Denoising-Aware Contrastive Learning for Noisy Time Series

Shuang Zhou[1], **Daochen Zha**[2], **Xiao Shen**[3], **Xiao Huang**[1*], **Rui Zhang**[4] and **Korris Chung**[1]
[1]Department of Computing, The Hong Kong Polytechnic University
[2]Department of Computer Science, Rice University
[3]School of Computer Science and Technology, Hainan University
[4]Department of Surgery, University of Minnesota
shuang.zhou@connect.polyu.hk, daochen.zha@rice.edu, shenxiaocam@163.com,
zhan1386@umn.edu, {xiaohuang, cskchung}@comp.polyu.edu.hk

Abstract

Time series self-supervised learning (SSL) aims to exploit unlabeled data for pre-training to mitigate the reliance on labels. Despite the great success in recent years, there is limited discussion on the potential noise in the time series, which can severely impair the performance of existing SSL methods. To mitigate the noise, the de facto strategy is to apply conventional denoising methods before model training. However, this pre-processing approach may not fully eliminate the effect of noise in SSL for two reasons: (*i*) the diverse types of noise in time series make it difficult to automatically determine suitable denoising methods; (*ii*) noise can be amplified after mapping raw data into latent space. In this paper, we propose denoising-aware contrastive learning (DECL), which uses contrastive learning objectives to mitigate the noise in the representation and automatically selects suitable denoising methods for every sample. Extensive experiments on various datasets verify the effectiveness of our method. The code is open-sourced.

1 Introduction

Time series learning has attached great importance in various real-world applications [Ismail Fawaz *et al.*, 2019], such as heart failure diagnosis and fault detection in the industry. Given the abundance of unlabeled time series data [Meng *et al.*, 2023b], there has been a surge in attention towards time series self-supervised learning (SSL) that extracts informative representations from unlabelled time series data for downstream tasks [Ma *et al.*, 2023]. Many time series SSL methods have been proposed in recent years [Zhang *et al.*, 2023a], including contrastive learning-based [Tonekaboni *et al.*, 2021], generative-based [Chowdhury *et al.*, 2022], and adversarial-based [Luo *et al.*, 2019] approaches.

Despite the encouraging progress made in time series SSL, the existing research often assumes that the given time series is clean, with limited discussion on the potential noise in the time series. Unfortunately, many time series (e.g., bio-signals collected from sensors) naturally suffer from noises that can severely change the data characteristics and impair representation learned by SSL algorithms [Zhang *et al.*, 2023b]. For instance, some samples in the ECG dataset PTB-XL [Wagner *et al.*, 2020] exhibit considerable high-frequency noise, depicted in Fig. 1(a). Directly applying the existing time series SSL methods CA-TCC [Eldele *et al.*, 2023] and TS2vec [Yue *et al.*, 2022] on this dataset for the classification task yields poor accuracy, as illustrated on the left in Fig.1(c). In contrast, by employing an appropriate denoising method like LOESS [Burguera, 2018], as illustrated in Fig.1(b), the accuracy experiences a significant improvement, as shown in the middle of Fig.1(c). Motivated by this, we study the following research question: *How can we effectively denoise noisy time series for SSL to learn better representations?*

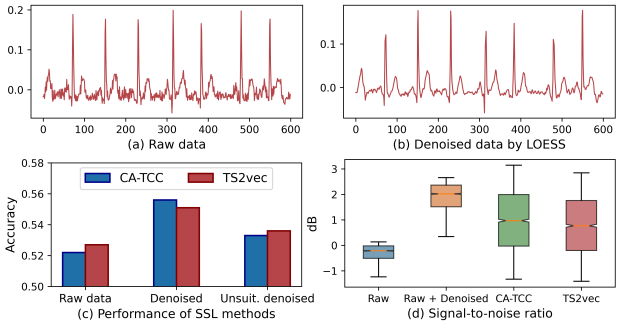

Figure 1: A motivating analysis (more details provided in Appendix). (a-c) show that the SSL methods achieve higher performance after pre-processing the noisy time series in the PTB-XL dataset with a suitable denoising method LOESS, while the performance improvement is not obvious when processed by unsuitable methods like median filter. (d) suggests SSL methods tend to amplify the noise in the representation.

The de facto strategy to handle noises in time series is to apply conventional denoising methods (e.g., LOESS mentioned above) and then perform model learning [Lai *et al.*, 2023]. However, we argue that this pre-processing approach cannot fully eliminate the effect of noise in time series. Firstly, the diverse types of noise in time series make it challenging to select the most suitable denoising methods [Zhang *et al.*, 2021a]. Many real-world datasets, e.g., ECG data, may contain thousands of samples, where each sample involves

*Corresponding author

different noises [He *et al.*, 2015], such as high-frequency noise, baseline wandering, and muscle artifacts [Zhang *et al.*, 2021b]. There is often no denoising method that can universally handle all types of noise [Robbins *et al.*, 2020; Zheng *et al.*, 2020], making it difficult to select suitable denoising methods in real-world applications. For example, on the right-hand side of Fig.1(c), the accuracy drops when we apply an unsuitable denoising method median filter. Secondly, noise can be amplified after mapping the raw data into latent space. To illustrate this, we compare the signal-to-noise ratio (SNR) [Chawla, 2011] of the raw time series, the denoised time series, and the representations learned upon the denoised time series in Fig. 1(d). The results show that (*i*) the SNR value of the denoised data is improved, meaning that the noise is mitigated in the raw data; (*ii*) however, if further mapping the denoised data into representations, the SNR value drops, which suggests that the representation learning process could amplify the noise. That is, even though the denoising methods can alleviate noise in the raw time series, the noise could "come back" in the representation space and still hamper SSL. Hence, how to mitigate the noise in time series SSL remains an open challenge.

In this paper, we propose DEnoising-aware Contrastive Learning (DECL), an end-to-end framework that can leverage any conventional denoising methods to guide noise mitigation in representations. Specifically, DECL involves two novel designs. Firstly, building upon an auto-regressive encoder, we propose a novel denoiser-driven contrastive learning objective to mitigate the noise. The key idea is to construct positive samples through the application of existing denoisers on raw time series, and concurrently generating negative samples by introducing noise into the same time series. Subsequently, through optimization using a contrastive learning objective, we guide the representations toward positive samples and distance them from negative samples, thereby reducing the noise. Secondly, we introduce an automatic denoiser selection strategy to learn to select the most suitable denoisers for each sample. Our motivation is that, in auto-regressive learning, noisy data usually tend to have large reconstruction errors, and vice versa. As a result, we can use the reconstruction error as a proxy for how suitable a denoiser is to the sample. We further incorporate this denoiser selection strategy into the proposed denoiser-driven contrastive learning and optimize them jointly. We summarize our contribution below.

