TOWARDS UNSUPERVISED TIME SERIES REPRESENTATION LEARNING: A DECOMPOSITION PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing contrastive methods of universal time series representation learning mainly rely on distilling invariant patterns at varying scales and building contrastive loss with the help of negative sampling. However, the invariance assumptions may not hold in real-world time-series data, and the infamous negative sampling could bring in new biases for representation learning. In this work, we propose a novel contrastive learning approach toward time series representation learning on top of trend-seasonality decomposition, namely TS-DC. TS-DC differentiates itself from prior methods in three folds: 1) a time series decomposition approach is devised to distill different aspects/components of a complex time series; 2) a novel component-wise contrastive loss is proposed in which negative sampling is not necessary; 3) the informative signals of time series can be captured comprehensively by means of adaptive contrasting. Extensive experiments on different public benchmark datasets validate the superior performance of our proposed representation learning method.

1 INTRODUCTION

Representing informative signals from intricate and noisy time series (TS) data is challenging for machine learning models (Gamboa, 2017). Time series representation learning (TSRL) has been found beneficial for many downstream tasks, including classification, forecasting and anomaly detection (Leng et al., 2009; Zerveas et al., 2021; Yue et al., 2022). Though there have been many supervised TS feature extraction methods (Wu et al., 2018; Ruiz et al., 2021; Xiao et al., 2021), they can hardly tackle the high-dimensional and sparsely labeled real-world TS data (Ching et al., 2018). Unsupervised representation learning aims to extract low-dimensional and unified representation without requiring labels, which sheds light on TS representation learning. In this work, we study the problem of unsupervised time series representation learning (UTSRL).

Numerous recent works have come up with creative and interesting techniques toward unsupervised representation learning in different domains (Pinheiro et al., 2020; Caron et al., 2018; Ren et al., 2019b; Cheng et al., 2019). Inspired by this, a few contrastive learning approaches have been proposed to address the UTSRL problem. The main idea goes like this: there exist key properties in a time series that keep invariant across time and contexts, then one can encourage the model to learn such underlying signals by means of data augmentation at certain or different semantic scales. For example, rather than focusing on sub-sequence invariant signals (Franceschi et al., 2019), neighborhood smoothness (Tonekaboni et al., 2020), or long-term dependency (Yang & Hong, 2022), Yue et al. (2022) propose to leverage scale invariant information by means of hierarchical contrasting and data augmentation methods (including series cropping and timestamp masking, *etc.*). Besides, cross-view contrasting is another well applied method to enable the learnt representation to be aware of transformation-/contextual-consistency (Eldele et al., 2021) in time series data.

However, due to the huge diversities among different time series data, it is challenging to construct high-quality negative samples toward universal time series representation. Furthermore, since multi-level noises exist in real-world time series data (Kostelich & Schreiber, 1993), it might be difficult to tell generated representations from noises. Though it is useful to leverage large amount of negative samples, increasing the number of negative samples does not necessarily help (Arora et al., 2019), and extra biases could be raised by negative sampling (Chuang et al., 2020). In addition, distilling the underlying and inherent features encoded in time series data, such as temporal dependencies (Xuan

& Murphy, 2007), is challenging yet critical for unsupervised TS representation learning. While the assumptions about TS-property-invariance may lead to spurious correlations, which could harm the generalizability of UTSRL.

To this end, we propose a time series decomposition contrastive learning approach (named TS-DC) for UTSRL. Specifically, we first propose a deep trend-seasonality decomposition method to examine underlying trend and seasonality in time series data with the help of variational auto-encoder (VAE). To thoroughly exploit the signals in a time series, the remainder is further decomposed recursively. Then, a component-wise contrastive loss is proposed to encourage the learnt representation be closer to informative components. In this way, negative sampling can be avoided since the negative samples are deterministic and conditioned on the decomposition process only. Afterward, a weight adjustment module is devised to balance the decaying weights of different components, such that the adaptive contrasting can be achieved. We conduct extensive experiments on various benchmark datasets toward different downstream tasks (*e.g.,* classification and forecasting), and the rich results demonstrate that TS-DC outperforms existing SOTA unsupervised time series representation learning methods with satisfactory margins. Main contributions of this work are summarized below:

- We propose a deep decomposition method to yield disentangled yet informative inherent features for time series data without seeking contextual-invariance signals.
- We propose a novel component-wise contrastive learning method devoted to time series data, such that the notorious negative sampling can be avoided.
- To fully capture the multi-faceted features in time series data, an adaptive contrasting mechanism is proposed.

2 Related Work

Unsupervised Time Series Representation Learning. 1) Generative. Time series representations can be obtained through generative models, like auto-encoders (Yingzhen & Mandt, 2018; Srivastava et al., 2015; Malhotra et al., 2017; Chung et al., 2016). These methods generally embed the input into a latent space via the encoder and decode the representation back to recover the input signals. Whereas, the representation distribution of the generative method is modeled at point-wise level, which may result in limited generalizability. **2) Contrastive.** Contrastive learning has achieved remarkable improvements toward unsupervised representation learning and has been applied to many different fields successfully, such as computer vision (Oord et al., 2018; Logeswaran & Lee, 2018; Tian et al., 2020; Chen et al., 2020; Gao et al., 2021; Kuang et al., 2021) and speech recognition (Schneider et al., 2019; Kharitonov et al., 2021; Cuervo et al., 2022; Wang et al., 2021). There have been works that successfully apply contrastive learning techniques to time series data (Franceschi et al., 2019; Tonekaboni et al., 2020; Yue et al., 2022; Eldele et al., 2021; Zerveas et al., 2021). Franceschi et al. (2019) proposed an encoder formed by dilated convolutions that admits variable-length inputs, and trained with a triplet loss using time-based negative sampling. Tonekaboni et al. (2020) studied the stationary properties of time series and incorporate sample weight adjustment to produce generalizable time series representation. Eldele et al. (2021) proposed to encourage the invariance of different data augmentations and put efforts to learn robust representation by means of cross-view prediction and contextual contrasting. However, the defects of these contrastive methods, such as sampling bias and invariance assumption, could lower the generalizability of learnt time series representation. To relieve the sampling bias, Yang & Hong (2022) exploit the instance-level augmentation with standard dropout on time-series data and devise a temporal-spectral method to enrich the representations. Instead of negative sampling, we build the contrasting pairs for UTSRL from a decomposition perspective, which is different from existing works.

Time Series Decomposition. STL (Seasonal-Trend decomposition using Loess) is one of the most classic and widely-used decomposition method (Cleveland et al., 1990; Dokumentov et al., 2015; Wen et al., 2019). However, the limited flexibility of these STL-based methods might not be able to handle the time series data with large diversities. Recently, the effectiveness of applying deep learning algorithms to the time series decomposition has been proved (Asadi & Regan, 2020; Wu et al., 2021; Taylor & Letham, 2018; Oreshkin et al., 2019). Our decomposition method is similar to N-Beats (Oreshkin et al., 2019), while we aim to maximize the gap between the decomposed trend and seasonality, and re-generate the trend and seasonality, separately. In addition, we implement a reverting process to assembly all the decomposed artifacts such that the input series can be recovered.

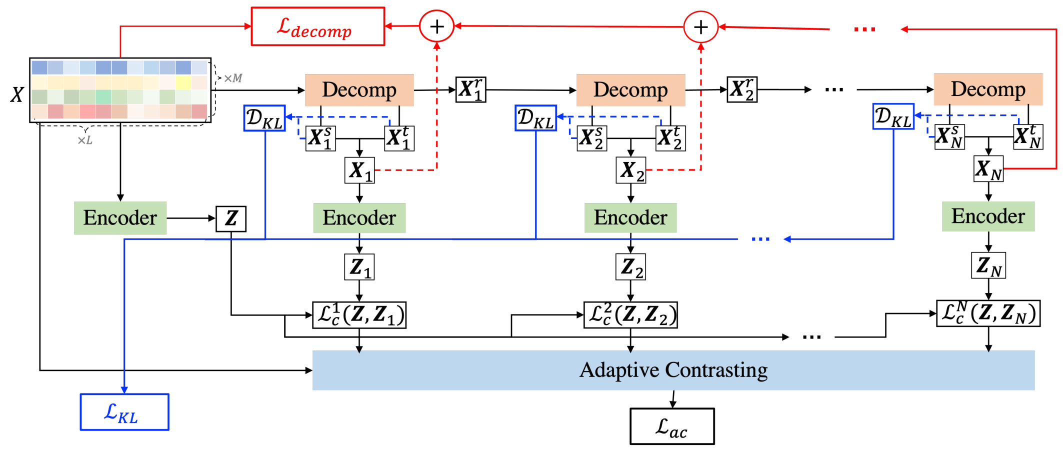


Figure 1: The framework overview of TS-DC. A time series $\boldsymbol{X} \in \mathbb{R}^{L\times M}$ is decomposed into $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N\}$, and the outcome representation $\boldsymbol{Z}$ is encouraged to imitate the representations of decomposed components, *i.e.*, $\{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_n\}$, separately. $\mathcal{L}_C^n(\cdot)$ denotes the component-wise contrastive loss, then an adaptive contrasting mechanism is employed to harness the signals in different components comprehensively.

3 METHODOLOGY

Problem Statement. Given an M-dimension multivariate time series $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M]$, $\forall m \in \{1, 2, \cdots, M\}$, $\boldsymbol{x}_m \in \mathbb{R}^{L_m}$ where L_m denotes the length of m-th series in $\boldsymbol{X}$. $\exists\, 1 \leq m_1 < m_2 \leq M$ such that $L_{m_1} \neq L_{m_2}$. Then, unsupervised time series representation learning (UTSRL) aims to learn a function $\boldsymbol{f}_\theta : \mathbb{R}^{L\times M} \to \mathbb{R}^{L\times D}$ ($L = \max(L_1, \cdots, L_M)$ and $D > 0$) that maps $\boldsymbol{X}$ into a representation $\boldsymbol{Z}$, where $\boldsymbol{Z}$ can describe $\boldsymbol{X}$ as much as possible.

As shown in Figure 1, TS-DC consists of multiple encoders. The first encoder absorbs the input $\boldsymbol{X} \in \mathbb{R}^{L\times M}$ and generates the initial representation $\boldsymbol{Z} \in \mathbb{R}^{L\times D}$. Meanwhile, we decompose $\boldsymbol{X}$ into additive trend, seasonality, and remainder. Then the remainder is decomposed into pairs of trend and seasonality recursively. Each pair of trend and seasonality are added back to form a component $\boldsymbol{X}_n$ ($n = 1, 2, \cdots, N$). After further embedding $\boldsymbol{X}_n$ into $\boldsymbol{Z}_n$, a component-wise contrastive loss is applied, separately. At last, the representations of different components are taken into account comprehensively by means of adaptive contrasting. We introduce the details of our approach in following subsections separately.

3.1 TIME SERIES DECOMPOSITION

The trend-seasonality decomposition can reveal underlying insights of a complex time series (Laptev et al., 2015; Wen et al., 2019). Trend usually reflects long-term increase and decrease, seasonality depicts periodic patterns, and spike & dip can be encoded in the remainder. The detailed operations in a single decomposition can be seen in Figure 2.

Specifically, we construct a time vector $\boldsymbol{T} = [\boldsymbol{1}, \boldsymbol{t}, \cdots, \boldsymbol{t}^c]$ where $\boldsymbol{t} = ([1, 2, \cdots, L-1]/L)^T$ and $c > 0$ is a constant that controls the range of trend. Then we can obtain the trend for the first decomposition as $\widetilde{\boldsymbol{X}_1^t} = \boldsymbol{T} \times \text{FC}(\boldsymbol{X})$, where $\text{FC}(\cdot)$ denotes a fully connected network. Similarly, the seasonality can be extracted with Fourier Transform: $\widetilde{\boldsymbol{X}_1^s} = \boldsymbol{S} \times \text{FC}(\boldsymbol{X})$, where

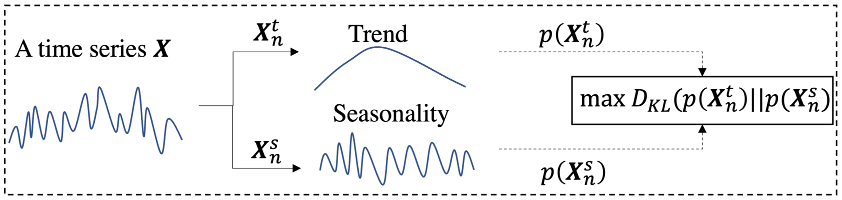


Figure 2: An illustration of a single decomposition in the TS-DC framework.

