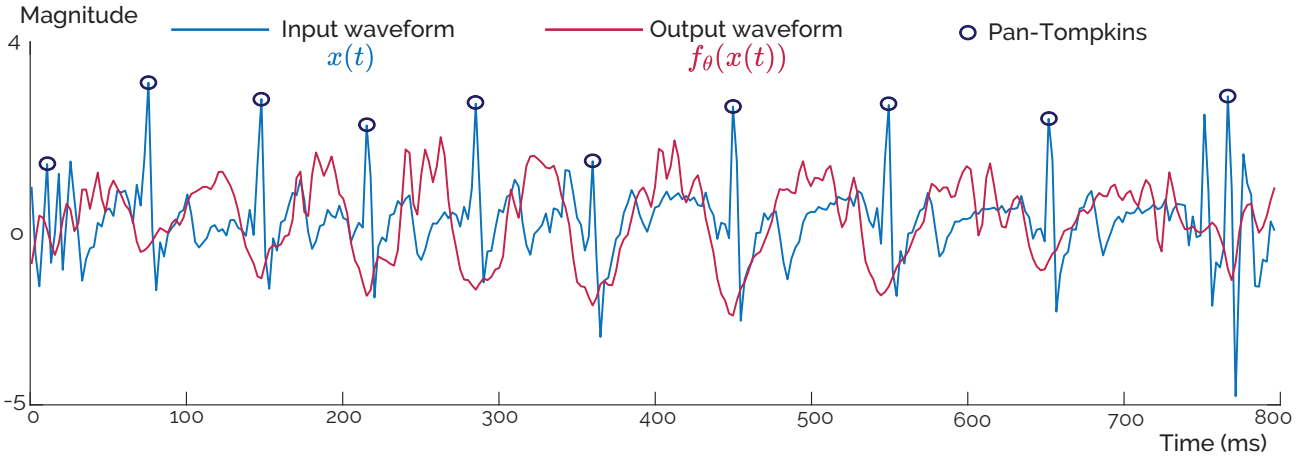



(b) A recording from PTB-XL that contains an abrupt impulse noise at the beginning, which causes Pan-Tompkins to fail.

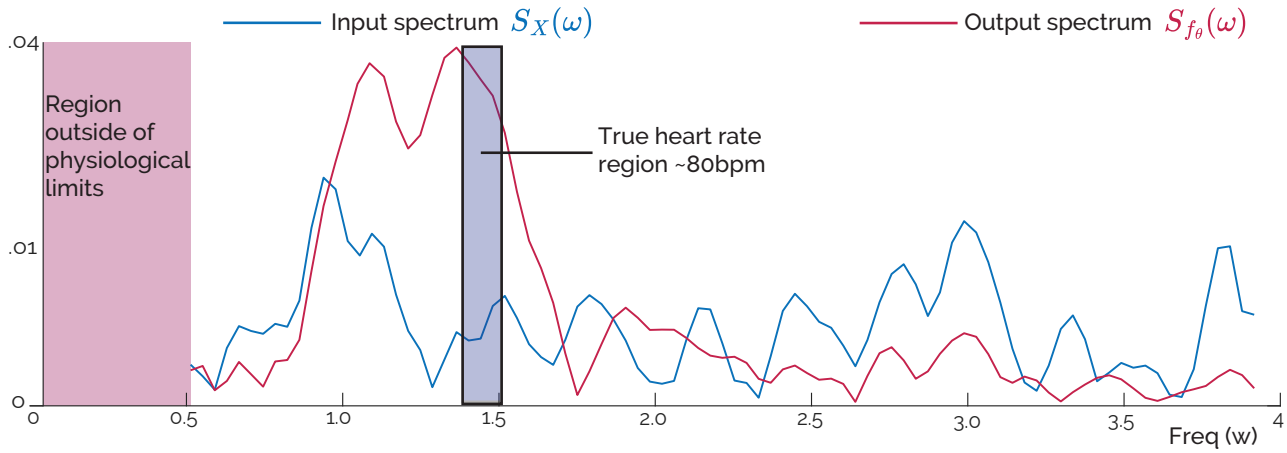
*Figure 8.* The visual examples for ECG signals with the raw signal $\mathbf{x}(t)$, the output of the model $f_\theta(\mathbf{x}(t))$, and the detected points using Pan-Tompkins algorithm. Despite its common use for identifying periodic patterns, QRS complexes, in ECG signals, we present two examples where the Pan-Tompkins algorithm fails due to various types of noise in the recordings.

One interesting outcome of these illustrations is that the model demonstrates the ability to entirely disregard the impulse noise at the beginning (See Figure 8a), and effectively approximate the periodicity in the signal. Moreover, a closer inspection of these two figures shows that the proposed method also works as an unsupervised segmentation algorithm of signals. For example, the end of Figure 8 demonstrates that the output waveform matches the desired patterns. However, when the time series becomes non-stationary in the segment, the segmentation shifts as the proposed algorithm assumes the signals are periodic or quasi-periodic in the investigated segment. This assumption is widely used in algorithm development for the human-generated data (Demirel & Holz, 2023), such as signals generated from the cardiovascular (ECGs, PPGs) or skeletal system (IMUs) of humans similar to our case. Therefore, we believe that the proposed algorithm can be applied to segmentation tasks, enabling the identification of the desired periodic events in a completely unsupervised manner.