- **Problem:** Motivated by the observation that removing noise boosts the performance of SSL, we formulate the problem of self-supervised learning on noisy time series.
- **Algorithm:** We propose a customized denoising-aware contrastive learning method [1] that automatically selects suitable denoising methods for each sample to guide mitigating data noise in representation learning.
- **Experimental Findings:** Extensive experiments show the effectiveness of our method. We also verify that DECL is robust with varying degrees of noise and the learned representations have less noise.

[1] https://github.com/betterzhou/DECL

2 Problem Statement

Self-Supervised Learning on Noisy Time Series. Given a set of time series $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ of N instances with a certain amount of noise $\mathcal{S}$, the goal is learning a nonlinear mapping function $\mathcal{F}$ in a self-supervised manner that maps each time series $\boldsymbol{x}_i$ to a representation $\mathbf{z}_i$ to best describe itself. Specifically, for a time series sample $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,T}] \in \mathbb{R}^{T\times d}$ with T timestamps and d feature dimensions, it involves noise $\mathbf{s}_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,T}] \in \mathbb{R}^{T\times d}$ and the denoised data is present as $\boldsymbol{v}_i = [(x_{i,1} - s_{i,1}), (x_{i,2} - s_{i,2}), \ldots, (x_{i,T} - s_{i,T})]$; the mapping function $\mathcal{F}$ aims to learn a representation $\mathbf{z}_i = [z_{i,1}, z_{i,2}, \ldots, z_{i,C}] \in \mathbb{R}^{C\times r}$, where $z_{i,t} \in \mathbb{R}^r$ is representation at timestamp t with r dimensions.

3 Methodology

In this section, we present the proposed DEnoising-aware Contrastive Learning (DECL), as shown in Fig. 2. It consists of three components: (*i*) auto-regressive learning, which is for generating informative representations in latent space; (*ii*) denoiser-driven contrastive learning, which exploits denoising methods to guide mitigating noise in representation learning; (*iii*) automatic denoiser selection, which selects suitable denoising methods for every sample in learning.

3.1 Auto-regressive Learning

The purpose is to map raw data into latent space via an encoder and exploit the obtained representations for self-supervised learning. Specifically, it involves an encoder f_{enc}, which is a 3-block convolutional architecture, and an auto-regressive (AR) module f_{ar}. For an input $\boldsymbol{x}_i$, the encoder maps it to a high-dimensional latent representation $\mathbf{z}_i = f_{\text{enc}}(\boldsymbol{x}_i)$, where $\mathbf{z}_i \in \mathbb{R}^{C\times r}$. Then, for the representation $\mathbf{z}_i$, the AR module summarizes all $z_{i,j\le t} = \{z_{i,1}, z_{i,2}, ..., z_{i,t}\}$ into a context vector $\mathbf{c}_i = f_{ar}\left(z_{i,j\le t}\right), \mathbf{c}_i \in \mathbb{R}^h$, where h is the hidden dimension of f_{ar}. The context vector $\mathbf{c}_i$ is used to predict the future timesteps from $z_{i,t+1}$ until $z_{i,t+k} (1 \le k < C)$, $z_{i,t+k} \in \mathbb{R}^r$, where k is the number of predicted timesteps and r is the number of output channels in f_{enc}. Here, we use Transformer [Vaswani *et al.*, 2017] as the f_{ar}, which is comprised of successive blocks of multi-headed attention followed by an MLP block. We stack L identical layers to generate the prediction. To enable using $\mathbf{c}_i$ to predict the timesteps from $z_{i,t+1}$ until $z_{i,t+k}$, we adopt a linear layer parameterized by $\mathbf{W} \in \mathbb{R}^{h\times r}$ to map $\mathbf{c}_i$ back into the same dimension as z_i. Finally, we obtain the predicted timesteps $\hat{z}_{i,t+1}$ until $\hat{z}_{i,t+k}$. Accordingly, the reconstruction loss e_i for $\boldsymbol{x}_i$ can be computed by the mean square error between $z_{i,t+j}$ and $\hat{z}_{i,t+j}$:

$$e_i = \frac{1}{k}\sum_{j=1}^{k}\left(z_{i,t+j} - \hat{z}_{i,t+j}\right)^2, \tag{1}$$

where $z_{i,t+j}$ is from $\mathbf{z}_i$ and $\hat{z}_{i,t+j}$ is the prediction. Minimizing the reconstruction loss enables jointly learning f_{enc} and f_{ar} for generating informative representations.

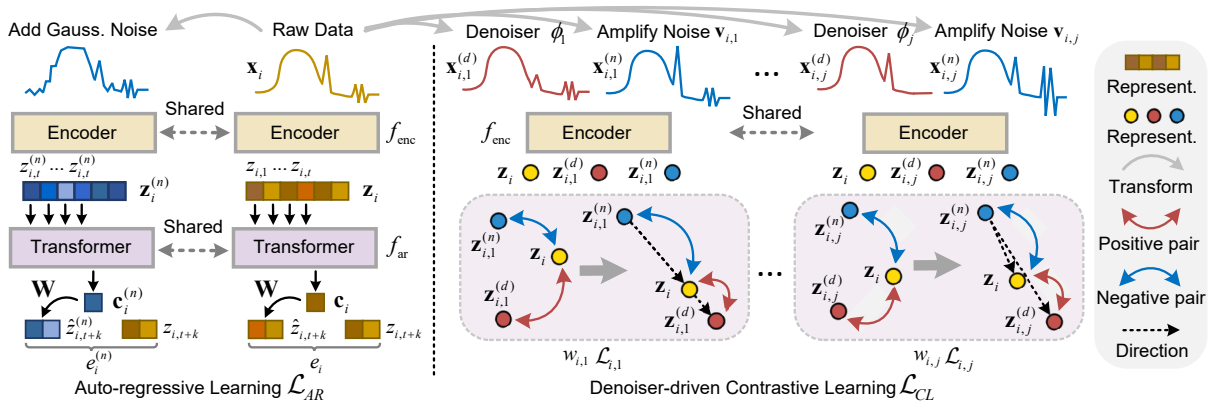

Figure 2: Overview of the method DECL. It involves (*i*) auto-regressive learning, which maps raw data into latent space and exploits the representations for SSL; (*ii*) denoiser-driven contrastive learning, which leverages denoising method ϕ_j to build positive sample $\mathbf{z}_{i,j}^{(d)}$, amplifies the corresponding noise to build negative sample $\mathbf{z}_{i,j}^{(n)}$, and mitigates noise in representation learning; (*iii*) automatic denoiser selection, which injects Gaussian noise to data to avoid overfitting and determines suitable denoising methods for the contrastive learning.