$\boldsymbol{S} = [1, cos(2\pi \boldsymbol{t}), \cdots, cos(2\pi(\frac{L}{2}-1)\boldsymbol{t}), sin(2\pi \boldsymbol{t}), \cdots, sin(2\pi(\frac{L}{2}-1)\boldsymbol{t})]$ is the matrix of sinusoidal waveforms for extracting seasonal and periodic patterns. To ensure the validity of decomposed trend and seasonality, *w.l.o.g.*, we assume that trend $\boldsymbol{X}_1^t$ and seasonality $\boldsymbol{X}_1^s$ are Gaussian variables, and the distributions of $\boldsymbol{X}_1^t$ and $\boldsymbol{X}_1^s$ can be obtained by a mapping function $g(\cdot)$,

$$(\boldsymbol{\mu}_1^t, \boldsymbol{\sigma}_1^t) = g(\widetilde{\boldsymbol{X}_1^t}), \quad (\boldsymbol{\mu}_1^s, \boldsymbol{\sigma}_1^s) = g(\widetilde{\boldsymbol{X}_1^s}). \tag{1}$$

We then re-sample the trend and seasonality, *i.e.,* $\boldsymbol{X}_1^t \sim p(\boldsymbol{X}_1^t) = \mathcal{N}(\boldsymbol{X}_1^t; \boldsymbol{\mu}_1^t, \boldsymbol{\sigma}_1^t), \boldsymbol{X}_1^s \sim p(\boldsymbol{X}_1^s) = \mathcal{N}(\boldsymbol{X}_1^s; \boldsymbol{\mu}_1^s, \boldsymbol{\sigma}_1^s)$. Since that trend and seasonality should behave differently, the following KL-divergence should be maximized,

$$D_{\mathrm{KL}}(p(\boldsymbol{X}_1^t) \| p(\boldsymbol{X}_1^s)) = \log \frac{\boldsymbol{\sigma}_1^s}{\boldsymbol{\sigma}_1^t} + \frac{(\boldsymbol{\sigma}_1^t)^2 + (\boldsymbol{\mu}_1^t - \boldsymbol{\mu}_1^s)^2}{2(\boldsymbol{\sigma}_1^s)^2} - \frac{1}{2}. \tag{2}$$

As shown in Figure 1, the decomposition operations can be carried out recursively. The n-th decomposition takes the residual of previous decomposition as input, *i.e.,* $\boldsymbol{X}_n^r = \boldsymbol{X}_{n-1}^r - (\boldsymbol{X}_n^s + \boldsymbol{X}_n^t)$. Then, we aim to maximize the sum of all KL-divergences in different decomposition operations,

$$\widetilde{\mathcal{L}}_{KL} = \sum_{n=1}^{N} D_{\mathrm{KL}}(p(\boldsymbol{X}_n^t) \| p(\boldsymbol{X}_n^s)). \tag{3}$$

To simplify the training procedure, we instead minimize $\mathcal{L}_{KL} = 1/(1 + e^{\alpha \cdot \widetilde{\mathcal{L}}_{KL}})$, where α is a hyper-parameter. In addition, in order to reduce the variance of decomposition operations, the sum of all the decomposed components should be close to the original time series $\boldsymbol{X}$,

$$\mathcal{L}_{decomp} = \|\boldsymbol{X} - \sum_{i=1}^{n} (\boldsymbol{X}_i^s + \boldsymbol{X}_i^t)\|^2. \tag{4}$$

By minimizing $\mathcal{L}_{decomp}$, we can ensure the quality of decomposition operations *w.r.t.* approximating original time series.

3.2 Component-Wise Contrasting

The above decomposition can be denoted as $\boldsymbol{X} \approx \sum_{n=1}^{N} \boldsymbol{X}_n$ where $\boldsymbol{X}_n = \boldsymbol{X}_n^s + \boldsymbol{X}_n^t$. Basically, component $\boldsymbol{X}_n$ implies the trend and seasonality of input $\boldsymbol{X}$ regarding a specific aspect. To enforce the outcome representation to learn the complex signals encoded in $\boldsymbol{X}$ from different perspectives, for the n-th decomposition, we treat embeddings $\boldsymbol{Z}_n$ and $\boldsymbol{Z}_m$ ($n \neq m$) as positive and negative, respectively. As a result, the trend and seasonality of $\boldsymbol{X}_n$ ($n = 1, 2, \cdots, N$) can be highlighted by contrasting positives with negatives,

$$\mathcal{L}_c^n(\boldsymbol{X}, \boldsymbol{X}_n) = -\log \frac{e^{\mathcal{E}(\boldsymbol{X})^T \mathcal{E}(\boldsymbol{X}_n)}}{\sum_{n=1}^{N} e^{\mathcal{E}(\boldsymbol{X})^T \mathcal{E}(\boldsymbol{X}_n)}} = -\log \frac{e^{\boldsymbol{Z}^T \boldsymbol{Z}_n}}{\sum_{n=1}^{N} e^{\boldsymbol{Z}^T \boldsymbol{Z}_n}}, \tag{5}$$

where $\mathcal{E}(\cdot)$: $\mathbb{R}^{L \times M} \to \mathbb{R}^{L \times D}$ denotes an arbitrary encoder. Essentially, in Equation (5), negative sampling is not required since all the negative samples are deterministic, *i.e.,* the remaining decomposed components $\{\boldsymbol{X}_m | m \neq n\}$. Negative sampling is required for most contrastive learning methods, which is yet often tricky, biased and time consuming (Liu et al., 2021). There exist pioneering works to get contrastive learning rid of negative samples in the computer vision field (Grill et al., 2020; Chen & He, 2021). TS-DC gets the contrastive learning rid of additional biases brought by negative sampling and avoids data augmentation in time series representation learning.

3.3 Adaptive Contrasting

Both informative signals and noises exist in decomposed components of the original time series. However, as the decomposition proceeds recursively, it is reasonable to assume that the capability of accounting for the raw time series is decaying to some extent because of the additivity of trend-seasonality decomposition, *i.e.,* $\boldsymbol{X}_n^r = \boldsymbol{X}_{n-1}^r - (\boldsymbol{X}_n^t + \boldsymbol{X}_n^s)$. Therefore, we combine the component-wise contrastive losses to form a new loss:

$$\mathcal{L}_c = \sum_{n=1}^{N} \frac{1}{n^\beta} \mathcal{L}_c^n(\boldsymbol{X}, \boldsymbol{X}_n), \quad \mathcal{L}_c^n \in \mathbb{R}^{L \times D}, \tag{6}$$

where $\beta > 1$ is a constant that controls the decaying speed. Through minimizing the objective in Equation (6), different component-wise contrastive losses can be jointly optimized. Besides, with the monotonically decreasing weights, prior decomposed components will be paid more attention when learning the outcome representation. However, the above configuration of decaying weights is not flexible enough to handle complex and highly diverse time series data.

Weight Adjustment. To implement a data-driven weight adjustment, we can inspect the decomposition at a different angle. The decomposition process can be written as: $\boldsymbol{X}_{1:N} = \mathcal{D}_\theta^N(\boldsymbol{X})$, where $\mathcal{D}_\theta^N(\cdot)$ denotes the decomposition module consisting of N decomposition operations with parameters θ, which means that the decomposition is determined by $\boldsymbol{X}$ and θ: $(\boldsymbol{X}, \theta) \rightarrow \boldsymbol{X}_{1:N}$. On the other hand, according to the Bayes' theorem (Ghosh et al., 2006), the parameters θ satisfy: $p(\theta|\boldsymbol{X}) = p(\boldsymbol{X}|\theta)p(\theta)/p(\boldsymbol{X})$. In other words, θ can be derived from $\boldsymbol{X}$: $\boldsymbol{X} \Rightarrow \theta$. Consequently, the decomposition $\boldsymbol{X}_{1:N}$ is actually determined by the original time series $\boldsymbol{X}$ only. Therefore, we adopt a non-linear function $h_\phi(\cdot)$ to learn correlations between decomposed components and the time series $\boldsymbol{X}$ directly.

$$\boldsymbol{W} = h_\phi(\boldsymbol{X}), \quad \|\boldsymbol{w}_n\|_1 = 1,\, n = 1, 2, \cdots, N, \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_N]^T$. We then rewrite the loss in Equation (6) below:

$$\mathcal{L}_{ac} = \sum_{n=1}^{N} \frac{\boldsymbol{w}_n}{n^\beta} \mathcal{L}_c^n(\boldsymbol{X}, \boldsymbol{X}_n). \tag{8}$$

In this way, through contrasting the outcome representation with the signals in different components jointly and adaptively, the outcome encoder will be encouraged to be aware of multi-faceted patterns in the time series data.

Optimization. Finally, we construct a composite loss and optimize it jointly. To be specific, by combining the alternative of Equation (3), plus Equations (4) and (8), we have

$$\mathcal{L} = \gamma \cdot \mathcal{L}_{KL} + \lambda \cdot \mathcal{L}_{decomp} + \zeta \cdot \textsc{Agg}(\mathcal{L}_{ac})\,, \tag{9}$$

where γ, λ, and ζ are trade-off parameters, and $\textsc{Agg}(\cdot)$ is a aggregator yielding real-value.

4 Experiment

We evaluate the representations learnt by TS-DC on univariate as well as multivariate time series datasets, respectively. Besides, we use the downstream tasks including classification, forecasting and anomaly detection to report the performance comparisons. The results of forecasting and anomaly detection are available in Appendices D and E.

4.1 Experiment Settings

Classification Datasets. The UCR archive (Dau et al., 2019) consisting of 128 datasets is adopted for univariate time series classification, and the UEA archive (Bagnall et al., 2018) (including 30 datasets) is employed for multivariate time series classification. We employ zero-padding for missing values to ensure the time series data are with the same length in a dataset. All the datasets in UCR and UEA archives are split into train and test data already, and the time series in 85 UCR datasets are already z-normalized. We split the train data as 4:1 for training and validation for both UCR and UEA archives. For UCR archive, we apply z-normalization of train data to validation and test data accordingly. While for UEA archive, we adopt an instance-wise z-normalization suggested by (Rakthanmanon et al., 2013) for the training, validation, and testing data, separately.

Baselines. We select six time series representation learning methods as competitors: **1) T-Loss** (Franceschi et al., 2019) uses time-based negative sampling and a triplet loss to obtain general-purpose representations; **2) CPC** (Oord et al., 2018) learns the representations by predicting the future in latent space with autoregressive model; **3) TNC** (Tonekaboni et al., 2020) learns the time series representations by ensuring the distribution of signals within a neighborhood is distinguishable from the distribution of non-neighboring signals; **4) TS2Vec** (Yue et al., 2022) performs contrastive learning over augmented context views to learn the robust contextual representation for each timestamp; **5) TST** (Zerveas et al., 2021) presents a Transformer-based framework for time series representation learning; and **6) TS-TCC** (Eldele et al., 2021) uses a temporal and contextual contrastive method to learn the time series representations. **7) Supervised.** We additionally compare a

Table 1: The averaged scores on 128 UCR datasets and 30 UEA datasets, the best are in boldface and the underline means the second best.

Dataset		TS-DC	TST	TS-TCC	TS2Vec	TNC	CPC	T-Loss	Supervised
UCR	ACC	**0.454**	0.424	0.416	0.382	0.358	0.424	0.416	0.449
	AUPRC	**0.686**	0.649	0.639	0.618	0.510	0.623	0.623	0.666
UEA	ACC	**0.399**	0.375	0.347	0.348	0.311	0.343	0.369	0.385
	AUPRC	**0.569**	0.564	0.546	0.541	0.514	0.551	0.503	0.569

Table 2: Performance comparisons on univariate time series datasets *w.r.t.* classification task.

Method	Coffee		ECG5000		FordB		Ham		Strawberry	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
TST	0.357	0.395	0.881	0.688	0.550	0.522	0.510	0.517	0.723	0.791
TS-TCC	0.607	0.628	0.846	0.735	0.529	0.528	0.563	**0.581**	0.668	0.687
TS2Vec	0.464	0.469	0.599	0.626	0.544	0.557	0.521	0.482	0.625	0.634
TNC	0.535	0.582	0.583	0.512	0.553	0.499	0.542	0.475	0.565	0.577
CPC	0.429	0.446	0.811	0.737	0.511	0.540	**0.573**	0.506	0.710	0.738
T-Loss	0.571	0.671	0.651	0.512	0.539	0.509	0.521	0.536	0.720	0.800
Supervised	0.500	0.531	**0.896**	0.698	0.549	0.547	0.542	0.552	0.801	0.838
TS-DC	**0.785**	**0.710**	0.868	**0.841**	**0.565**	**0.573**	0.531	0.566	**0.864**	**0.919**

supervised method which is implemented with the same encoder adopted by the unsupervised models, but the encoder as well as the downstream model (*e.g.,* classifier or forecaster) are learned synchronously with the supervision signals (see Appendix A.1). Note that, to fairly compare the effectiveness of different TS representation learners, we employ a Transformer-based encoder (Vaswani et al., 2017) as the backbone encoder for both TS-DC and baselines in the experiments, and more implementation details are available in Appendix A.2.

Implementation Details. The dimensionality of representation $\boldsymbol{Z}$ is set to 128. In time series decomposition, a fully connected network (*i.e.,* $\text{FC}(\cdot)$) consisting of 2 nonlinear layers with the hidden size 256 is adopted. The trend-seasonality decomposition operation is carried out 3 times in the experiments (*i.e.,* $N = 3$). The hyperparameter α used in $\mathcal{L}_{KL}$ is 0.5. The constant controlling the decaying weights, *i.e.,* β, is 2. The trade-off parameters in Equation (9) is configured as $\gamma = \lambda = \zeta = 1$. Besides, we construct an RNN-based deep model with the hidden size 128 as the downstream classifier. The non-linear function $h_\phi(\cdot)$ in Equation (7) is a 2-layer fully connected network with the hidden size being 256.

Evaluation Settings. All the models are trained on an Nvidia P40 GPU with CUDA 10.2. More parameter settings employed in the evaluation can be found in Appendix A.3. In addition, we use metrics Accuracy and AUPRC (*i.e.,* area under precision-recall curve) for evaluating classification performance. For both Accuracy and AUPRC, the larger the better.

4.2 Time Series Classification

The averaged Accuracy and AUPRC scores of six unsupervised methods and the supervised method on the UCR and UEA archives can be seen in Table 1. On average, we have achieved 7% accuracy and 6% AUPRC improvements on 128 UCR datasets, and 6% accuracy and 0.9% AUPRC improvements on 30 UEA datasets, respectively. Furthermore, with comparison to the supervised method, TS-DC also achieves comparable performance.

4.2.1 Univariate Time Series Results

The full ACC. results of 128 UCR datasets can be found in Appendix F.1. Here we report the performance of five selected datasets in Table 2 to elaborate the effectiveness of different representation learning methods. The descriptions about these five datasets can be found in Appendix A.4.

TS-DC outperforms most of the UTSRL methods and performs comparably with the supervised method. Besides, we have following observations in Table 2. **1)** TST achieves better performance when the training data is of large amount, which is also the requisite of Transformer-based model. **2)**

Table 3: Performance comparisons on multivariate time series data *w.r.t.* classification task.