Here, we provided an example of how the proposed method is similar to an adaptive sinusoidal selector based on the Fourier transform. In uniformly sampled time sequences, periodicity is typically identified by the highest value in the spectral density (Li et al., 2010). Nevertheless, this method proves ineffective in situations where multiple periodicities exist within the frequency range of interest or noise, as illustrated in Figure 9.




(a) An 8-second ECG recording from DaLiA $\mathbf{x}(t)$, and the output waveform from the trained model $f_\theta(\mathbf{x}(t))$. It can be seen that the recording is noisy at the beginning and the end where the Pan-Tompkins algorithm confuses true periodic patterns with noise in the signal while giving a good approximation of the heart rate.

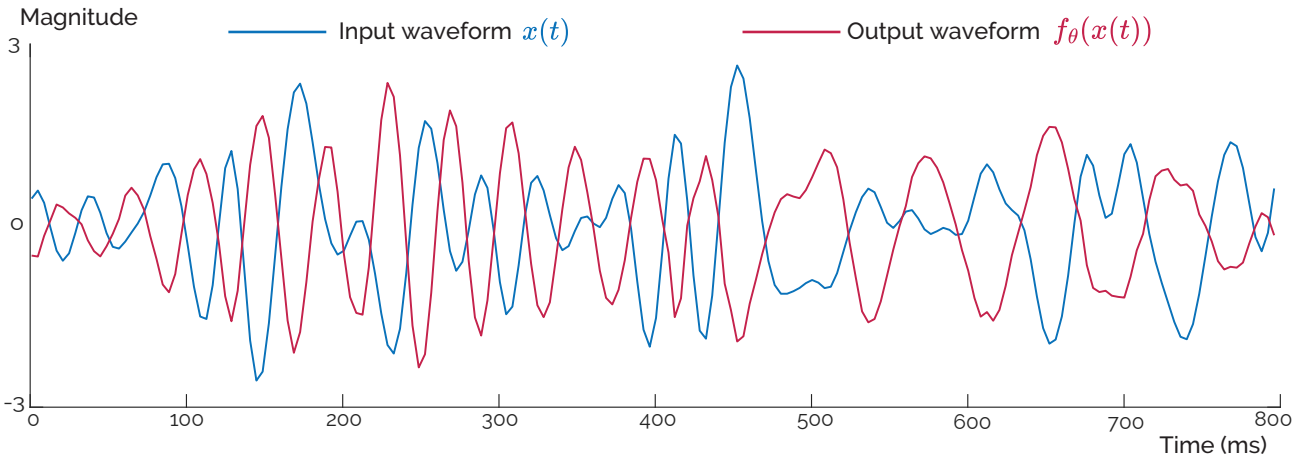



(b) The spectra of the input and output signal for comparison. The frequencies lower than 0.5 are shown with red as they correspond to the heart rate outside of physiological limits, i.e., lower than 30 beats per minute. The input spectrum shows the main periodic signal is at around 1 Hz due to corrupted waveform, especially at the beginning and end. A rule-based algorithm depending on the input spectrum may incorrectly identify the desired periodicity due to inaccuracies in the peak location. However, the spectrum of the output waveform increased the magnitude of the sinusoidal at the desired periodicity region significantly, close to a 7 dB increase. More importantly, the output waveform includes the desired pattern as the dominant peak rather than noisy ones, which allows the detection of targeted periodicity in the signal.

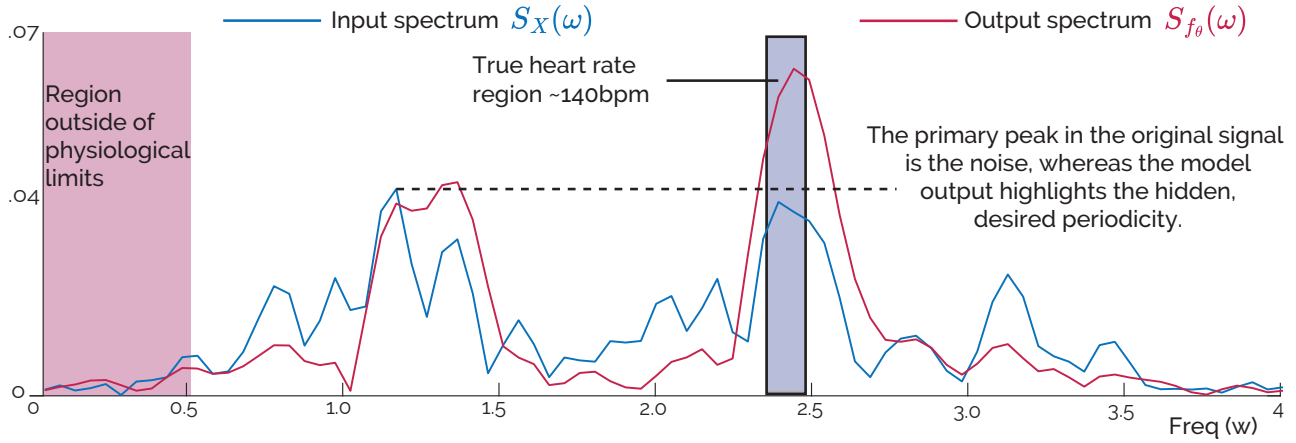
*Figure 9.* The visual examples for ECG signals with the raw signal $\mathbf{x}(t)$, the output of the model $f_\theta(\mathbf{x}(t))$, with their corresponding spectral densities. The figures show how the trained model can select and amplify the desired pattern even though it is under the noise level.