3.2 Denoiser-driven Contrastive Learning

Considering that noise could be amplified in the latent space, we propose to directly eliminate noise from representations via denoiser-driven contrastive learning. Our motivation is that conventional denoising methods have been proven effective for noise removal if they are properly used [Zheng *et al.*, 2020]; intuitively, the representations of the denoised data suffer from less noise than that of raw data and can be exploited as positive samples to guide learning. Vice versa, it is also viable to amplify noise in the raw data, and the representations of the noise-enhanced data are not desired and can act as negative samples. Following this, we propose to exploit the noise-reduced and noise-enhanced data in contrastive learning for better representations.

It involves two steps: (1) generating denoised and noise-enhanced counterparts for raw data and obtaining the representations; (2) mapping the representations of raw data close to that of the denoised ones (i.e., positive samples) and far away from the noise-enhanced counterparts (i.e., negative samples). Specifically, we first conduct data augmentation on raw data. Given a suitable denoising method ϕ_j for raw data $\boldsymbol{x}_i$, we can generate the denoised data $\boldsymbol{x}_{i,j}^{(d)}$. As for synthesizing noise-enhanced data, the principle is to add more noise w.r.t. the corresponding noise type. To this end, we compare raw data $\boldsymbol{x}_i$ with the denoised one $\boldsymbol{x}_{i,j}^{(d)}$ to identify what are the "noises" and then amplify them. In particular, we first obtain the data noise $\boldsymbol{v}_{i,j}$, which has been removed by denoising method ϕ_j, by using $\boldsymbol{x}_i$ subtracts $\boldsymbol{x}_{i,j}^{(d)}$. Then, we scale the values of $\boldsymbol{v}_{i,j}$ to further amplify noise and later add it back to $\boldsymbol{x}_i$. In this way, we obtain the noise-enhanced counterpart $\boldsymbol{x}_{i,j}^{(n)}$ for raw data $\boldsymbol{x}_i$.

After data augmentation, we obtain their representations via the f_{enc} and perform denoiser-driven contrastive learning. Given a sample $\boldsymbol{x}_i$ and a denoising method ϕ_j, we first build a triplet of representations $\mathcal{A}(\mathbf{z}_i, \phi_j)$ as follows:

$$\mathcal{A}(\mathbf{z}_i, \phi_j) = \{\mathbf{z}_{i,j}^{(n)}, \mathbf{z}_i, \mathbf{z}_{i,j}^{(d)}\} \tag{2}$$

where $\mathbf{z}_{i,j}^{(n)}$ and $\mathbf{z}_{i,j}^{(d)}$ are the representation of $\boldsymbol{x}_{i,j}^{(n)}$ and $\boldsymbol{x}_{i,j}^{(d)}$, and are respectively taken as negative and positive sample in contrastive learning. Then, by taking representation $\mathbf{z}_i$ as an anchor, we pull the anchor towards the positive sample while pushing it far away from the negative sample in the latent space. Furthermore, considering the triplet of $\{\mathbf{z}_{i,j}^{(n)}, \mathbf{z}_i, \mathbf{z}_{i,j}^{(d)}\}$ consists of the representations with a descending degree of noise, the direction from large noise to trivial noise in the latent space may indicate a better denoising effect. Accordingly, we also enforce that the mapping direction of positive and negative samples to the anchor shall be opposite for each triplet. Overall, the contrastive learning loss $\mathcal{L}_{i,j}$ for a data $\boldsymbol{x}_i$ and a denoising method ϕ_j is shown as:

$$\begin{aligned}\mathcal{L}_{i,j} &= -\log \frac{\exp\left(\langle \mathbf{z}_i, \mathbf{z}_{i,j}^{(d)} \rangle / \tau\right)}{\sum_{\mathbf{z}_a \in \mathcal{A}(\mathbf{z}_i, \phi_j)} \mathbb{1}_{\mathbf{z}_a \neq \mathbf{z}_i} \exp\left(\langle \mathbf{z}_i, \mathbf{z}_a \rangle / \tau\right)} - \alpha \mathcal{L}_{i,j}^{reg}, \\ \mathcal{L}_{i,j}^{reg} &= \langle (\mathbf{z}_i - \mathbf{z}_{i,j}^{(n)}), (\mathbf{z}_{i,j}^{(d)} - \mathbf{z}_{i,j}^{(n)}) \rangle,\end{aligned} \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity, $\mathbb{1}_{\mathbf{z}_a \neq \mathbf{z}_i} \in \{0, 1\}$ is a binary indicator that equals to 0 when $\mathbf{z}_a$ denotes $\mathbf{z}_i$, τ is a temporal parameter, $\mathcal{L}_{i,j}^{reg}$ is a regularization term that maps representation $\mathbf{z}_i$ towards a noise-free direction.

3.3 Automatic Denoiser Selection

Notably, the above contrastive learning requires a suitable denoising method ϕ_j for sample $\boldsymbol{x}_i$. However, it is nontrivial to fulfill this requirement. Although many conventional denoising methods have been proven effective for noise removal, they may not well handle noise for a given time series $\boldsymbol{x}_i$ when the noise type does not match. To address this issue, we propose to collect a set of commonly used denoising methods $\mathcal{M} = \{\phi_1, \phi_2, \ldots, \phi_m\}$ (shown in Appendix Table 2) in related works and automatically select suitable ones from them.

Here, we propose to exploit the reconstruction error from f_{ar} to determine suitable methods in $\mathcal{M}$. Our inspiration is that noisy data usually cause relatively large reconstruction

errors. Following this, the data processed by suitable denoising methods would have small reconstruction errors, while the unsuitable methods would cause large errors. In this way, suitable methods in $\mathcal{M}$ can be automatically determined for every sample $\boldsymbol{x}_i$. However, directly using the above reconstruction error (Eq.(1)) as the learning objective may not reach the goal. This is because the AR module may overfit the noisy data, thus rendering raw data to have smaller reconstruction errors than the denoised ones. To tackle the issue, we propose another regularization term that encourages f_{ar} to learn global patterns from the noisy time series and avoid overfitting. The key idea is to augment data with more noise, feed their representations to f_{ar}, and enforce reconstructing the raw data that are of less noise. Specifically, we add Gaussian noise to raw data for data augmentation and map it into the latent space to get the representation $\mathbf{z}_i^{(n)}$ as follows:

$$\mathbf{z}_i^{(n)} = \left[z_{i,1}^{(n)}, z_{i,2}^{(n)}, \ldots, z_{i,C}^{(n)}\right]. \tag{4}$$

Similarly, we feed the $z_{i,j\leq t}^{(n)}$ into f_{ar} to obtain $\mathbf{c}_i^{(n)}$ and the prediction $\hat{z}_{i,t+j}^{(n)}$. Formally, the overall learning objective for auto-regressive learning is:

$$\mathcal{L}_{AR} = \frac{1}{N}\sum_{i=1}^{N}\left(e_i + e_i^{(n)}\right), e_i^{(n)} = \frac{1}{k}\sum_{j=1}^{k}\left(z_{i,t+j} - \hat{z}_{i,t+j}^{(n)}\right)^2, \tag{5}$$

where N is the number of training samples. During optimization, $\mathcal{L}_{AR}$ encourages capturing global patterns from the noisy time series for prediction, which avoids overfitting the raw data. Thereby, the samples processed by suitable denoising methods would eventually own smaller reconstruction errors than the raw data, and it is feasible to use reconstruction errors for the automatic selection.