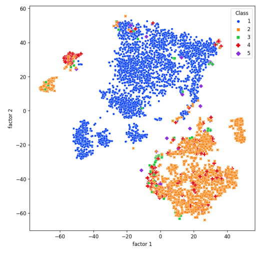
Method	Epilepsy		FaceDetection		Heartbeat		PenDigits		RacketSports	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
TST	0.367	0.528	0.578	0.617	0.667	0.495	0.903	0.985	0.361	0.549
TS-TCC	0.281	0.524	0.517	0.534	0.615	0.467	0.963	0.997	0.333	0.501
TS2Vec	0.313	0.496	0.521	0.516	0.630	**0.503**	0.946	0.991	0.375	0.532
TNC	**0.383**	0.564	0.520	0.518	0.672	0.482	0.647	0.930	0.285	0.509
CPC	0.281	0.535	0.502	0.499	0.641	0.487	0.828	0.962	0.292	0.561
T-Loss	0.344	0.592	0.585	0.605	0.646	0.484	0.889	0.984	0.292	0.557
Supervised	0.328	0.555	0.577	0.602	0.583	0.476	**0.981**	**0.998**	0.333	0.554
TS-DC	0.320	**0.642**	**0.602**	**0.653**	**0.703**	**0.503**	0.978	0.997	**0.458**	**0.728**

Both TS-TCC and TS2Vec use the temporal and contextual information to construct the contrastive loss. TS2Vec utilizes sampled subseries at different scales, while TS-TCC explores the full time series, this difference may account for that TS-TCC performs better than TS2Vec. **3)** TNC highlights the local smoothness in time series data, which might be helpful in handling limited data (*e.g.,* Coffee dataset) or noisy data (*e.g.,* FordB dataset). **4)** CPC and T-Loss are two distinct contrastive methods, the former adopts a probabilistic contrastive loss and the latter applies instance-wise contrastive loss. On ECG5000 dataset, CPC achieves better performance than T-Loss, which demonstrates the strength of probabilistic contrastive loss in modeling instant data.

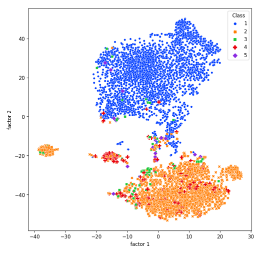
4.2.2 Multivariate Time Series Results

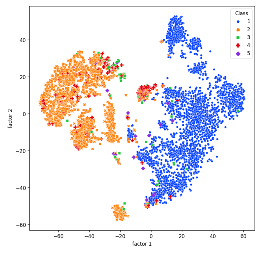
The classification performance on 5 UEA datasets are reported in Table 3 (see Appendix F.2 for full ACC. results). The brief descriptions about these five datasets are available in Appendix A.5.

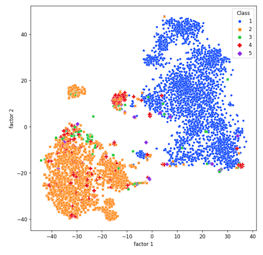
In general, as depicted in Table 3, TS-DC still performs the best most of the time. Besides, it can be seen that, TST achieves very competitive performance toward multivariate time series classification, which can validate the superiority of Transformer-based model on high-dimensional data. T-Loss outperforms CPC *w.r.t.* multivariate time series classification, which demonstrates the limitation of the probabilistic contrastive loss for high dimensional data.

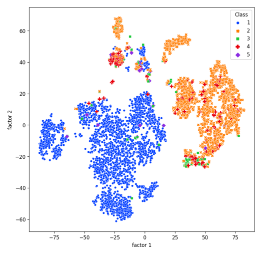


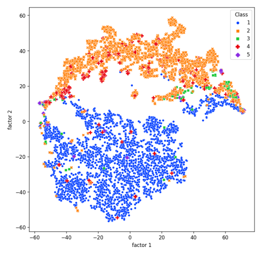
(a) TS-DC.

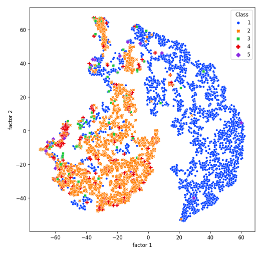

(b) Supervised.


(c) TST.


(d) TS-TCC.


(e) TS2Vec.


(f) TNC.


(g) CPC.

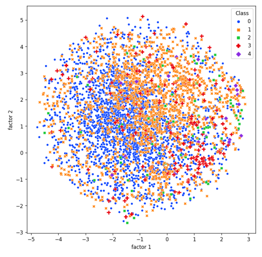

(h) T-Loss.

Figure 3: t-SNE plots of the representations learnt by different methods on the ECG5000 dataset.

4.2.3 Visualizing Learnt Representations

In addition, we employ t-SNE plots to visualize the representations generated by different TS representation methods on ECG5000 dataset in Figure 3. There exist 5 types of time series in ECG5000

Table 4: Ablation study of loss terms in the final objective of TS-DC.

Method	Univariate						Multivariate					
	Strawberry		ECG5000		FordB		Heartbeat		FaceDetection		RacketSports	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
w/o $\mathcal{L}_{KL}$	0.842	0.907	0.822	0.731	0.548	0.550	0.703	0.467	0.605	0.653	**0.458**	**0.775**
w/o $\mathcal{L}_{decomp}$	0.842	0.909	0.812	0.734	0.493	0.527	0.703	0.540	**0.630**	**0.675**	0.451	0.745
w/o $\mathcal{L}_{KL}, \mathcal{L}_{decomp}$	0.840	0.914	0.819	0.707	0.523	0.529	0.703	**0.542**	0.593	0.635	0.465	0.761
w/o $\mathcal{L}_{ac}$	0.826	0.898	**0.879**	0.758	0.548	0.572	0.703	0.497	0.597	0.636	0.319	0.681
TS-DC	**0.864**	**0.919**	0.868	**0.841**	**0.565**	**0.573**	0.703	0.503	0.602	0.653	**0.458**	0.728

dataset. In Figure 3, it can be seen that TS-DC achieves competitive performance comparing to the supervised method with respect to clustering samples in the same category. Besides, for clustering samples of the rare class, *i.e.,* class 3, 4 and 5, TS-DC outperforms most of the alternative unsupervised TS representation learning methods. Another showcase can be found in Appendix C.1.2.

4.3 Model Analysis

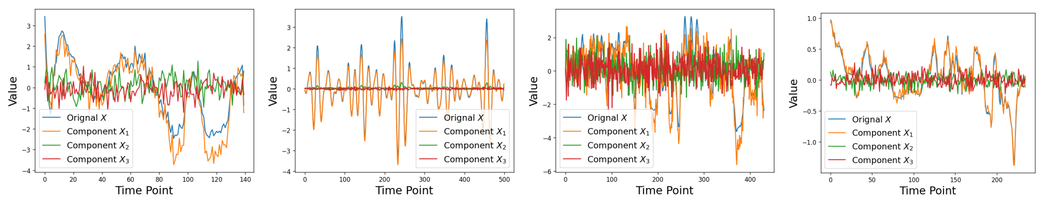
The superior performance achieved toward downstream tasks validates the effectiveness of TS-DC in learning informative TS representations. Here, we explain underlying mechanisms contributing to the performance achieved by TS-DC (the additional parameter sensitivity analysis can be found in Appendix B).

4.3.1 Ablation Study of Loss Terms in Final Objective

We implement tailored TS-DC by omitting $\mathcal{L}_{KL}$, $\mathcal{L}_{decomp}$, and $\mathcal{L}_{ac}$ in Equation (9), respectively. Table 4 reports the corresponding performance comparisons. As depicted, the proposed contrasting mechanism benefits the downstream classifier consistently (refer to w/o $\mathcal{L}_{ac}$). Besides, the performance of univariate TS classification might be lowered if the decomposed components lack expected properties, such as trend-seasonality distinguishability (attributed to $\mathcal{L}_{KL}$) and TS reconstructability (attributed to $\mathcal{L}_{decomp}$). However, for multivariate TS classification, the contributions of decomposition are not good enough, which is probably accounted for the limitation of temporal decomposition on high-dimensional data. Basically, the relative contributions of $\mathcal{L}_{KL}$, $\mathcal{L}_{decomp}$, and $\mathcal{L}_{ac}$ in Equation (9) are not impressive since the loss terms cannot fully reflect the contributions of decomposition and contrasting mechanism of TS-DC. $\mathcal{L}_{KL}$ and $\mathcal{L}_{decomp}$ can enforce the decomposed components behave as expected, and $\mathcal{L}_{ac}$ possesses the contrasting power of TS-DC to a certain extent. In addition, we showcase t-SNE plots of TS-DC and its variants in Appendix C.1.1.

4.3.2 Inspecting Time Series Decomposition

With the time series decomposition, we can dispatch trend and seasonality of a time series into different components, which can describe original time series to varying degrees. We showcase the univariate time series and the decomposed components on ECG5000, FordB, Ham, and Strawberry datasets in Figure 4. We can see that more informative signals are captured by the earlier component, and the first component usually acts as the most informative one that accounts for original time series. As the decomposition proceeds, the latter components contain less informative signals and the level of noise grows as well. Thus, the prior components should often be paid more attention. In



(a) ECG5000. (b) FordB. (c) Ham. (d) Strawberry.

Figure 4: Showcases of the decomposed series yielded by TS-DC on four datasets. Here, the input time series is decomposed into 3 components, *i.e.,* $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3\}$.

Table 5: Ablation study of the adaptive contrasting mechanism in TS-DC.

Method	Univariate						Multivariate					
	Coffee		Ham		FordB		Heartbeat		FaceDetection		RacketSports	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
TS-DC$_{-\boldsymbol{W}}$	0.714	**0.718**	0.531	0.541	0.508	0.511	0.495	**0.602**	0.500	0.501	0.188	0.513
TS-DC$_{\beta=0}$	0.679	0.703	0.427	0.470	0.494	0.495	0.297	0.478	0.497	0.510	0.208	0.544
TS-DC$_{\beta=0,-\boldsymbol{W}}$	0.643	0.705	0.458	0.500	0.525	0.510	0.297	0.481	0.500	0.486	0.125	0.460
TS-DC	**0.785**	0.710	**0.531**	**0.566**	**0.565**	**0.573**	**0.703**	0.503	**0.602**	**0.653**	**0.458**	**0.728**

addition, it is noteworthy that for some cases, the decomposed components behave similarly (*e.g.,* Figure 4c), which can validate the necessity of adaptive contrasting of TS-DC toward representing complex time series data. More results can be found in Appendix C.2.2.

4.3.3 Ablation Study of Adaptive Contrasting

As discussed in Section 4.3.2, the amount of insightful signals contained in different components differs. We combine the monotonically decreasing weights plus a data-driven weight adjustment to realize the adaptive contrasting. Though we showcased the adaptive contrasting is necessary, we still wonder how the monotonically decreasing weights and the data-driven weight adjustment contribute to the overall performance of downstream tasks. We separately train the model without weight adjustment (*i.e.,* TS-DC$_{-\boldsymbol{W}}$), without predefined monotonically decreasing weight (*i.e.,* TS-DC$_{\beta=0}$) and without adaptive contrasting (TS-DC$_{\beta=0,-\boldsymbol{W}}$) to inspect the effects of the decaying weights and the weight adjustment comprehensively. Here we report the classification performance on the univariate time series and multivariate time series datasets in Table 5, the t-SNE plots of the obtained representations can be found in Appendix C.3.1. In particular, the predefined decaying weights play an important role in learning useful representations (see TS-DC$_{-\boldsymbol{W}}$). While the weight-adjustment-only mechanism performs much worse (see TS-DC$_{\beta=0}$), even harms the learnt representations without any adaptive contrasting mechanisms (*i.e.,* TS-DC$_{\beta=0,-\boldsymbol{W}}$), such as, the performance reported on Ham, FordB, and FaceDetection datasets. But the weight adjustment mechanism can greatly improve the effectiveness of the representation learning with the help of predefined decaying weights, which further validates the effectiveness of the adaptive contrasting in TS-DC.

5 Discussion

The challenges encountered in UTSRL are different from those in other domains because of the intricate characteristics in time series data. Besides the common issues of contrastive learning, like sampling bias and learning robustness, how to highlight informative time series patterns effectively is of great importance as well. In this work, we found that through the time series decomposition, the performance of representation learning can be improved with the help of contrastive learning. Due to the nature of non-stationary, the decomposed components of a time series differ from each other. To harness the decomposed components comprehensively, we propose an adaptive contrasting mechanism consisting of weight decaying and weight adjustment. However, it is challenging to learn the best weight configuration for highly diverse time series data. Therefore, a more robust and automatic weighing mechanism need to be explored. In addition, we propose a component-wise contrastive learning method toward UTSRL, while the components with high level noise should be resisted, thus, how to pick the high-quality components is another open problem.

6 Conclusion

In this work, we studied the problem of unsupervised time series representation learning which is beneficial for multiple downstream tasks. To tackle the time series representation learning problem, we proposed a time series decomposition contrastive learning approach, namely TS-DC. Specifically, TS-DC employs trend-seasonality decomposition to construct a serial of components for a time series. Then, a component-wise contrastive loss was proposed, such that the negative sampling is avoided and the informative signals of time series data can be fully captured. Moreover, we devised an adaptive contrasting mechanism to comprehensively leverage the decomposed components in a time series for better representation learning. Extensive experiments on multiple benchmark datasets toward different downstream tasks validate the effectiveness of TS-DC.

7 Reproducibility Statement

The source urls to the baselines and the links to the benchmark datasets used in this work are summarized in 'src-urls.pdf' (in the uploaded supplementary materials). In addition, detailed descriptions about the implementations are listed below. **1) Dataset.** The datasets used for the classification tasks are introduced in Section 4.1, the first paragraph in Section 4.2.1 and Appendix A.5. The forecasting datasets are introduced in Appendix D, and the datasets used for anomaly detection are introduced in Appendix E. **2) Source Code of** TS-DC**.** The source code of the proposed algorithm in this paper is included in the supplementary materials, named 'tsdc.src.zip'. **3) Implementation Details.** The parameter settings of our model can be found in Appendix A.3, and the implementation details of the baseline methods are described in Appendix A.2.