According to these two figures, we can infer that the model learns the periodic representations of interest from the time domain waveform such that when it is evaluated with a signal that the model has never seen before, it can detect these patterns and surpass the noise. The task becomes relatively simpler for ECG signals due to their distinct periodicity indicator, the QRS complex. This allows the model to learn and effectively ignore impulse noise or other periodic noise sources even though they are both within the frequency range of interest.

Here, we present a more challenging example involving signals generated by blood volume changes. These signals are susceptible to noise and exhibit less evident periodicity, lacking specific complexes like ECG signals.




(a) A 8-second PPG recording from IEEE SPC $\mathbf{x}(t)$, and the output waveform from the trained model $f_\theta(\mathbf{x}(t))$. The input waveform is so noisy that the true volumetric blood changes are not observable at all. It can be seen from the below figure that the desired periodicity is under the noise level.




(b) The spectra of the input and output blood volume changes for comparison. The frequencies lower than 0.5 Hz are shown with red as they correspond to the heart rate outside of physiological limits, i.e., lower than 30 beats per minute. Similarly, we limit the highest frequency to 4 Hz. The input spectrum shows the main periodic signal is at around 1.2 Hz due to the corrupted signal. Similar, to the ECG case, a rule-based algorithm depending on the input spectrum may incorrectly identify the desired periodicity due to inaccuracies of the dominant peak in the power spectrum. The model outputs a waveform that contains the amplified sinusoidal at the desired periodicity while increasing the overall signal-to-noise ratio.

*Figure 10.* The visual examples for photoplethysmogram signals with the raw waveform $\mathbf{x}(t)$, and the output of the model $f_\theta(\mathbf{x}(t))$. A waveform example where the Fourier transformation-based periodicity detection fails because of a noisy periodic pattern at the same frequency band. In contrast, the proposed model effectively learns and amplifies the pattern of interest for accurate detection.

What stands out in these two figures is that the model learns the pattern of the desired periodic waveform from the time domain during training such that it can detect these patterns during inference (even though they are under the noise level) and surpass the other periodic noise.

These illustrative examples further support the claim that the proposed method operates as an adaptive sinusoidal selector by learning the periodic source of interest during training and amplifying them during inference without the need for supervision. We believe the results can be easily improved more by selecting clean samples during training, i.e., curriculum learning, to reinforce the pattern learning, or incorporating physiological limits more effectively into the loss function.

## G. Computational Efficiency

In this section, we provide details on the computational efficiency of our proposed method. We employed a U-Net architecture with 300K parameters, making it lightweight and suitable for deployment on various platforms, including mobile devices. For context, MobileNetV2, often considered a lightweight model in deep learning, has 3.4M parameters, which is approximately 10 times larger than our model.

To provide a comprehensive comparison, we conducted two types of model overhead analyses using the blood volume pulse signals as input similar to the parameter calculations in Appendix D.1. First, we calculated the inference time on a computer equipped with an Intel Core i7-10700k CPU running at 3.80 GHz, 32 GB RAM. Second, we deployed our model on a MAX78000 AI Accelerator, which has previously implemented U-Net architectures (Moss et al., 2023). Detailed performance metrics and comparisons with the average of five runs with the standard deviations are provided in Table 17 where the running time is given in milliseconds.

*Table 17.* Memory and inference time comparison

| **Method** | MAX78000 | | PC | |
|---|---|---|---|---|
| | Memory Requirement | Inference Time | Memory Requirement | Inference Time |
| FFT | 2.8±0.1 KB | 0.05±0.02 | 1.92±0.2 KB | 0.10±0.03 |
| Wavelet | Out-of-Memory | N/A | 10.4±1.2 MB | 41.1±3.6 |
| Robust Period | Out-of-Memory | N/A | 24.7±3.6 MB | 675±30.2 |
| Ours | 19.1±0.7 KB | 26.9±0.23 | 16.4±1.4 KB | 3.2±0.7 |

An important aspect to note is that PyTorch (or MATLAB) efficiently computes the FFT; when analyzed with the profiler, the memory usage is close to that of the input (in-memory computation). However, for the MAX78000, our implementation, instead of DSP, requires slightly more memory. Overall, as shown by these experiments, our method not only offers better solutions for periodicity detection but also emphasizes practical implementation by having a very low memory footprint with fast inference. Moreover, some traditional methods do not fit the MCU flash memory due to heavy wavelet computations.