Considering that a data sample may contain multiple types of noises and require different denoising methods, we leverage all the suitable methods in $\mathcal{M}$ by assigning them large weights while setting small weights for the unsuitable ones. Specifically, we first feed the processed sample $\boldsymbol{x}_{i,j}^{(d)}$ by a denoising method ϕ_j to the optimized f_{ar} and obtain the reconstruction error $e_{i,j}$ based on the Eq. (1). Then, we compute the weight value $w_{i,j}$ for ϕ_j with a softmax function as below:

$$w_{i,j} = \frac{e_{i,j}^{-1}}{\sum_{j=1}^{m} e_{i,m}^{-1}}, \tag{6}$$

where the m is the number of methods in $\mathcal{M}$. Hence, the contrastive learning objective $\mathcal{L}_{CL}$ is a weighted combination of $\mathcal{L}_{i,j}$ for the denoising methods in $\mathcal{M}$, i.e.,

$$\mathcal{L}_{CL} = \sum_{i=1}^{N}\sum_{j=1}^{m} w_{i,j}\mathcal{L}_{i,j}, \tag{7}$$

where N is the number of the training samples, $w_{i,j}$ is the weight score for method ϕ_j. By minimizing $\mathcal{L}_{CL}$, our method encourages mapping the representations of raw data toward a noise-free direction, thus mitigating the data noise in latent space. Finally, we combine auto-regressive learning and contrastive learning for joint optimization:

$$\mathcal{L} = \gamma\mathcal{L}_{AR} + \mathcal{L}_{CL}, \tag{8}$$

Datasets	# Train	# Valid	# Test	Length	# Channel	# Class
SleepEDF	16,923	8,462	16,923	3,000	1	5
FaultDiagnosis	5,456	2,728	5,456	5,120	1	3
CPSC18	13,754	6,877	13,754	2,000	12	9
PTB-XL	6,509	3,254	6,509	2,000	12	5
Georgia	6,334	3,167	6,334	2,000	12	6

Table 1: Statistics of the noisy time series datasets.

where γ is a weight value to balance the two terms.

4 Experiments

We perform empirical evaluations to answer the following research questions: **RQ1:** How effective is DECL for unsupervised representation learning? **RQ2:** Is the method effective with fine-tuning? **RQ3:** What are the effects of each component? **RQ4:** Is it robust with varied degrees of noise? **RQ5:** How sensitive is it to the hyper-parameters? **RQ6:** How does the method work in practice?

4.1 Dataset

We employ five noisy time series datasets. SleepEDF [Goldberger *et al.*, 2000] is an EEG dataset in which each sample records human brain activity. The data in FaultDiagnosis [Lessmeier *et al.*, 2016] are collected from sensor readings of bearing machine under different working conditions. CPSC18 [Liu *et al.*, 2018], PTB-XL [Wagner *et al.*, 2020], and Georgia [Alday *et al.*, 2020] are ECG datasets wherein each sample reflects heart activity. The data statistics are shown in Table 1. See Appendix A.1 for more details.

4.2 Experimental Settings

Comparative Methods. We compare our method with representative SSL methods, including contrastive learning-based, such as TF-C [Zhang *et al.*, 2022], TS2vec [Yue *et al.*, 2022], TS-CoT [Zhang *et al.*, 2023b], and CA-TCC [Eldele *et al.*, 2023], as well as generative-based methods, e.g., CRT [Zhang *et al.*, 2023c] and SimMTM [Dong *et al.*, 2023]. We provide detailed descriptions in Appendix A.2.

Evaluation Metrics. We follow previous works [Eldele *et al.*, 2023][Yue *et al.*, 2022] and adopt *Accuracy* and *Weighted-*F_1 scores for classification performance evaluation.

Implementation Details. We split the data into 40%, 20%, and 40% for training, validation, and test set. We set the learning epochs as 100 and adopt a batch size of 128 for both pre-training and downstream tasks, as we notice the training loss does not further decrease. In the transformer, we set L as 4, the number of heads as 4, and the hidden dimension size as 100. The details of the encoder and AR module can be referred to in Appendix C.2. As for the hyper-parameters, we set k as 30% of the total timestamps, assign α as 0.5, and set γ as 0.1 for all the datasets. The method is optimized with Adam optimizer; we set the learning rate as $1e$-4 and the weight decay as $5e$-4. We collect the commonly used denoising methods from related papers for different types of time series and apply the reported hyper-parameters or the default ones for noise removal (see the details in Appendix Table 2). For the baselines, we select the hyper-parameters with