References

Fevzi Alimoglu, Dr Doc, Ethem Alpaydin, and Yagmur Denizhan. Combining multiple classifiers for pen-based handwritten digit recognition. 1996.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *36th International Conference on Machine Learning, ICML 2019*, pp. 9904–9923. International Machine Learning Society (IMLS), 2019.

Reza Asadi and Amelia C Regan. A spatio-temporal decomposition based deep neural network for time series forecasting. *Applied Soft Computing*, 87:105963, 2020.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Romain Briandet, E Katherine Kemsley, and Reginald H Wilson. Discrimination of arabica and robusta in instant coffee by fourier transform infrared spectroscopy and chemometrics. *Journal of agricultural and food chemistry*, 44(1):170–174, 1996.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

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

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

Fangzhou Cheng, Ajay Raghavan, Deokwoo Jung, Yukinori Sasaki, and Yosuke Tajika. High-accuracy unsupervised fault detection of industrial robots using current signal analysis. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–8. IEEE, 2019.

Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee. Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. *arXiv preprint arXiv:1603.00982*, 2016.

Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

Santiago Cuervo, Maciej Grabias, Jan Chorowski, Grzegorz Ciesielski, Adrian Łańcucki, Paweł Rychlikowski, and Ricard Marxer. Contrastive prediction strategies for unsupervised segmentation and categorization of phonemes and words. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3189–3193. IEEE, 2022.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Alexander Dokumentov, Rob J Hyndman, et al. Str: A seasonal-trend decomposition procedure based on regression. *Monash University: Melbourne, Australia*, pp. 23, 2015.

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

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Jayanta K Ghosh, Mohan Delampady, and Tapas Samanta. *An introduction to Bayesian analysis: theory and methods*, volume 725. Springer, 2006.

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.

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

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 215–222. IEEE, 2021.

Eric J Kostelich and Thomas Schreiber. Noise reduction in chaotic time-series data: A survey of common methods. *Physical Review E*, 48(3):1752, 1993.

Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3195–3204, 2021.

Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1939–1947, 2015.

Mingwei Leng, Xinsheng Lai, Guolv Tan, and Xiaohui Xu. Time series representation for anomaly detection. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pp. 628–632. IEEE, 2009.

Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Chengyu Liu, David Springer, Qiao Li, Benjamin Moody, Ricardo Abad Juan, Francisco J Chorro, Francisco Castells, José Millet Roig, Ikaro Silva, Alistair EW Johnson, et al. An open access database for the evaluation of heart sound algorithms. *Physiological measurement*, 37(12):2181, 2016.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge & Data Engineering*, (01):1–1, 2021.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*, 2017.

Raquel Olias, Barbara Maldonado, Pauline Radreau, Gwénaëlle Le Gall, Francis Mulholland, Ian J Colquhoun, and E Katherine Kemsley. Sodium dodecyl sulphate-polyacrylamide gel electrophoresis of proteins in dry-cured hams: Data registration and multivariate analysis across multiple gels. *Electrophoresis*, 27(7):1288–1299, 2006.

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

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Pedro O. Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *Advances in Neural Information Processing Systems*, 33:4489–4500, 2020.

Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–31, 2013.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019a.

Yi Ren, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Almost unsupervised text to speech and automatic speech recognition. In *International Conference on Machine Learning*, pp. 5410–5419. PMLR, 2019b.

Ramin Rezaiifar and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. 06 1995.

Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pp. 843–852. PMLR, 2015.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

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

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jose R Villar, Paula Vergara, Manuel Menéndez, Enrique de la Cal, Víctor M González, and Javier Sedano. Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition. *International journal of neural systems*, 26(06): 1650037, 2016.

Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang. Unispeech: Unified speech representation learning with labeled and unlabeled data. In *International Conference on Machine Learning*, pp. 10937–10947. PMLR, 2021.

Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, Huan Xu, and Shenghuo Zhu. Robuststl: A robust seasonal-trend decomposition algorithm for long time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5409–5416, 2019.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Lingfei Wu, Ian En-Hsu Yen, Jinfeng Yi, Fangli Xu, Qi Lei, and Michael Witbrock. Random warping series: A random features method for time-series embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 793–802. PMLR, 2018.

Zhiwen Xiao, Xin Xu, Huanlai Xing, Shouxi Luo, Penglin Dai, and Dawei Zhan. Rtfn: a robust temporal feature network for time series classification. *Information Sciences*, 571:65–86, 2021.

Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pp. 1055–1062, 2007.

Yahoo! Webscope dataset ydata-labeled-time-series-anomalies-v1_0, 2015. URL `http://labs.yahoo.com/Academic_Relations`.

Ling Yang and Shenda Hong. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*, pp. 25038–25054. PMLR, 2022.

Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pp. 5670–5679. PMLR, 2018.

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

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124, 2021.

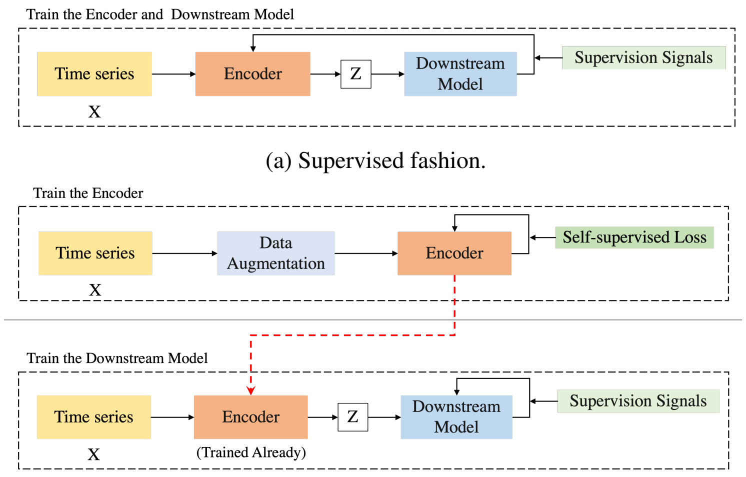
Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. URL `http://jmlr.org/papers/v20/19-011.html`.

A Extra Experiment Settings

In the empirical study, to fairly compare different time series representation learning methods, the same encoder as well as the same downstream algorithm (for classification or forecasting or anomaly detection) is employed to evaluate the effectiveness of learnt representations.

A.1 Train The Encoder: Supervised v.s. Unsupervised

As introduced in the descriptions of baseline settings in Section 4.1, we implement a supervised method as a competitor. In particular, the training procedure of supervised time series representation learner is different from unsupervised time series representation learning, which is demonstrated in Figure 5. For supervised method of TS representation learning, the representation learner (*i.e.,* the Encoder in Figure 5a) is trained together with the downstream model (*e.g.,* a classification model), guided by the supervision signals. Whereas, as depicted in Figure 5b, for UTSRL, the representation learner is trained with the help of self-supervised loss, such as contrastive loss or reconstruction loss. Then, the trained Encoder is adopted directly to generate embeddings, and the downstream model is optimized with the supervision signals independently.



(a) Supervised fashion.

(b) Unsupervised fashion.

Figure 5: Two different ways of training the time series representation learner: supervised and unsupervised.

A.2 Implementation Details

Implementation Details of Unsupervised Baselines.
1) TNC (Tonekaboni et al., 2020) chooses the adjacent segments as the positive samples and non-neighborhood segments as the negative samples. In particular, the size of sampling window is set to be $\lfloor 1/6 \cdot L \rfloor$ where L is the time series length, and the number of negative samples is set as 5. Then, we follow the other settings as recommended. **2)** For **CPC** (Oord et al., 2018) and **T-Loss** (Franceschi et al., 2019), we follow most of the implementation settings introduced in (Tonekaboni et al., 2020). Besides, the number of time steps for CPC is set as $\lfloor 1/10 \cdot L \rfloor$ or 2 if $L < 20$ (L is the time series length). The number of negative samples is 3 for T-Loss and the window size is also set as $1/10 \cdot L$. **3)** For **TS2Vec** (Yue et al., 2022), the temporal_unit is set to 0 and the max_train_length is 3000. **4)** For **TS-TCC** (Eldele et al., 2021) and **TST** (Zerveas et al., 2021), the recommended parameter settings are employed for the evaluation.

Implementation Details of Backbone Encoder and Downstream Classifier.
The encoder used in this paper is implemented following the encoder in Transformer which is built on top of multi-head attention (MHA). In particular, the number of heads is set to 8, and 3 encoder layers are adopted. Besides, the embedding dimension in MHA is set to 128. The encoders in TS-DC are with the same architecture and share the same parameters. For classification, the downstream classifier consists of a 1-layer GRU with hidden size 256 and a 2-layer MLP network with hidden size 256, and the ReLU activation function is adopted.

A.3 More Evaluation Settings

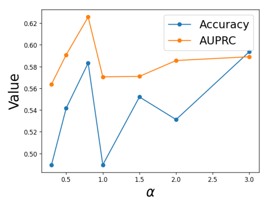
The shared experimental settings in the evaluation are as follows. We utilize an Adam optimizer with learning rate $1e-3$ in the experiments. The batch size is 16 for the datasets that contain more than 100 samples, otherwise, the batch size is set as 2. The training procedure is repeated in 20 epochs at most, and early stopping is adopted where the patience is set to 5.

A.4 Descriptions of The Datasets for Univariate Time Series Classification

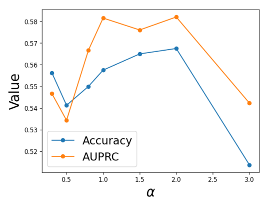
We briefly introduce the selected datasets for elaborating univaraite time series classification results in the main text (refer to Section 4.2.1) as follow. 1) Coffee (Briandet et al., 1996) consists of food spectrographs and contains 28 train samples and 28 test samples with 286 time points; 2) ECG5000 (Goldberger et al., 2000) is the ECG recordings with the series length 140, and there are 500 samples for training and 4500 for testing; 3) FordB has 3636 samples for training and 810 for testing, each time series has 500 time points (the training data were collected in operating conditions, while the test set were collected under noisy conditions); 4) Ham (Olias et al., 2006) composes of the food spectrographs as well, the train/test size is $109/105$; 5) Strawberry is obtained by Fourier-transform infrared spectroscopy (FTIR) including 613 and 370 samples for training and testing, respectively, with 235 time points being collected.

A.5 Description of the Datasets for Multivariate Time Series Classification

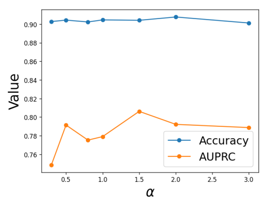
The descriptions of the selected datasets for multivariate time series classification (see Section 4.2.2) are briefly introduced below. **1)** Epilepsy (Villar et al., 2016) was collected from 6 participants using a tri-axial accelerometer on the dominant wrist whilst conducting 4 different activities. This dataset contains 137 training samples and 136 testing samples, and the dimension and length of each sample is 3 and 207, respectively. **2)** FaceDetection is an EEG dataset recorded from 16 different subjects, the train/test size is $5890/3524$, the length and dimension of each sample is 62 and 114, respectively. **3)** Heartbeat (Liu et al., 2016) is the heart sound recording dataset. It includes 204 training samples and 205 testing samples, and the length of each sample is 405 with dimension 61. **4)** PenDigits (Alimoglu et al., 1996) includes 7494 training samples and 3498 testing samples, and each sample contains 8 points where each one is with 2 dimensions. **5)** RacketSports contains 151 training instances and 152 testing instances, for each instance, the time series length is 30 and the dimensionality is 6.



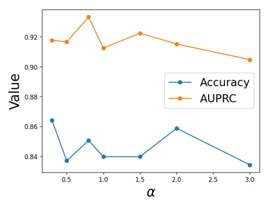
(a) Ham.



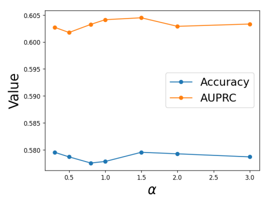
(b) FordB.



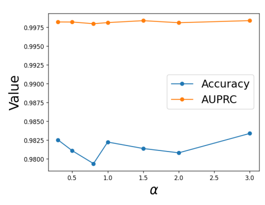
(c) ECG5000.



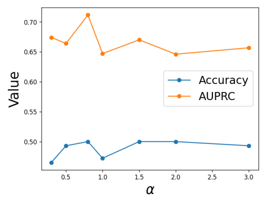
(d) Strawberry.



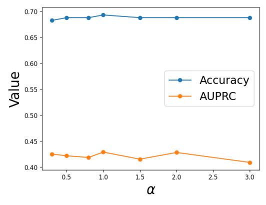
(e) FaceDetection.



(f) PenDigits.



(g) RacketSports.



(h) Heartbeat.

Figure 6: Sensitivity analysis of α in terms of classification performance (Accuracy and AUPRC) on univariate time series (Figures 6a to 6d) and multivariate time series (Figures 6e to 6h).

Table 6: Comparisons of classification performance against different β values.

Method	Univariate						Multivariate					
	ECG5000		Ham		FordB		RacketSports		PenDigits		FaceDetection	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
TS-DC$_{\beta=1}$	**0.890**	0.721	**0.615**	0.532	0.555	0.572	0.438	0.467	0.983	**0.998**	0.573	0.610
TS-DC$_{\beta=2}$	0.868	**0.841**	0.531	**0.566**	0.565	0.573	**0.458**	**0.728**	0.978	0.997	**0.602**	**0.653**
TS-DC$_{\beta=3}$	0.884	0.714	0.583	0.563	0.556	0.572	0.422	0.566	**0.984**	**0.998**	0.576	0.618
TS-DC$_{\beta=4}$	0.885	0.678	0.583	0.536	**0.579**	**0.612**	0.414	0.568	0.976	0.997	0.578	0.622

B Parameter Sensitivity Analysis

In accordance with the experimental results in the main text, we report the parameter sensitivity analysis on the same datasets adopted in Section 4.2.