Methods	SleepEDF		FaultDiagnosis		CPSC18		PTB-XL		Georgia	
	Accuracy	*Weighted-F_1*	*Accuracy*	*Weighted-F_1*	*Accuracy*	*Weighted-F_1*	*Accuracy*	*Weighted-F_1*	*Accuracy*	*Weighted-F_1*
TF-C	62.07±0.61	61.24±0.54	76.82±0.72	74.69±0.71	38.46±0.50	35.91±0.64	51.21±0.62	40.93±0.37	48.97±0.34	33.43±0.38
TF-C + *DN*	65.49±0.67	64.83±0.61	78.39±0.64	77.54±0.75	40.53±0.57	37.15±0.52	53.19±0.65	43.24±0.48	51.29±0.45	35.18±0.39
TF-C + *Merge*	61.72±0.72	60.49±0.67	77.51±0.85	75.33±0.70	39.07±0.68	36.03±0.56	52.54±0.71	41.59±0.47	49.58±0.59	33.97±0.43
TS2vec	63.68±0.56	62.75±0.42	77.63±0.49	77.15±0.44	40.56±0.59	37.47±0.47	52.71±0.59	44.37±0.21	51.61±0.43	36.42±0.32
TS2vec + *DN*	67.41±0.63	67.92±0.47	81.15±0.33	80.39±0.25	42.83±0.63	39.08±0.51	55.14±0.32	46.18±0.16	53.27±0.28	38.48±0.17
TS2vec + *Merge*	63.07±0.70	62.38±0.56	79.29±0.56	78.24±0.48	40.98±0.72	37.63±0.69	53.48±0.54	45.24±0.32	52.19±0.41	36.93±0.34
CRT	62.25±0.48	61.03±0.38	75.82±0.58	76.07±0.32	39.25±0.65	36.34±0.34	52.32±0.58	42.31±0.35	49.92±0.27	34.51±0.15
CRT + *DN*	65.90±0.39	65.16±0.15	78.93±0.41	78.49±0.26	41.92±0.51	37.96±0.38	54.17±0.36	44.92±0.14	52.17±0.24	36.34±0.21
CRT + *Merge*	61.94±0.53	61.21±0.39	76.75±0.54	77.04±0.37	39.83±0.79	36.53±0.53	52.89±0.42	43.12±0.23	50.61±0.35	34.79±0.26
SimMTM	63.84±0.57	62.95±0.43	78.39±0.43	77.52±0.39	40.74±0.56	37.72±0.32	52.63±0.43	44.75±0.39	51.34±0.32	35.63±0.24
SimMTM + *DN*	68.62±0.41	68.19±0.24	81.07±0.32	80.93±0.08	43.21±0.44	39.65±0.25	55.18±0.37	46.39±0.16	53.82±0.29	38.15±0.23
SimMTM + *Merge*	63.31±0.52	62.27±0.36	79.64±0.50	78.43±0.33	41.30±0.62	38.14±0.41	53.41±0.51	45.17±0.40	52.19±0.23	36.32±0.32
TS-CoT	64.62±0.59	63.98±0.28	76.32±0.44	75.96±0.32	39.82±0.78	37.43±0.39	52.92±0.54	43.14±0.35	50.97±0.31	34.85±0.21
TS-CoT + *DN*	66.21±0.44	67.05±0.31	79.41±0.37	78.65±0.27	41.36±0.53	38.57±0.25	54.28±0.49	45.30±0.24	52.83±0.25	37.09±0.08
TS-CoT + *Merge*	64.29±0.52	63.17±0.45	77.83±0.46	76.42±0.31	40.27±0.74	37.64±0.46	53.32±0.66	43.85±0.27	51.38±0.46	35.41±0.16
CA-TCC	63.91±0.48	63.37±0.33	78.05±0.43	77.39±0.26	40.79±0.54	38.29±0.49	52.27±0.32	44.91±0.22	52.07±0.27	36.26±0.34
CA-TCC + *DN*	68.48±0.27	68.11±0.21	81.39±0.29	80.74±0.15	43.67±0.47	39.87±0.24	55.54±0.25	46.12±0.13	53.69±0.24	38.37±0.29
CA-TCC + *Merge*	63.67±0.43	63.09±0.37	79.81±0.35	78.26±0.28	41.18±0.61	38.95±0.43	53.11±0.38	45.33±0.18	52.43±0.33	36.84±0.37
Only AR-	62.35±0.51	61.77±0.42	76.95±0.41	75.81±0.32	38.64±0.56	36.26±0.32	51.80±0.53	41.39±0.35	50.42±0.28	34.29±0.35
Only AR	62.89±0.56	62.31±0.36	77.46±0.52	76.73±0.39	39.22±0.48	36.41±0.21	52.27±0.44	41.94±0.29	51.16±0.17	35.03±0.29
CL	67.63±0.39	67.34±0.19	80.84±0.36	80.21±0.11	42.65±0.52	38.93±0.37	54.89±0.52	45.72±0.24	52.98±0.25	37.62±0.27
DECL-	68.35±0.42	67.87±0.34	81.16±0.47	80.58±0.34	43.12±0.33	39.16±0.26	55.06±0.31	45.93±0.16	53.29±0.20	38.01±0.14
DECL*	70.19±0.48	69.52±0.30	81.53±0.33	81.26±0.29	43.96±0.49	39.84±0.34	56.30±0.37	48.24±0.18	54.65±0.12	39.48±0.23
DECL (Ours)	**71.74±0.43**	**70.96±0.25**	**82.78±0.38**	**82.17±0.23**	**45.01±0.32**	**41.65±0.39**	**57.41±0.36**	**49.32±0.27**	**55.37±0.26**	**40.35±0.21**

Table 2: Overall performance (%) comparison on the datasets. *DN* means pre-processed by a suitable denoising method.

the best performance on the validation set for the downstream task. For a fair comparison, we pre-process the raw data for all the baselines by (*i*) applying each denoising method in $\mathcal{M}$ for the dataset and reporting the one with the best performance (i.e., *DN*), and (*ii*) combining all the denoising methods for noise removal (i.e., *Merge*). We run experiments 10 times and report the averaged results.

4.3 Linear Evaluation of Representations (RQ1)

We first pre-train SSL methods with unlabeled data for representation learning, then use a portion of labeled data (i.e., 10% and 30%) to evaluate the effectiveness of the learned representations. Following the standard linear evaluation scheme [Eldele *et al.*, 2023], we fix the parameters of the self-supervised pre-trained model and regard it as an encoder, and then train a linear classifier (single fully connected layer) on top of the encoder. The results with 10% and 30% training labels are shown in Table 2 and Appendix Table 1. We find that existing SSL methods achieve sub-optimal performance on noisy data and the performances get boosted after applying a suitable denoising method for noise mitigation. Besides, directly combining all the denoising methods for preprocessing causes unsatisfactory performance. This can be explained by that certain denoising methods may induce extra noise into data when the noise type does not match. Additionally, DECL achieves superior performance than other SSL methods. Specifically, the performance boost over the best baseline on the CPSC18 dataset is about 3%. This is because our method exploits suitable denoising methods to mitigate noise and guide representation learning.

4.4 Fine-tuning Performance Evaluation (RQ2)

To simulate real-world scenarios where a few labeled data are accessible, we fine-tune our pre-trained model with the labels and investigate its effectiveness. There are two setups.

Fine-tuning on the Source Dataset. Following previous works [Lan *et al.*, 2022], we pre-train the model with unlabeled data and then fine-tune it with different amounts of the training labels (i.e., 1%, 5%, 10%, 50%, 75%, and 100%) from the same dataset. Fig. 3 shows the performance comparison against its supervised learning counterpart and a strong baseline on three datasets. We observe that supervised learning performs poorly with limited labeled data (e.g., 1%), while the fine-tuned models achieve significantly better performance. It verifies that self-supervised pre-training can alleviate the label scarcity issue. Besides, DECL outperforms the strong baseline when using different ratios of training labels in fine-tuning. In brief, it shows the effectiveness of our method under the fine-tuning mode.

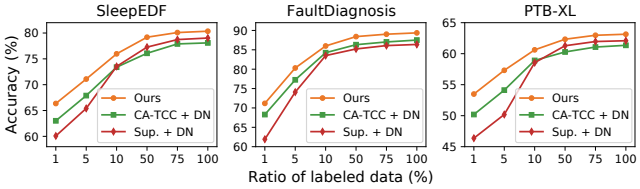

Figure 3: Performance comparison for semi-supervised representation learning with different percentages of labeled data.

Fine-tuning on New Dataset. To evaluate the generalizability of the learned representations, we further pre-train the model on one dataset and perform supervised fine-tuning on another dataset. Specifically, we follow the one-to-one evaluation scheme [Zhang *et al.*, 2022] and use a portion of labeled data (10%) for fine-tuning. Table 3 shows the results under six cross-dataset scenarios on the ECG data. Similarly, we find that combining all the denoising methods would render unsatisfactory performance for the baselines than that of applying a suitable one. Besides, our fine-tuned model consistently outperforms the strong baselines. Furthermore, pre-training on a comprehensive dataset (e.g., CPSC18) usu-

Methods	P → C	G → C	C → P	G → P	C → G	P → G
SimMTM + *DN*	43.74	42.59	55.48	54.59	54.73	52.62
SimMTM + *Merge*	41.62	40.81	53.96	52.18	52.80	50.91
CA-TCC + *DN*	44.17	42.83	57.43	56.21	54.52	52.47
CA-TCC + *Merge*	42.61	40.75	55.69	54.74	53.25	51.16
DECL (Ours)	**46.23**	**44.25**	**60.21**	**59.03**	**56.41**	**55.69**

Table 3: Performance (accuracy %) of the transferability evaluation on CPSC18 (C), PTB-XL (P), and Georgia (G) datasets.

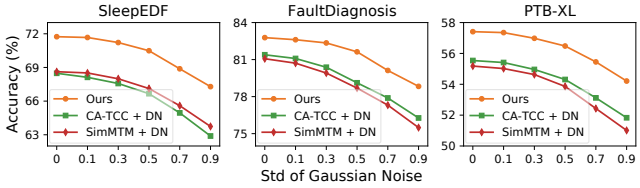

Figure 4: Unsupervised representation learning performance under varying degrees of data noise.

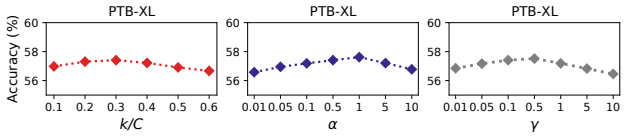

Figure 5: Hyper-parameter analysis results.

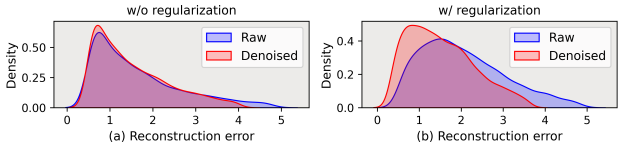

Figure 6: Comparison of the reconstruction error distribution between raw data and denoised data. After adding regularization, the distribution gap is more distinguishable.

ally promises better performance on the new dataset, which is consistent with the findings from related papers [Yang *et al.*, 2023]. Overall, the results verify that DECL can alleviate the impact of noise on learning informative representations and generalize well on cross-dataset scenarios.

4.5 Ablation Study (RQ3)

We also examine the effect of each component in DECL on its overall performance. Specifically, we derive different method variants for comparison: (1) DECL-, which drops the regularization term in the auto-regressive learning and remains others unchanged; (2) DECL*, which deletes the direction constraint in the contrastive learning; (3) CL, which assigns equal weights to the denoising methods and keeps others unchanged; (4) Only AR, which merely leverages $\mathcal{L}_{AR}$ as the learning objective; (5) Only AR-, which exploits $\mathcal{L}_{AR}$ without the regularization term. The results are shown in Table 2. We find that (*i*) merely using $\mathcal{L}_{AR}$ renders unsatisfactory performance. This is because the noise in latent space cannot be eliminated and hampers representation learning. (*ii*) Our method outperforms CL and DECL-, illustrating that attaching equal importance to the denoising methods would hinder representation learning and the regularization term in $\mathcal{L}_{AR}$ is conducive to the overall performance. (*iii*) Our method achieves higher performance than DECL*, showing that the direction constraint in contrastive learning helps to eliminate the noise in representation learning.

4.6 Robustness Analysis against Data Noise (RQ4)

We further investigate the robustness of DECL against varying degrees of data noise. Specifically, we induce Gaussian noise to the raw data (with zero mean and varying standard deviations) and then conduct the linear evaluation for performance comparison. The results on three datasets are shown in Fig. 4; see Appendix B.2 for more results. We have two interesting findings. (*i*) By inducing a larger amount of noise, the performances of the methods gradually drop. This occurs since the adopted denoising method(s) cannot eliminate the noise sufficiently, thereby impairing representation learning. (*ii*) Our method consistently outperforms the strong baselines. Because DECL selects suitable denoising methods for every sample and further diminishes noise from representations with contrastive learning.

4.7 Sensitivity Analysis (RQ5)

We perform sensitivity analysis to study the main hyper-parameters: the number of predicted future timesteps k in Eq. (5), the weight α in Eq. (3), and the weight γ in Eq. (8). Specifically, we adopt the same setup as the linear evaluation experiment, present the results of the PTB-XL dataset in Fig. 5, and show more results in Appendix B.3. We first analyze the impact of k, where the x-axis denotes the percentage k/C and C is the total number of timesteps. It shows that, as the percentage value increases, the performance first boosts and then declines. Hence, we suggest setting it to the scope of 0.1-0.4 in practice. Regarding the hyper-parameter α and γ, when raising its value, the performance first rises and later declines. It occurs because either a small or a large value fails to achieve a balance between the learning objectives. Given this, we recommend setting the value of α to 0.1-5 and the value of γ to 0.1-1.

4.8 Visualization Results (RQ6)

The Effect of the Regularization Term. Here, we examine whether the proposed regularization term in $\mathcal{L}_{AR}$ (see Eq. (5)) can alleviate the overfitting issue and enable the reconstruction error as an indicator for choosing suitable denoising methods. Specifically, we compare the model learned with $\mathcal{L}_{AR}$ (i.e., with regularization) and the counterpart without regularization by visualizing the distribution of reconstruction errors on the PTB-XL dataset. See Appendix B.4 for more details. The results in Fig. 6 show that (*i*) without a regularization term, the distributions of reconstruction errors between raw and denoised data become similar, rendering it hard to use the reconstruction error as the indicator. (*ii*) After adding the regularization term, the distributions of reconstruction errors are more distinguishable. In brief, it verifies that the proposed regularization term indeed alleviates overfitting and benefits determining suitable denoising methods.