B.1 Divergence Between Trend and Seasonality in Decomposition

The parameter α in $\mathcal{L}_{KL}$ (see Section 3.1) affects the divergence between the decomposed trend and seasonality of a time series. We analyze the classification performance of the learned representation yielded by TS-DC on different datasets with varying α from 0.1 to 3, which is demonstrated in Figure 6. As depicted (Figures 6a to 6d), univariate time series classification is more sensitive to α, *i.e.,* the KL-divergence between the decomposed trend and seasonality. The probable reason behind is that the KL-divergence between decomposed trend and seasonality of univariate time series data is relative large, which means that the decomposition of univariate time series data is more easier. This can be further validated by examining the effect of α on multivariate time series data (see Figures 6e to 6h), which show that multivariate time series is immune to the time series decomposition.

B.2 The Constant Controlling Decaying Weights

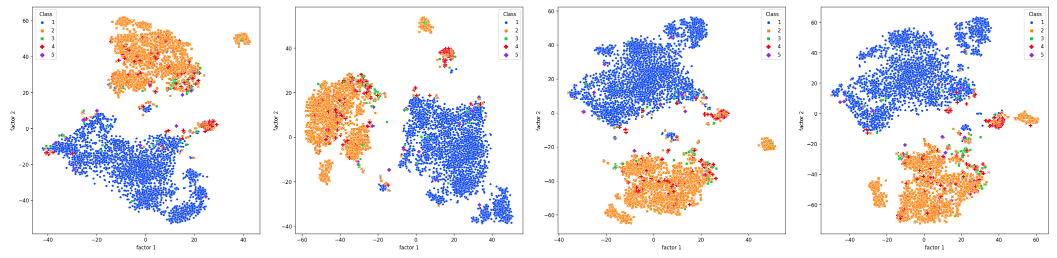
To evaluate the influence of decaying function in adaptive contrasting, we vary the constant β in Equation (6) (refer to Section 3.3 of the main text) and examine the representation learning performance accordingly. We report the classification performance of TS-DC with β ranging in $\{1, 2, 3, 4\}$. The classification performance on both univariate and multivariate datasets are reported in Table 6, where $\beta = 2$ is used in the main experiment. Generally speaking, the effectiveness of the representation learning function is relatively insensitive to β in terms of classification performance. Whereas, the most suitable β on different datasets differs. In particular, when the time series data is of high quality, the decomposed components are distinguishable from each other, a decaying function with conservative β value ($\beta \leq 2$) is preferred. While if the time series is of high level noise, like FordB, a rapid decaying function is required to distinguish the decomposed components, thus, a large β value is beneficial.

C In-Depth Model Analysis

C.1 More Qualitative Analysis

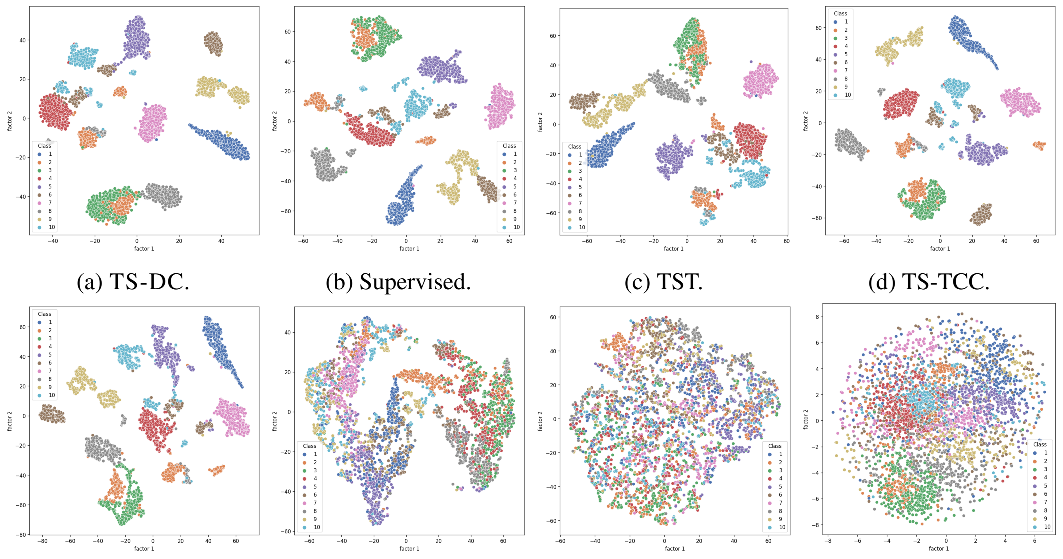
C.1.1 Qualitative Ablation Study of $\mathcal{L}_{KL}$, $\mathcal{L}_{decomp}$, and $\mathcal{L}_{ac}$

We showcase t-SNE plots of TS-DC and its variants by omitting $\mathcal{L}_{KL}$, $\mathcal{L}_{decomp}$, and $\mathcal{L}_{ac}$ in Equation (9), respectively. The clustering results on ECG5000 dataset are demonstrated in Figure 7. As discussed in the main text of Section 4.3.1, the relative contributions of $\mathcal{L}_{KL}$, $\mathcal{L}_{decomp}$, and $\mathcal{L}_{ac}$ are not significant, therefore, the tailored variants of TS-DC can achieve competitive clustering results as well.



(a) TS-DC (b) TS-DC w/o $\mathcal{L}_{KL}$ (c) TS-DC w/o $\mathcal{L}_{decomp}$ (d) TS-DC w/o $\mathcal{L}_{ac}$

Figure 7: Visualizing the embeddings generated by TS-DC and its variants on the ECG5000 dataset. There exist 5 classes in the ECG5000 dataset.



(a) TS-DC. (b) Supervised. (c) TST. (d) TS-TCC.

(e) TS2Vec. (f) TNC. (g) CPC. (h) T-Loss.

Figure 8: t-SNE plots of the representations yielded by different TSRL methods on the PenDigits dataset. There exist 10 classes in the PenDigits dataset.

C.1.2 More Results of Visualizing Learnt Representations

In addition to the case study in Section 4.2.3, we showcase more t-SNE plots on the PenDigits dataset (Figure 8). Frankly speaking, as the time series in PenDigits dataset are not dominated by certain classes, the clustering effect of outcome embeddings yielded by TS-DC is more easy to tell. Besides, TST, TS-TCC, and TS2Vec perform much better than TNC, CPC, and T-Loss.

C.2 Revisit Time Series Decomposition in TS-DC

Figure 9 demonstrates the detailed process of time series decomposition in TS-DC framework, which has been described in the main text of Section 3.1.

C.2.1 How to Learn the TS Decomposition in TS-DC

As introduced in Section 3.1, the decomposition is repeated in a recursive way, which can be formulated as follows:

$$\begin{aligned}\boldsymbol{X}_0^r &= \boldsymbol{X},\\ \boldsymbol{X}_n^r &= \boldsymbol{X}_{n-1}^r - (\boldsymbol{X}_n^t + \boldsymbol{X}_n^s), \quad n = 1, \cdots, N.\end{aligned} \tag{10}$$

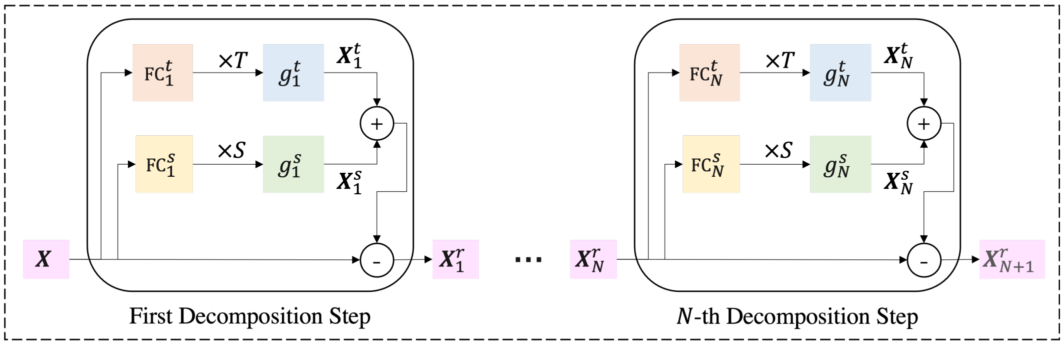


Figure 9: The detailed process of time series decomposition in TS-DC framework.

Algorithm 1 Training procedure of TS decomposition in TS-DC.

```
1:  Input: Original time series $X$, the number of decomposition steps $N$.
2:  Initialize $N$ decomposition modules $\mathbb{H}_{\boldsymbol{\theta}_{1:N}}$.
3:  repeat
4:      $\boldsymbol{X}_0^r = \boldsymbol{X}$
5:      for $n \leftarrow 1$ to $N$ do
6:          $\boldsymbol{X}_n^t, \boldsymbol{X}_n^s = \boldsymbol{H}_{\boldsymbol{\theta}_n}(\boldsymbol{X}_{n-1}^r)$
7:          Calculate $D_{\mathrm{KL}}(p(\boldsymbol{X}_n^t)\|p(\boldsymbol{X}_n^s))$
8:          $\boldsymbol{X}_n^r = \boldsymbol{X}_{n-1}^r - (\boldsymbol{X}_n^t + \boldsymbol{X}_n^s)$
9:      end for
10:     $\mathcal{L}_{decomp} = \|\boldsymbol{X} - \sum_{n=1}^N (\boldsymbol{X}_n^t + \boldsymbol{X}_n^s)\|^2$
11:     $\widetilde{\mathcal{L}}_{KL} = \sum_{n=1}^N D_{\mathrm{KL}}(p(X_n^t)\|p(X_n^s))$; $\mathcal{L}_{KL} = 1/(1+e^{\alpha\widetilde{\mathcal{L}}_{KL}})$
12:     $\boldsymbol{\theta}_{1:N} \leftarrow \arg\min(\mathcal{L}_{KL} + \mathcal{L}_{decomp})$
13: until Convergence
14: Output: Trained $\mathbb{H}_{\boldsymbol{\theta}_{1:N}}$
```

The decomposition process described above absorbs the remainder $\boldsymbol{X}^r$ recursively, such that the complex signals encoded in time series can be fully captured. Though similar to traditional greedy algorithms in signal recovery, like Orthogonal Matching Pursuit (OMP) (Rezaiifar & Krishnaprasad, 1995), the parameters that govern the decomposition process in TS-DC are determined in an end-to-end fashion. Specifically, we construct a time vector $\boldsymbol{T}$ to extract time features, a set of trigonometric functions $\boldsymbol{S}$ as the *orthogonal functions* to utilize the properties of Fourier Transform, and fully connected networks $\mathrm{FC}(\cdot)$ to implement the trend and seasonality extractions. Then, let $\boldsymbol{\theta}_n$ denote the parameter set in the n-th decomposition, and $\mathbb{H}$ denote the set of neural networks $\mathbb{H} = \{\mathrm{FC}_n^t, \mathrm{FC}_n^s, g_n^t, g_n^s\}$, we can rewrite the above decomposition process:

$$\boldsymbol{X}_n^r = \boldsymbol{X}_{n-1}^r - \mathbb{H}_{\boldsymbol{\theta}_n}(\boldsymbol{X}_{n-1}^r), \quad \boldsymbol{X}_n^t, \boldsymbol{X}_n^s = \mathbb{H}_{\boldsymbol{\theta}_n}(\boldsymbol{X}_{n-1}^r). \tag{11}$$

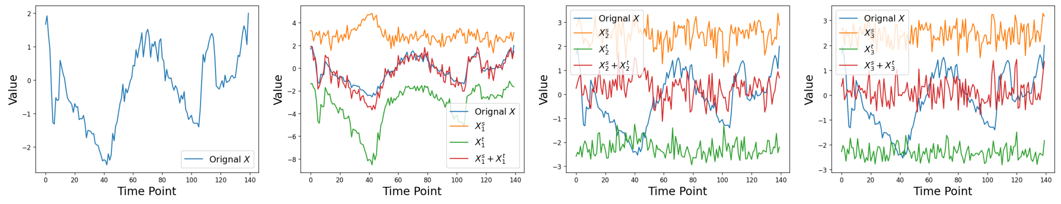
Afterward, with the proposed objectives $\mathcal{L}_{KL}$ and $\mathcal{L}_{decomp}$ (see Section 3.1), the related parameters $\boldsymbol{\theta}_n$ can be learned accordingly:

$$\boldsymbol{\theta}_n \leftarrow \boldsymbol{\theta}_n + \Delta\boldsymbol{\theta}_n = \boldsymbol{\theta}_n - \eta\frac{\partial \mathcal{L}_{\boldsymbol{\theta}_n}}{\partial \boldsymbol{\theta}_n}, \tag{12}$$

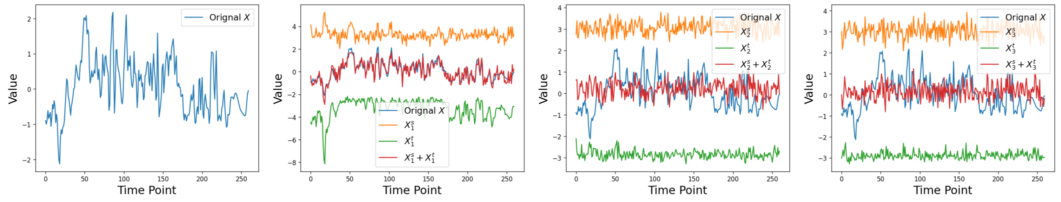
where η is the learning rate and $\mathcal{L}_{\boldsymbol{\theta}_n} = \mathcal{L}_{KL} + \mathcal{L}_{decomp}$. For more details of the training procedure, please refer to Algorithm 1.

C.2.2 More Visualization Results of TS Decomposition

We separately plot the decomposed trend and seasonality in each decomposition step for ECG5000 and Coffee datasets, respectively. The plots are demonstrated in Figure 10. We can see that, in the first decomposition step, the combination of trend and seasonality can recover the original series quite well. As the decomposition proceeds, less informative signals can be observed in the decomposed trend and seasonality. Besides, the decomposed trend prefer to be conservative in recovering original time series, while seasonality most likely to be more radical.