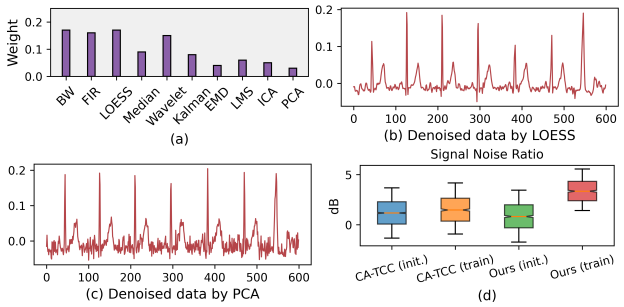

Figure 7: (a-c) The learned weights for denoising methods and their effects. (d) The comparison of SNR on representation.

Analysis on the Selected Denoising Methods. We also verify whether DECL can assign more weights to the suitable denoising methods. In detail, we adopt a case study on the PTB-XL dataset to (*i*) visualize the weight values of the denoising methods in $\mathcal{M}$ for a sample with high-frequency noise, and (*ii*) showcase a raw data and the denoised counterparts to examine the denoising effect. As shown in Fig. 7(a), the AR module assigns different weights on the set of potentially feasible denoising methods. For example, the LOESS method, which is reported to remove high-frequency noise for ECG [Burguera, 2018], obtains large a weight value; whereas the PCA method, which does not suit this type of noise [Alickovic and Subasi, 2015], owns a small value. We further visualize the denoised data by LOESS in Fig. 7(b), which demonstrates that noises are well mitigated. However, as shown in Fig. 7(c), the denoised data by PCA still involve severe noises. More results are presented in Appendix B.5. The above analyses prove that our method can automatically select suitable denoising methods for the noisy time series.

Analysis on the Learned Representations. Further, we analyze the effect of DECL on representations. Similar to the pilot study, we induce some Gaussian noise to the raw data and compare the SNR value of the randomly initialized representations and that of the learned representations after optimization. The result on the PTB-XL dataset is presented in Fig. 7(d). We observe that the SNR value of a strong baseline CA-TCC almost remains steady after training, while the SNR value of our method is boosted significantly. See Appendix B.6 for detailed illustrations. The analyses verify that DECL can indeed mitigate noise in representation learning.

5 Related Work

5.1 Time Series Self-supervised Learning

To alleviate the reliance on numerous annotated training data for supervised learning, an increasing number of researches focus on learning representations of time series in a self-supervised manner. Time series SSL methods roughly fall into three categories [Zhang *et al.*, 2023a]: (1) contrastive-based methods [Meng *et al.*, 2023b][Ma *et al.*, 2023][Meng *et al.*, 2023a], which are characterized by constructing positive and negative samples for contrastive learning, (2) generative-based methods [Zerveas *et al.*, 2021][Li *et al.*, 2022], which minimize the reconstruction error between raw data and the generated counterparts for model learning, (3) adversarial-based methods [Seyfi *et al.*, 2022], which usually leverage generator and discriminator for adversarial learning. Among them, the contrastive-based methods are the dominant ones. The key steps of contrastive learning involve building similar (positive) and dissimilar (negative) pairs of data samples and mapping the representations of positive pairs nearby while mapping those of negative pairs farther apart. The contrastive-based methods can be further divided into sampling-based [Yèche *et al.*, 2021][Tonekaboni *et al.*, 2021], augmentation-based [Yue *et al.*, 2022][Yang and Hong, 2022], and prediction-based [Oord *et al.*, 2018][Deldari *et al.*, 2021]. A recent method called TS-CoT [Zhang *et al.*, 2023b] assumes that complementary information from different views can be used to mitigate data noise. Different from it, our method leverages the power of conventional denoising methods to guide representation learning. More details are shown in Appendix D.1.

5.2 Time Series Denoising

Existing time series denoising methods can be divided into two categories: conventional methods and learning-based methods. The conventional methods include empirical mode decomposition [Huang *et al.*, 1998], wavelet filtering [Chaovalit *et al.*, 2011], sparse decomposition-based, Bayesian filtering [Barber *et al.*, 2011], and hybrid method (e.g., wavelet-based ICA [Castellanos and Makarov, 2006]) for noise mitigation. The advantages of conventional methods lie in the great denoising effects if properly adopted to match the noise type. However, selecting suitable denoising methods requires prior knowledge or trial-and-error. To avoid human efforts, many learning-based methods that can mitigate the effect of noise have been proposed in recent years. Based on network architecture, these methods include wavelet neural networks, RNN-based [Zhang *et al.*, 2023d][Yoon *et al.*, 2022], and auto-encoders [Zheng *et al.*, 2022]. Differing from the previous efforts, our work leverages the conventional denoising methods to guide mitigating noise in learning, which combines the advantages of the two method categories.

6 Conclusion

In this work, we investigate the problem of mitigating the effect of data noise for time series SSL. Accordingly, we propose an end-to-end method called DEnoising-aware Contrastive Learning (DECL) for noise elimination in representation learning. It automatically selects suitable denoising methods for every sample to guide learning and performs a customized contrastive learning toward obtaining noise-free representations. Extensive empirical results verify the effectiveness of our method. Additionally, we perform comprehensive analyses to verify our claims, such as the distribution visualization of reconstruction errors and the denoising effect visualization of the selected methods. Future works can explore (*i*) how to automatically determine suitable hyperparameters of the denoising methods; (*ii*) examine the effectiveness of the method for more downstream tasks, e.g., forecasting and anomaly detection [Lai *et al.*, 2021].

Ethical Statement

There are no ethical issues.

Acknowledgments

This research was supported by the grant of DaSAIL project P0030970 funded by PolyU (UGC).

References

[Alday *et al.*, 2020] Erick A Perez Alday, Annie Gu, Amit J Shah, Chad Robichaux, An-Kwok Ian Wong, Chengyu Liu, Feifei Liu, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, et al. Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020. *Physiological measurement*, 41(12):124003, 2020.

[Alickovic and Subasi, 2015] Emina Alickovic and Abdulhamit Subasi. Effect of multiscale pca de-noising in ecg beat classification for diagnosis of cardiovascular diseases. *Circuits, Systems, and Signal Processing*, 34:513–533, 2015.

[Barber *et al.*, 2011] David Barber, A Taylan Cemgil, and Silvia Chiappa. *Bayesian time series models*. Cambridge University Press, 2011.

[Burguera, 2018] Antoni Burguera. Fast qrs detection and ecg compression based on signal structural analysis. *IEEE journal of biomedical and health informatics*, 23(1):123–131, 2018.

[Castellanos and Makarov, 2006] Nazareth P Castellanos and Valeri A Makarov. Recovering eeg brain signals: Artifact suppression with wavelet enhanced independent component analysis. *Journal of neuroscience methods*, 158(2):300–312, 2006.