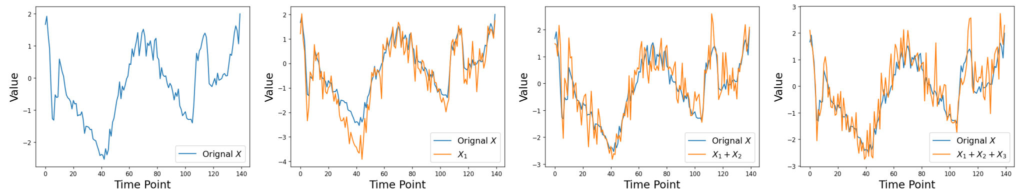


(a) ECG5000. (b) 1-st decomposition. (c) 2-nd decomposition. (d) 3-rd decomposition.

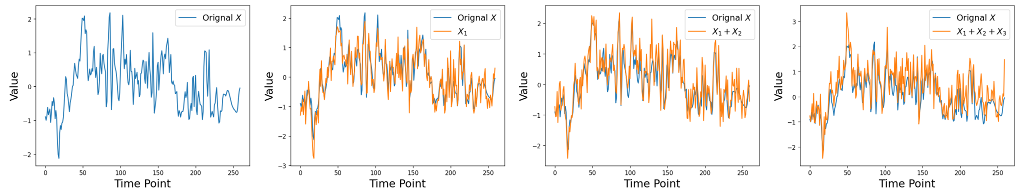


(e) Coffee. (f) 1-st decomposition. (g) 2-nd decomposition. (h) 3-rd decomposition.

Figure 10: The visualization of trend and seasonality in each decomposition step on ECG5000 and Coffee datasets. The trend and seasonality of n-th decomposition are represented as X_n^t and X_n^s.



(a) ECG5000. (b) $N = 1$. (c) $N = 2$. (d) $N = 3$.



(e) Coffee. (f) $N = 1$. (g) $N = 2$. (h) $N = 3$.

Figure 11: Showcases of the original TS and recovered TS when the decomposition is repeated in varying steps. The recovered TS is realized by fusing the decomposed components.

Moreover, since the information contained in the remainder keeps being lowered as the decomposition proceeds, the trend and seasonality in following decomposition steps become more noisier. In Figure 11, we plot the original TS as well as the recovered TS, and the recovered TS is realized by fusing the decomposed components. It can be seen that, as the number of decomposition steps climbs up, the recovered time series fit the original TS better yet more noises come along. The existence of noise does negatively affect the TS decomposition. However, this is beneficial for contrastive learning, since the robustness of the TS representation learning can be enhanced.

C.2.3 Time Series Classification Results In Terms Of N

In Table 7, we investigate the effectiveness of outcome representations yielded by TS-DC toward the time series classification task. In particular, we report classification scores against the number of components to be decomposed (*i.e.,* N) in TS-DC framework. We can see that, with regards to TS classification, the right number of components differs case by case. Therefore, it is important to determine the decomposition level for different time series data in a more heuristic and intelligent manner.

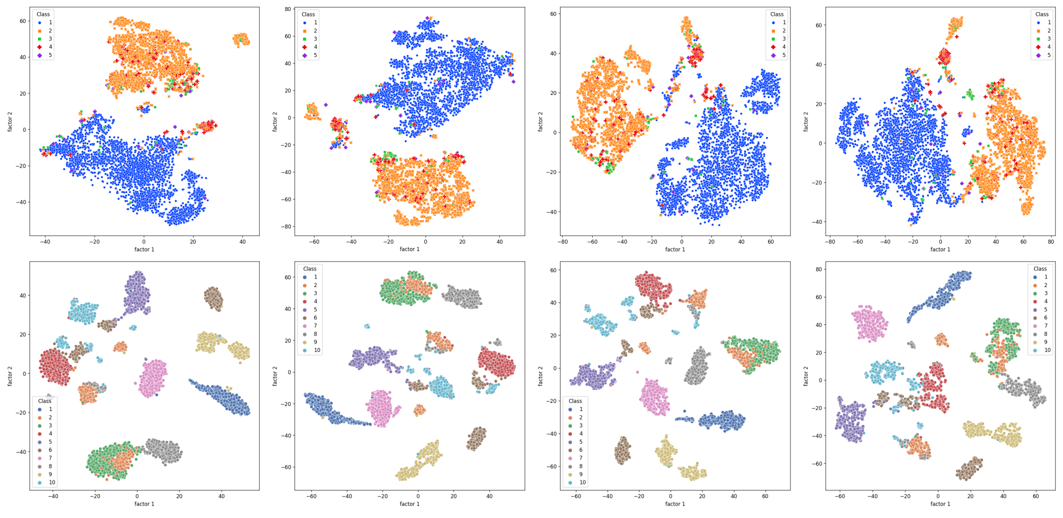
Table 7: Comparisons of classification performance regarding time series decomposition with different number of components decomposed (*i.e.,* N).

# Components	Univariate						Multivariate					
	ECG5000		FordB		Ham		Epilepsy		FaceDetection		RacketSports	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
N = 1	0.856	0.723	0.520	0.555	0.573	0.641	0.382	0.609	**0.615**	**0.667**	0.410	0.740
N = 2	0.846	0.759	0.523	0.535	**0.625**	0.638	0.297	0.662	0.608	0.654	0.438	**0.788**
N = 3	**0.868**	**0.841**	**0.565**	**0.573**	0.531	0.566	0.320	0.642	0.602	0.653	**0.458**	0.728
N = 4	0.831	0.736	0.506	0.536	0.572	**0.645**	**0.430**	**0.762**	0.604	0.645	0.299	0.586

C.3 Further Analysis of Adaptive Contrasting

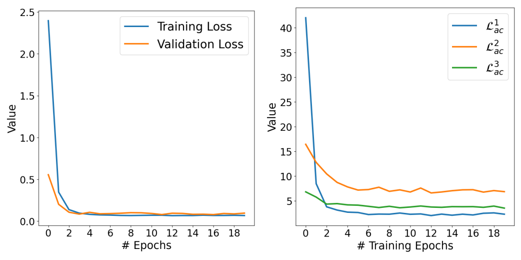
C.3.1 Ablation Study of Adaptive Contrasting: Qualitative Analysis

In addition to the quantitative ablation study of the adaptive contrasting of TS-DC (see Section 4.3.3 in the main text), we visualize the representations learned by TS-DC and its variants on univariate and multivariate time series datasets, respectively, in Figure 12. In particular, we visualize the t-SNE plots of the representations learned by TS-DC and its variants, including TS-DC w/o weight adjustment (TS-DC$_{-\boldsymbol{W}}$), TS-DC w/o predefined monotonically decreasing weights (TS-DC$_{\beta=0}$) and TS-DC w/o adaptive contrasting (TS-DC$_{\beta=0,-\boldsymbol{W}}$). The visualizations are carried out on ECG5000 (univariate TS) and PenDigits (multivariate TS) datasets, respectively. We can observe that, the representations yielded by TS-DC$_{-\boldsymbol{W}}$ are less distinguishable, while more distinguishable representations can be learned by TS-DC$_{\beta=0}$ yet too excessive (even the samples in the same category cannot be grouped together). This is consistent with the analysis of Section 4.3.3 in the main text. Moreover, the representations learned by TS-DC$_{\beta=0,-\boldsymbol{W}}$ mistakenly group the samples in different classes and separate the samples in the same category. Therefore, not only weight decaying but also weight adjustment are helpful in contributing to the outcome representation learning.

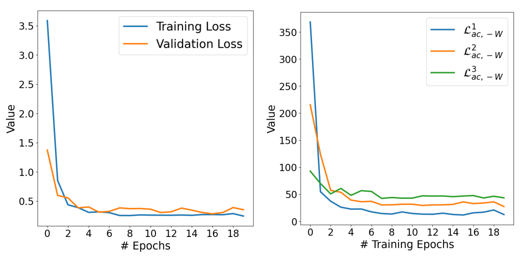


(a) TS-DC (b) TS-DC$_{-\boldsymbol{W}}$ (c) TS-DC$_{\beta=0}$ (d) TS-DC$_{\beta=0,-\boldsymbol{W}}$

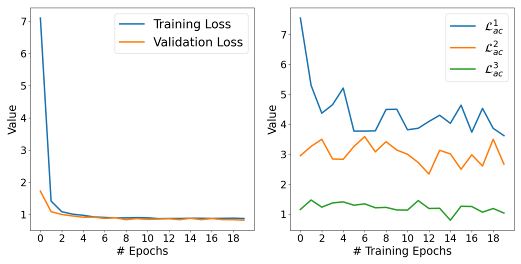
Figure 12: Ablation study of adaptive contrasting in TS-DC on univariate ECG5000 (top row) and multivariate PenDigits (bottom row) time series datasets. There exist 5 and 10 classes in ECG5000 and PenDigits datasets, respectively.



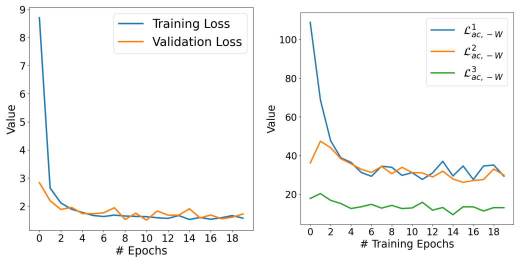
(a) w/ weight adjustment (on ECG5000 dataset).



(b) w/o weight adjustment (on ECG5000 dataset).



(c) w/ weight adjustment (on Ham dataset).



(d) w/o weight adjustment (on Ham dataset).

Figure 13: Learning curves w/ and w/o weight adjustment. The learning curves of the original time series (left) and the decomposed components (right) are plotted, respectively.

C.3.2 Weight Adjustment Analysis: Impacts on Learning Procedure

The weight adjustment in TS-DC framework can not only weigh different components in the time series data but also handle the heterogeneity among different types of time series, and thus improve the robustness of representation learning. Here we analyze the impacts of the weight adjustment on the training procedure of TS-DC by inspecting the learning curves w/ and w/o the weight adjustment. In Figure 13, we plot the curves of contrastive losses of decomposed components $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N\}$ where $N = 3$, and compare the loss curves w/ and w/o the weight adjustment, denoted by $\mathcal{L}^n_{ac}$ and $\mathcal{L}^n_{ac,-\boldsymbol{W}}$ $(n = 1, 2, 3)$. Besides, the curves of the original training loss and validation loss are also plotted. In general, the weight adjustment improves the stability of the overall learning procedure as the validation loss decreases smoothly. In addition, the curves of different decomposed components behave more discriminatively when the weight adjustment is introduced.

C.4 Contrasting with Trend/Seasonality at Different Decomposition Levels

Rather than contrasting the decomposed component at different levels, in this experiment, we attempt to rebuild a new contrastive loss by contrasting trend/seasonality at different levels. Specifically, at each decomposition level $n \in \{1, 2, \cdots, N\}$, the decomposed trend/seasonality is treated as the positive and the corresponding seasonality/trend as the negative. For instance, if we contrast trend $\boldsymbol{X}^t_n$ (positive) with $\boldsymbol{X}^s_n$ (negative), then the loss function in Equation (5) can be rewritten as:

$$\widetilde{\mathcal{L}}^n_c(\boldsymbol{X}, \boldsymbol{X}^t_n, \boldsymbol{X}^s_n) = -\log \frac{e^{\mathcal{E}(\boldsymbol{X})^T \mathcal{E}(\boldsymbol{X}^t_n)}}{e^{\mathcal{E}(\boldsymbol{X})^T \mathcal{E}(\boldsymbol{X}^t_n)} + e^{\mathcal{E}(\boldsymbol{X})^T \mathcal{E}(\boldsymbol{X}^s_n)}}\,, \tag{13}$$

then the corresponding objective for adaptive contrasting is:

$$\widetilde{\mathcal{L}}_{t^+/s^-} = \sum_{n=1}^{N} \frac{\boldsymbol{w}_n}{n^\beta} \widetilde{\mathcal{L}}^n_c(\boldsymbol{X}, \boldsymbol{X}^t_n, \boldsymbol{X}^s_n)\,. \tag{14}$$

Vice versa, we can build an alternative of Equation (14) by contrasting seasonality $\boldsymbol{X}^s_n$ with $\boldsymbol{X}^t_n$ (denoted by $\widetilde{\mathcal{L}}_{s^+/t^-}$). Afterward, we compare the classification performance of TS-DC on univariate and multivariate time series datasets with different contrastive loss functions and report corresponding experiment results in Table 8. We can see that, in terms of AUPRC, the classification performance on Ham and Epilepsy datasets can be improved with the new contrasting mechanism, which

Table 8: Performance comparisons toward univariate and multivariate time series classification. TS-DC$_{t+/s-}$ denotes the variant of TS-DC by replacing original adaptive contrasting mechanism with Equation (14), and TS-DC$_{s+/t-}$ is an alternative that contrasts seasonality with trend.

Method	Univariate						Multivariate					
	ECG5000		FordB		Ham		Epilepsy		FaceDetection		RacketSports	
	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC	ACC	AUPRC
TS-DC$_{t+/s-}$	0.907	0.764	0.553	0.569	0.583	0.639	0.398	0.622	0.591	0.615	0.382	0.649
TS-DC$_{s+/t-}$	0.915	0.817	0.545	0.567	0.572	0.586	0.406	0.671	0.597	0.632	0.431	0.729
TS-DC	0.868	0.841	0.565	0.573	0.531	0.566	0.320	0.642	0.602	0.653	0.458	0.728

mainly indicates that a certain temporal pattern (trend or seasonality) in these two datasets should be paid more attentions.