[Chaovalit *et al.*, 2011] Pimwadee Chaovalit, Aryya Gangopadhyay, George Karabatis, and Zhiyuan Chen. Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2):1–37, 2011.

[Chawla, 2011] MPS Chawla. Pca and ica processing methods for removal of artifacts and noise in electrocardiograms: A survey and comparison. *Applied Soft Computing*, 11(2):2216–2226, 2011.

[Chowdhury *et al.*, 2022] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. Tarnet: Task-aware reconstruction for time-series transformer. In *KDD*, pages 212–220, 2022.

[Deldari *et al.*, 2021] Shohreh Deldari, Daniel V Smith, Hao Xue, and Flora D Salim. Time series change point detection with self-supervised contrastive predictive coding. In *Proceedings of the Web Conference*, pages 3124–3135, 2021.

[Dong *et al.*, 2023] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. In *NeurIPS*, 2023.

[Eldele *et al.*, 2023] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. Self-supervised contrastive representation learning for semi-supervised time-series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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

[He *et al.*, 2015] Hong He, Yonghong Tan, and Yuexia Wang. Optimal base wavelet selection for ecg noise reduction using a comprehensive entropy criterion. *Entropy*, 17(9):6093–6109, 2015.

[Huang *et al.*, 1998] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995, 1998.

[Ismail Fawaz *et al.*, 2019] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[Lai *et al.*, 2021] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *NeurIPS*, 2021.

[Lai *et al.*, 2023] Jiewei Lai, Huixin Tan, Jinliang Wang, Lei Ji, Jun Guo, Baoshi Han, Yajun Shi, Qianjin Feng, and Wei Yang. Practical intelligent diagnostic algorithm for wearable 12-lead ecg via self-supervised learning on large-scale dataset. *Nature Communications*, 14(1):3741, 2023.

[Lan *et al.*, 2022] Xiang Lan, Dianwen Ng, Shenda Hong, and Mengling Feng. Intra-inter subject self-supervised learning for multivariate cardiac signals. In *AAAI*, volume 36, pages 4532–4540, 2022.

[Lessmeier *et al.*, 2016] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, volume 3, 2016.

[Li *et al.*, 2022] Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *NeurIPS*, 35:23009–23022, 2022.

[Liu *et al.*, 2018] Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu,

Caiyun Ma, Shoushui Wei, Zhiqiang He, et al. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7):1368–1373, 2018.

[Luo *et al.*, 2019] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *AAAI*, pages 3094–3100. AAAI Press Palo Alto, CA, USA, 2019.

[Ma *et al.*, 2023] Qianli Ma, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, Zhongzhong Yu, and James T Kwok. A survey on time-series pre-trained models. *arXiv*, 2023.

[Meng *et al.*, 2023a] Qianwen Meng, Hangwei Qian, Yong Liu, Lizhen Cui, Yonghui Xu, and Zhiqi Shen. MHCCL: masked hierarchical cluster-wise contrastive learning for multivariate time series. In *AAAI*, pages 9153–9161, 2023.

[Meng *et al.*, 2023b] Qianwen Meng, Hangwei Qian, Yong Liu, Yonghui Xu, Zhiqi Shen, and Lizhen Cui. Unsupervised representation learning for time series: A review. *arXiv*, 2023.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018.

[Robbins *et al.*, 2020] Kay A Robbins, Jonathan Touryan, Tim Mullen, Christian Kothe, and Nima Bigdely-Shamlo. How sensitive are eeg results to preprocessing methods: a benchmarking study. *IEEE transactions on neural systems and rehabilitation engineering*, 28(5):1081–1090, 2020.

[Seyfi *et al.*, 2022] Ali Seyfi, Jean-Francois Rajotte, and Raymond Ng. Generating multivariate time series with common source coordinated gan (cosci-gan). *NeurIPS*, 35:32777–32788, 2022.

[Tonekaboni *et al.*, 2021] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *ICLR*, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[Wagner *et al.*, 2020] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

[Yang and Hong, 2022] Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *ICML*, 2022.

[Yang *et al.*, 2023] Chaoqi Yang, M Brandon Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. In *NeurIPS*, 2023.

[Yèche *et al.*, 2021] Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. Neighborhood contrastive learning applied to online patient monitoring. In *ICML*, pages 11964–11974, 2021.

[Yoon *et al.*, 2022] TaeHo Yoon, Youngsuk Park, Ernest K Ryu, and Yuyang Wang. Robust probabilistic time series forecasting. In *AISTATS*, pages 1336–1358. PMLR, 2022.

[Yue *et al.*, 2022] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *AAAI*, volume 36, pages 8980–8987, 2022.

[Zerveas *et al.*, 2021] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *KDD*, pages 2114–2124, 2021.

[Zhang *et al.*, 2021a] Haoming Zhang, Mingqi Zhao, Chen Wei, Dante Mantini, Zherui Li, and Quanying Liu. Eegdenoisenet: a benchmark dataset for deep learning solutions of eeg denoising. *Journal of Neural Engineering*, 18(5):056057, 2021.

[Zhang *et al.*, 2021b] Yatao Zhang, Junyan Li, Shoushui Wei, Fengyu Zhou, and Dong Li. Heartbeats classification using hybrid time-frequency analysis and transfer learning based on resnet. *IEEE Journal of Biomedical and Health Informatics*, 25(11):4175–4184, 2021.

[Zhang *et al.*, 2022] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *NeurIPS*, 35:3988–4003, 2022.

[Zhang *et al.*, 2023a] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *arXiv*, 2023.

[Zhang *et al.*, 2023b] Weiqi Zhang, Jianfeng Zhang, Jia Li, and Fugee Tsung. A co-training approach for noisy time series learning. In *CIKM*, pages 3308–3318, 2023.

[Zhang *et al.*, 2023c] Wenrui Zhang, Ling Yang, Shijia Geng, and Shenda Hong. Self-supervised time series representation learning via cross reconstruction transformer. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Zhang *et al.*, 2023d] Xueli Zhang, Cankun Zhong, Jianjun Zhang, Ting Wang, and Wing WY Ng. Robust recurrent neural networks for time series forecasting. *Neurocomputing*, 526:143–157, 2023.

[Zheng *et al.*, 2020] Jianwei Zheng, Jianming Zhang, Sidy Danioko, Hai Yao, Hangyuan Guo, and Cyril Rakovski. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific data*, 7(1):48, 2020.

[Zheng *et al.*, 2022] Zhong Zheng, Zijun Zhang, Long Wang, and Xiong Luo. Denoising temporal convolutional recurrent autoencoders for time series classification. *Information Sciences*, 588:159–173, 2022.