D TIME SERIES FORECASTING RESULTS

In this section, we evaluate the effectiveness of representation learning towards the time series forecasting task. We conduct the time series forecasting with the learnt TS representation functions on three publicly available datasets: Traffic [1], Weather [2] and Electricity [3]. In particular, Traffic dataset includes 862 variables, and it is a collection of 48 months (2015-2016) hourly recorded data from the California Department of Transportation; Electricity dataset collects the electricity consumption of 321 clients from 2012 to 2014. Weather dataset was recorded between 2020-2021 every 10 minutes including 21 weather indicators. We split the training/validation/test set as 7:1:2 for these three datasets, and three prediction lengths (*i.e.,* 32, 64 and 128) are tested. To construct the instance set suitable for TS forecasting, an input-L_x-predict-L_y sliding window is adopted to roll the train, validation and test sets with stride 1 time step, respectively, which is illustrated in Figure 14. In the experiment, the input length is the same with the prediction length, *i.e.,* $L_x = L_y$. Besides, we employ MSE and MAE as the evaluation metrics. Moreover, the same settings of the representation learning for the classification task (see Appendices A.1 to A.3) are adopted.

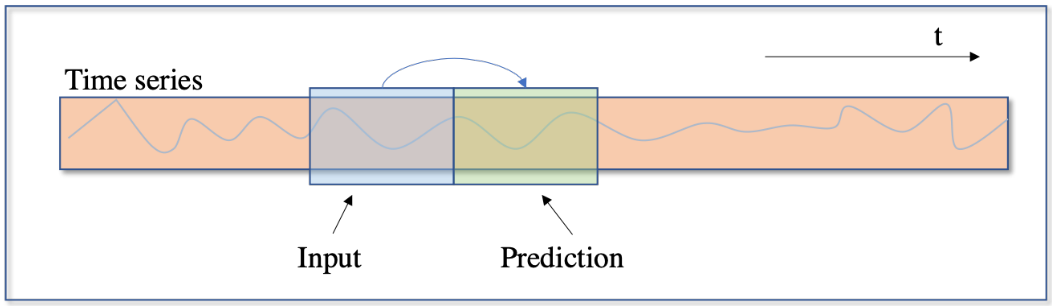


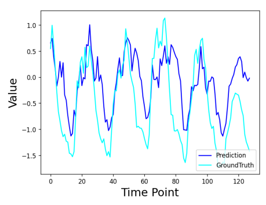
Figure 14: The schematic diagram of the time series forecasting. An input-L_x-predict-L_y sliding window is adopted to roll the series with stride 1 time step.

The forecasting results are reported in Table 9, which shows that TS-DC consistently achieves better forecasting results compared to other baseline representation learning methods. Moreover, TST also performs well toward the time series forecasting task, which can demonstrate the superiority of the Transformer-based model on the high-dimension data among these baseline contrastive learning methods. TS-TCC and TS2Vec both stress the temporal and contextual patterns of the time series data, and the superior forecasting performance than other representation learning methods highlight

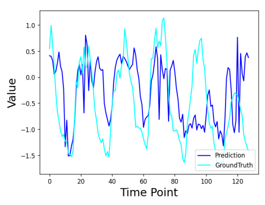
[1]http://pems.dot.ca.gov
[2]https://www.bgc-jena.mpg.de/wetter/
[3]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

Table 9: Comparisons of different representation learning methods in terms of multivariate time series forecasting performance.

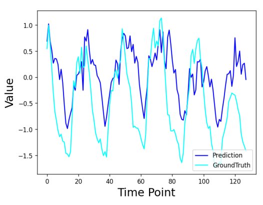
Dataset		TS-DC		Supervised		TST		TS-TCC		TS2Vec		TNC		CPC		T-Loss	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Traffic	32	0.427	0.769	**0.400**	**0.724**	0.447	0.820	0.489	0.887	0.791	1.406	0.970	2.092	0.574	1.064	0.777	1.411
	64	0.446	**0.801**	**0.430**	0.802	0.457	0.836	0.607	1.055	0.762	1.419	0.818	1.467	0.571	1.071	0.730	1.305
	128	0.465	0.856	**0.441**	**0.838**	0.463	0.864	0.572	1.049	0.783	1.431	0.935	1.789	0.533	0.980	0.628	1.165
Electricity	32	**0.389**	**0.304**	0.457	0.436	0.497	0.498	0.550	0.572	0.719	0.856	1.079	1.801	0.762	0.926	0.647	0.731
	64	**0.405**	**0.330**	0.460	0.444	0.484	0.475	0.669	0.770	0.760	0.906	1.777	5.092	0.608	0.663	0.685	0.793
	128	**0.429**	**0.348**	0.473	0.473	0.486	0.474	0.728	0.888	0.798	0.970	1.857	5.961	0.649	0.737	0.670	0.771
Weather	32	**0.444**	**0.551**	0.807	1.368	0.844	1.426	0.837	1.348	0.838	1.432	0.896	1.548	0.902	1.587	0.861	1.442
	64	**0.509**	**0.682**	0.785	1.276	0.792	1.313	0.809	1.302	0.825	1.406	0.881	1.510	0.899	1.570	0.840	1.410
	128	**0.483**	**0.546**	0.706	1.058	0.703	1.004	0.720	0.995	0.811	1.348	0.811	1.278	0.929	1.660	0.819	1.322



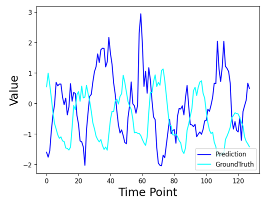
(a) TS-DC.



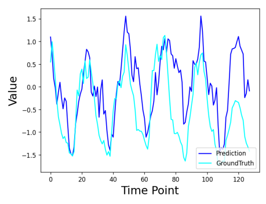
(b) Supervised.



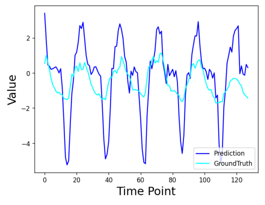
(c) TST.



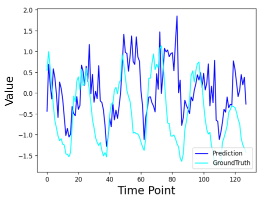
(d) TS2Vec.



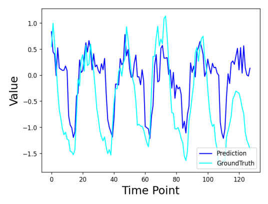
(e) TS-TCC.



(f) TNC.



(g) T-Loss.



(h) CPC.

Figure 15: A showcase of TS-DC and six baseline unsupervised representation learning methods plus one supervised method on Electricity dataset. The target value is the "MT_321" dimension of the Electricity dataset and the prediction length is 128.

the important role of temporal pattern modeling in forecasting. Due to the limited capability of exploring the essential characteristics of time series data, like temporal dependencies, CPC and TNC perform not satisfactory for the time series forecasting. Moreover, TNC performs worse than other baselines, which suggests that the signals focusing on local smoothness hinders the model foreseeing the future changes, thus limits the capacity of forecasting.

Showcases of forecasting results. We additionally showcase the results of different representation learning methods toward time series forecasting on Electricity dataset, which is shown in Figure 15. Compared to other unsupervised representation learning methods as well as the supervised representation learning method, TS-DC represents the trends and peaks more accurately, which further validate the superiority of TS-DC in capturing the temporal dynamics that are helpful in forecasting the future time series.

E TIME SERIES ANOMALY DETECTION RESULTS

To evaluate the time series representation learning for the time series anomaly detection task, we conduct experiments on two public time series anomaly detection datasets: KPI (Ren et al., 2019a) and Yahoo (Yahoo!, 2015). KPI dataset was released by AIOPS challenge in 2017. This dataset

Table 10: The anomaly detection results on KPI and Yahoo datasets. The best and the 2nd best scores are boldfaced and underlined, respectively.

Method	KPI					Yahoo				
	F1	Recall	Precision	ACC	AUC	F1	Recall	Precision	ACC	AUC
CPC	0.0659	0.3045	0.0413	0.8902	0.6035	0.0077	0.0531	0.0041	0.8931	0.4047
T-Loss	0.1236	0.5168	0.0767	0.8973	0.7114	<u>0.0169</u>	<u>0.1172</u>	<u>0.0091</u>	<u>0.8941</u>	0.5116
TNC	0.1132	0.4439	0.0715	0.8963	0.6747	0.0159	0.1043	0.0081	0.8939	0.5094
TST	0.1281	0.5104	<u>0.0802</u>	<u>0.8980</u>	0.7084	0.0168	0.1163	0.0090	0.8940	**0.5242**
TS-TCC	<u>0.1314</u>	<u>0.5246</u>	**0.0823**	**0.8984**	0.7156	0.0160	0.1108	0.0086	0.8940	<u>0.5225</u>
TS2Vec	0.0656	0.2882	0.0408	0.8901	0.5952	0.0164	0.1136	0.0088	0.8940	<u>0.5225</u>
TS-DC	**0.1354**	**0.5299**	0.0801	0.8975	**0.7182**	**0.0172**	**0.1190**	**0.0092**	**0.8941**	0.5096

includes 29 different minutely sampled KPI curves from various Internet companies. Yahoo dataset is an open anomaly detection dataset released by Yahoo Lab. This dataset contains 367 hourly sampled time series with anomaly points being labeled.

The settings of constructing training/validation/test sets are as follows. For the KPI dataset, we follow the training/test split setting of released data, and further split each time series in the training set as 4:1 following the time order to form the training and validation sets accordingly. For each time series in Yahoo dataset, we chronically split a time series as 7:1:2 and construct the training, validation and test sets. Subsequently, we employ a streaming evaluation protocol. To be specific, given a time series in training/validation sets, a length l sliding window is utilized to roll the series with stride d time steps. In the evaluation, we set $l = 128$ and $d = 12$. Then, given a sliced time series with fixed length l, for each time point within this sliced series, we predict whether a point is anomaly or not. Besides, a time series in the test set is evaluated without being sliced.

The representation learning is conducted on training and validation sets, and the dimensionality of the latent representation is 32 and the batch size is 100. The other settings of the representation learning for TS-DC and baselines are the same with the descriptions in Appendices A.1 to A.3. Then, a learned time series representing function takes the time series data in the test set as input and encodes them accordingly. Afterward, the outcome representations are fed into a downstream anomaly detector. To fairly compare the effectiveness of different representation learning methods, we employ the same anomaly detection method for different representation learning models. We take ECOD[4] as the anomaly detector (Zhao et al., 2019; Li et al., 2022) to evaluate the quality of obtained representations yielded by different methods. The evaluation metrics including accuracy, precision, recall, F1 and AUC are reported. The results can be seen in Table 10.

We can see that TS-DC consistently outperforms baseline representation learning models, while the margins are not considerable. In particular, the poor performance of CPC suggests that the forecasting capacity relying on the temporal dependencies does not help in anomaly detection task. Basically, the overall anomaly detection performance of the representation learning algorithms are not impressive. The probable reason is that the current time series representation learning models mostly emphasize the macroscopic signals, which is immune to the rare, sparse and disordered outliers. Therefore, it is still an open question for unsupervised time series representation learning to fuel the downstream anomaly detection task.

[4]https://github.com/yzhao062/pyod.

F Full Time Series Classification Results

F.1 Full Results of Univariate Time Series Classification

The full results of univariate time series classification on 128 UCR datasets can be found in Table 11. In general, the averaged accuracy of TS-DC outperforms other baseline unsupervised time series representation learning methods. Besides, TS-DC even outperforms the supervised method in terms of averaged accuracy.

Table 11: Full accuracy results of univariate time series classification on 128 UCR datasets.

Dataset	TS-DC	Supervised	TS2Vec	TS-TCC	TST	TNC	CPC	T-Loss
Adiac	0.195	0.292	0.096	0.102	0.229	0.099	0.216	0.169
ArrowHead	0.374	0.414	0.362	0.402	0.443	0.31	0.333	0.437
Beef	0.333	0.367	0.233	0.233	0.4	0.267	0.3	0.267
BeetleFly	0.65	0.650	0.55	0.55	0.7	0.65	0.6	0.7
BirdChicken	0.55	0.550	0.5	0.55	0.5	0.5	0.55	0.55
Car	0.317	0.283	0.35	0.3	0.25	0.4	0.317	0.283
CBF	0.438	0.496	0.501	0.483	0.527	0.521	0.457	0.511
ChlorineConcentration	0.548	0.564	0.555	0.549	0.562	0.554	0.561	0.547
CinCECGTorso	0.37	0.437	0.243	0.285	0.292	0.253	0.278	0.338
Coffee	0.786	0.500	0.464	0.607	0.357	0.536	0.429	0.571
Computers	0.642	0.371	0.475	0.527	0.471	0.5	0.478	0.479
CricketX	0.224	0.268	0.266	0.216	0.211	0.182	0.214	0.221
CricketY	0.245	0.310	0.253	0.203	0.245	0.214	0.271	0.25
CricketZ	0.221	0.299	0.271	0.201	0.216	0.167	0.281	0.237
DiatomSizeReduction	0.578	0.611	0.327	0.575	0.549	0.258	0.592	0.592
DistalPhalanxOutlineAgeGroup	0.523	0.539	0.469	0.633	0.523	0.477	0.57	0.492
DistalPhalanxOutlineCorrect	0.728	0.676	0.68	0.688	0.676	0.658	0.656	0.669
DistalPhalanxTW	0.609	0.648	0.461	0.641	0.664	0.352	0.586	0.531
Earthquakes	0.813	0.789	0.758	0.805	0.813	0.758	0.797	0.789
ECG200	0.708	0.792	0.719	0.667	0.708	0.74	0.74	0.813
ECG5000	0.868	0.896	0.599	0.846	0.882	0.583	0.811	0.651
ECGFiveDays	0.547	0.594	0.533	0.651	0.623	0.516	0.68	0.606
ElectricDevices	0.695	0.497	0.291	0.309	0.558	0.257	0.213	0.341
FaceAll	0.142	0.146	0.1	0.163	0.174	0.085	0.098	0.114
FaceFour	0.284	0.352	0.295	0.295	0.33	0.261	0.261	0.33
FacesUCR	0.304	0.403	0.227	0.3	0.315	0.192	0.338	0.299
FiftyWords	0.201	0.230	0.163	0.185	0.225	0.125	0.234	0.183
Fish	0.338	0.369	0.25	0.313	0.381	0.219	0.369	0.325
FordA	0.55	0.532	0.544	0.524	0.521	0.501	0.536	0.51
FordB	0.565	0.549	0.544	0.529	0.55	0.553	0.511	0.539
GunPoint	0.713	0.767	0.673	0.673	0.593	0.58	0.673	0.593

Dataset	TS-DC	Supervised	TS2Vec	TS-TCC	TST	TNC	CPC	T-Loss
Ham	0.531	0.542	0.521	0.563	0.51	0.542	0.573	0.521
HandOutlines	0.796	0.780	0.704	0.802	0.69	0.674	0.795	0.777
Haptics	0.27	0.309	0.296	0.303	0.329	0.257	0.326	0.283
Herring	0.594	0.578	0.594	0.594	0.578	0.469	0.516	0.625
InlineSkate	0.186	0.184	0.189	0.175	0.165	0.176	0.217	0.169
InsectWingbeatSound	0.172	0.197	0.211	0.194	0.188	0.147	0.184	0.194
ItalyPowerDemand	0.836	0.933	0.58	0.877	0.899	0.619	0.907	0.804
LargeKitchenAppliances	0.467	0.318	0.353	0.321	0.332	0.389	0.335	0.326
Lightning2	0.633	0.633	0.533	0.667	0.6	0.75	0.6	0.55
Lightning7	0.431	0.444	0.292	0.333	0.5	0.264	0.5	0.431
Mallat	0.228	0.277	0.183	0.21	0.256	0.227	0.169	0.215
Meat	0.383	0.383	0.433	0.383	0.333	0.483	0.417	0.383
MedicalImages	0.523	0.540	0.517	0.508	0.528	0.513	0.522	0.525
MiddlePhalanxOutlineAgeGroup	0.278	0.368	0.153	0.403	0.361	0.292	0.383	0.375
MiddlePhalanxOutlineCorrect	0.601	0.573	0.594	0.542	0.656	0.573	0.625	0.611
MiddlePhalanxTW	0.514	0.535	0.382	0.521	0.542	0.361	0.516	0.549
MoteStrain	0.704	0.769	0.645	0.767	0.744	0.595	0.757	0.669
NonInvasiveFetalECGThorax1	0.306	0.528	0.212	0.379	0.385	0.134	0.324	0.322
NonInvasiveFetalECGThorax2	0.444	0.657	0.172	0.481	0.552	0.11	0.466	0.484
OliveOil	0.533	0.467	0.4	0.467	0.367	0.367	0.5	0.367
OSULeaf	0.263	0.358	0.188	0.267	0.292	0.204	0.272	0.292
PhalangesOutlinesCorrect	0.642	0.657	0.636	0.647	0.662	0.627	0.657	0.642
Phoneme	0.108	0.087	0.064	0.034	0.066	0.081	0.055	0.075
Plane	0.344	0.583	0.313	0.396	0.427	0.302	0.406	0.458
ProximalPhalanxOutlineAgeGroup	0.818	0.844	0.563	0.875	0.802	0.448	0.859	0.786
ProximalPhalanxOutlineCorrect	0.698	0.726	0.688	0.729	0.705	0.677	0.694	0.694
ProximalPhalanxTW	0.74	0.734	0.396	0.719	0.74	0.37	0.719	0.75
RefrigerationDevices	0.435	0.326	0.288	0.351	0.353	0.351	0.366	0.337
ScreenType	0.34	0.370	0.356	0.372	0.391	0.302	0.375	0.367
ShapeletSim	0.483	0.489	0.5	0.511	0.528	0.45	0.494	0.522
ShapesAll	0.047	0.074	0.017	0.063	0.074	0.027	0.085	0.041
SmallKitchenAppliances	0.429	0.353	0.348	0.397	0.348	0.454	0.355	0.332
SonyAIBORobotSurface1	0.433	0.592	0.568	0.535	0.542	0.422	0.578	0.602
SonyAIBORobotSurface2	0.712	0.734	0.639	0.727	0.714	0.542	0.739	0.724
StarLightCurves	0.844	0.883	0.779	0.81	0.827	0.677	0.843	0.815
Strawberry	0.864	0.802	0.625	0.668	0.723	0.565	0.71	0.72
SwedishLeaf	0.558	0.546	0.171	0.329	0.479	0.196	0.438	0.439
Symbols	0.467	0.438	0.248	0.339	0.387	0.338	0.465	0.387
SyntheticControl	0.42	0.486	0.385	0.378	0.444	0.292	0.365	0.403

Dataset	TS-DC	Supervised	TS2Vec	TS-TCC	TST	TNC	CPC	T-Loss
ToeSegmentation1	0.5	0.474	0.68	0.509	0.461	0.5	0.496	0.544
ToeSegmentation2	0.746	0.431	0.708	0.508	0.423	0.5	0.469	0.438
Trace	0.479	0.750	0.5	0.531	0.635	0.531	0.667	0.604
TwoLeadECG	0.567	0.579	0.514	0.559	0.573	0.49	0.536	0.551
TwoPatterns	0.51	0.667	0.371	0.627	0.532	0.408	0.638	0.62
UWaveGestureLibraryAll	0.378	0.460	0.224	0.309	0.41	0.234	0.345	0.331
UWaveGestureLibraryX	0.462	0.489	0.473	0.349	0.446	0.334	0.416	0.404
UWaveGestureLibraryY	0.426	0.490	0.457	0.374	0.459	0.301	0.426	0.393
UWaveGestureLibraryZ	0.43	0.508	0.454	0.341	0.425	0.293	0.419	0.401
Wafer	0.944	0.955	0.933	0.941	0.94	0.899	0.952	0.927
Wine	0.481	0.389	0.537	0.556	0.5	0.426	0.481	0.426
WordSynonyms	0.253	0.303	0.287	0.288	0.296	0.218	0.291	0.292
Worms	0.516	0.406	0.516	0.344	0.375	0.359	0.359	0.422
WormsTwoClass	0.531	0.500	0.531	0.531	0.453	0.5	0.5	0.5
Yoga	0.534	0.652	0.535	0.606	0.592	0.533	0.572	0.588
ACSF1	0.25	0.271	0.229	0.208	0.292	0.208	0.292	0.229
AllGestureWiimoteX	0.209	0.131	0.129	0.093	0.113	0.102	0.098	0.102
AllGestureWiimoteY	0.189	0.108	0.118	0.113	0.094	0.102	0.125	0.102
AllGestureWiimoteZ	0.15	0.129	0.108	0.119	0.115	0.102	0.109	0.115
BME	0.393	0.420	0.333	0.387	0.413	0.347	0.427	0.407
Chinatown	0.693	0.480	0.512	0.523	0.471	0.523	0.515	0.43
Crop	0.286	0.172	0.093	0.133	0.175	0.05	0.095	0.141
DodgerLoopDay	0.275	0.213	0.138	0.238	0.25	0.15	0.213	0.15
DodgerLoopGame	0.558	0.543	0.529	0.486	0.522	0.522	0.507	0.551
DodgerLoopWeekend	0.63	0.696	0.681	0.71	0.71	0.739	0.659	0.71
EOGHorizontalSignal	0.207	0.224	0.159	0.179	0.156	0.205	0.182	0.199
EOGVerticalSignal	0.156	0.188	0.185	0.168	0.122	0.165	0.162	0.142
EthanolLevel	0.25	0.242	0.254	0.258	0.254	0.266	0.26	0.254
FreezerRegularTrain	0.612	0.627	0.807	0.661	0.591	0.591	0.642	0.603
FreezerSmallTrain	0.643	0.654	0.78	0.617	0.641	0.58	0.627	0.605
Fungi	0.075	0.075	0.059	0.048	0.086	0.07	0.086	0.113
GestureMidAirD1	0.07	0.125	0.102	0.109	0.141	0.039	0.18	0.156
GestureMidAirD2	0.086	0.109	0.102	0.094	0.078	0.039	0.102	0.078
GestureMidAirD3	0.047	0.102	0.094	0.039	0.055	0.039	0.078	0.063
GesturePebbleZ1	0.275	0.356	0.381	0.263	0.213	0.175	0.306	0.263
GesturePebbleZ2	0.243	0.292	0.278	0.201	0.215	0.167	0.125	0.194
GunPointAgeSpan	0.526	0.523	0.563	0.513	0.526	0.48	0.556	0.553
GunPointMaleVersusFemale	0.727	0.757	0.507	0.678	0.691	0.53	0.719	0.78

Dataset	TS-DC	Supervised	TS2Vec	TS-TCC	TST	TNC	CPC	T-Loss
GunPointOldVersusYoung	0.98	0.536	0.497	0.493	0.503	0.487	0.535	0.553
HouseTwenty	0.746	0.669	0.415	0.627	0.686	0.5	0.644	0.627
InsectEPGRegularTrain	0.835	0.556	0.48	0.48	0.46	0.508	0.456	0.488
InsectEPGSmallTrain	0.794	0.460	0.363	0.512	0.492	0.464	0.464	0.504
MelbournePedestrian	0.404	0.263	0.109	0.157	0.182	0.101	0.098	0.101
MixedShapesRegularTrain	0.382	0.282	0.232	0.303	0.275	0.213	0.318	0.326
MixedShapesSmallTrain	0.226	0.288	0.209	0.27	0.254	0.213	0.29	0.307
PickupGestureWiimoteZ	0.16	0.080	0.1	0.12	0.08	0.1	0.16	0.16
PigAirwayPressure	0.019	0.053	0.014	0.019	0.053	0.038	0.047	0.043
PigArtPressure	0.038	0.038	0.038	0.029	0.034	0.063	0.057	0.058
PigCVP	0.038	0.038	0.038	0.034	0.053	0.034	0.042	0.038
PLAID	0.163	0.157	0.17	0.182	0.102	0.063	0.17	0.167
PowerCons	0.898	0.625	0.557	0.597	0.619	0.625	0.731	0.568
Rock	0.44	0.360	0.36	0.34	0.32	0.34	0.34	0.28
SemgHandGenderCh2	0.667	0.630	0.659	0.652	0.669	0.65	0.674	0.65
SemgHandMovementCh2	0.292	0.199	0.167	0.188	0.185	0.185	0.19	0.179
SemgHandSubjectCh2	0.308	0.272	0.185	0.234	0.237	0.23	0.234	0.232
ShakeGestureWiimoteZ	0.22	0.180	0.1	0.14	0.2	0.1	0.14	0.14
SmoothSubspace	0.667	0.576	0.41	0.597	0.563	0.361	0.688	0.556
UMD	0.424	0.424	0.396	0.465	0.382	0.34	0.306	0.347
Average	0.454	0.449	0.382	0.416	0.424	0.358	0.424	0.416

F.2 Full Results of Multivariate Time Series Classification

Table 9 shows the full results of TS-DC and other baseline unsupervised representation learning methods as well as the supervised representation learning method on 30 UEA datasets. As depicted, TS-DC achieves better classification performance in terms of averaged accuracy.

Table 9: Full accuracy results of multivariate time series classification on 30 UEA datasets.

Dataset	TS-DC	Supervised	TS2Vec	TS-TCC	TST	TNC	CPC	T-Loss
ArticularyWordRecognition	0.337	0.441	0.201	0.316	0.330	0.111	0.382	0.299
AtrialFibrillation	0.357	0.500	0.429	0.357	0.429	0.500	0.286	0.286
BasicMotions	0.675	0.300	0.200	0.300	0.375	0.300	0.225	0.300
CharacterTrajectories	0.060	0.048	0.058	0.060	0.048	0.050	0.062	0.060
Cricket	0.438	0.453	0.453	0.328	0.469	0.359	0.297	0.438
DuckDuckGeese	0.200	0.220	0.140	0.180	0.260	0.160	0.240	0.120
EigenWorms	0.383	0.305	0.430	0.250	0.313	0.234	0.305	0.359
Epilepsy	0.320	0.328	0.313	0.281	0.367	0.383	0.281	0.344
EthanolConcentration	0.246	0.254	0.234	0.270	0.277	0.246	0.266	0.238
ERing	0.715	0.800	0.637	0.678	0.696	0.341	0.733	0.741
FaceDetection	0.602	0.577	0.521	0.517	0.578	0.520	0.502	0.585
FingerMovements	0.458	0.510	0.458	0.490	0.469	0.500	0.479	0.510
HandMovementDirection	0.297	0.391	0.234	0.250	0.219	0.266	0.250	0.328
Handwriting	0.072	0.100	0.068	0.084	0.085	0.078	0.097	0.097
Heartbeat	0.703	0.583	0.630	0.615	0.667	0.672	0.641	0.646
InsectWingbeat	0.100	0.101	0.101	0.099	0.103	0.099	0.100	0.100
JapaneseVowels	0.065	0.125	0.092	0.125	0.092	0.120	0.076	0.084
Libras	0.250	0.261	0.244	0.188	0.227	0.142	0.170	0.284
LSST	0.521	0.193	0.198	0.199	0.259	0.249	0.195	0.188
MotorImagery	0.615	0.563	0.458	0.427	0.490	0.490	0.458	0.604
NATOPS	0.773	0.659	0.773	0.642	0.733	0.676	0.778	0.767
PenDigits	0.978	0.981	0.946	0.963	0.903	0.647	0.828	0.889
PEMS-SF	0.363	0.563	0.300	0.263	0.263	0.231	0.331	0.338
PhonemeSpectra	0.042	0.040	0.032	0.028	0.038	0.031	0.026	0.026
RacketSports	0.458	0.333	0.375	0.333	0.361	0.285	0.292	0.292
SelfRegulationSCP1	0.566	0.788	0.510	0.757	0.740	0.431	0.681	0.639
SelfRegulationSCP2	0.506	0.466	0.506	0.500	0.489	0.574	0.472	0.506
SpokenArabicDigits	0.099	0.107	0.100	0.101	0.100	0.103	0.103	0.100
StandWalkJump	0.357	0.214	0.357	0.500	0.500	0.357	0.429	0.500
UWaveGestureLibrary	0.409	0.356	0.453	0.309	0.375	0.178	0.319	0.400
Average	0.399	0.385	0.348	0.347	0.375	0.311	0.343	0.369