Cross-Domain Contrastive Learning for Time Series Clustering

Furong Peng[1, 2], **Jiachen Luo**[1, 2], **Xuan Lu**[3*], **Sheng Wang**[4], **Feijiang Li**[1, 2]

[1] Institute of Big Data Science and Industry, Shanxi University
[2] School of Computer and Information Technology, Shanxi University
[3] College of Physics and Electronic Engineering, Shanxi University
[4] School of Automation, Zhengzhou University of Aeronautics
pengfr@sxu.edu.cn, luojike1418743@gmail.com, xuanlu@sxu.edu.cn,wangsheng1910@zua.edu.cn, fjli@sxu.edu.cn

Abstract

Most deep learning-based time series clustering models concentrate on data representation in a separate process from clustering. This leads to that clustering loss cannot guide feature extraction. Moreover, most methods solely analyze data from the temporal domain, disregarding the potential within the frequency domain.

To address these challenges, we introduce a novel end-to-end **C**ross-**D**omain **C**ontrastive learning model for time series **C**lustering (CDCC). Firstly, it integrates the clustering process and feature extraction using contrastive constraints at both cluster-level and instance-level. Secondly, the data is encoded simultaneously in both temporal and frequency domains, leveraging contrastive learning to enhance within-domain representation. Thirdly, cross-domain constraints are proposed to align the latent representations and category distribution across domains. With the above strategies, CDCC not only achieves end-to-end output but also effectively integrates frequency domains. Extensive experiments and visualization analysis are conducted on 40 time series datasets from UCR, demonstrating the superior performance of the proposed model.

Introduction

Data clustering, a technique for exploring data structure, has attracted significant attention (Li et al. 2018, 2019). Unlike image or text processing, the temporal variation of the series should be fully considered for similarity measurement, especially when data are distorted or shifted. Various methods have been proposed, such as Dynamic Time Wrapping (DTW) (Wang et al. 2018), Longest Common Subsequence (LCSS) (Górecki 2014), and Pearson correlation coefficient (Rodgers and Nicewander 1988). However, these methods have limitations in handling abnormalities, sensitivity, or complexity for long-term series clustering.

Along with similarity measurement, feature extraction is also crucial in time series clustering. For example, Zerveas *et al.* (Zerveas et al. 2021) utilized the transformer to extract features in an unsupervised manner, achieving superior results compared to some supervised methods. Tiano *et al.* (Tiano, Bonifati, and Ng 2021) proposed extracting statistical features for clustering. However, *the extracted features may not be beneficial for clustering tasks (Q1)* if the representation learning is separated from the clustering process. Ma *et al.* (Ma et al. 2021a) proposed an unsupervised model for clustering incomplete time series by integrating the K-means objective into an encoder-decoder network. The integration operation enhances the quality of clustering. Nevertheless, it is worth noting that these methods only analyze time series data from the temporal domain but *ignore utilizing the frequency domain information (Q2)*, which captures periodic patterns better and is more resilient against noise and outliers. (Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015).

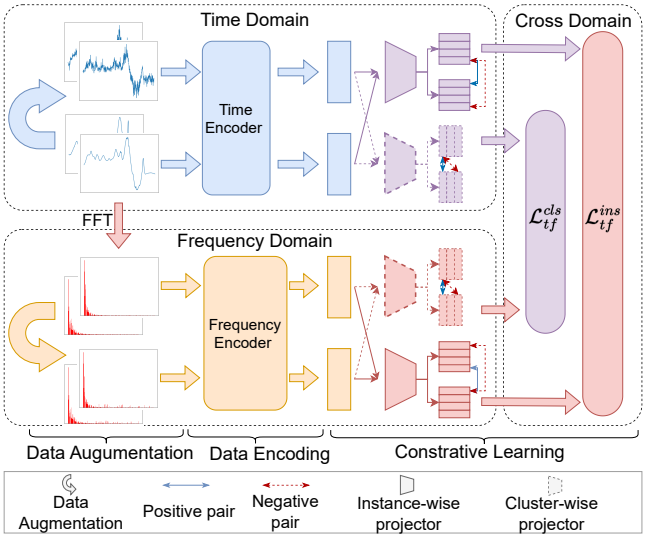

Figure 1: The framework of cross domain contrastive learning for time series clustering.

To address these issues, a **C**ross-**D**omain **C**ontrastive learning model for time series **C**lustering (CDCC) is proposed in this paper. Initially, the Fast Fourier Transform (FFT) (Brigham and Morrow 1967) is utilized to derive the frequency spectrum data, and augmentation techniques are applied to enhance both the temporal and spectrum data. Then, encoding networks in both domains are used for feature extraction. Instance-level and cluster-level con-

*Corresponding author

trastive constraints are leveraged to achieve end-to-end clustering in the temporal and frequency domain. By these intra-domain contrastive constraints, CDCC optimizes representations and category distributions for each domain separately. Furthermore, we employs a cross-domain contrastive constraint, including instance-level and cluster-level, to align the spectrum representation with the temporal domain representation, so that to capture waveform characteristics and periodicity for temporal domain representation. Finally, the clustering results are generated by the category assignments of the cluster-level contrast in the temporal domain, because most time series are labeled in this domain. In summary, the main contributions of this work are as follows:

- Proposing a cross-domain contrastive time series clustering framework that incorporates information in temporal and frequency domains by enabling comparison of representations within and between both domains.
- Adopting cluster-level constraints within and across domains to align cluster distributions and output sample categories from temporal domain.
- Conducting extensive experiments on 40 time series datasets, demonstrating that the proposed model achieves superior clustering performance.

Related Work

Deep Time Series Clustering

Deep clustering have demonstrated promising clustering quality via advanced data representation. These methods can be categorized into multi-step clustering and joint clustering, in view of the pipeline (Alqahtani et al. 2021).

Multi-step Clustering Multi-step clustering involves extracting time series representations or features, followed by traditional clustering algorithms such as K-means or hierarchical clustering. For instance, Chen *et al.* (Chen, Krishnan, and Sontag 2022) used RNN to learn encoded representations of time series, which were then clustered using K-means. Baytas *et al.* (Baytas et al. 2017) employed an improved LSTM to capture long-term dependencies in patient data for clustering purposes. CNNs have also been used to convert time series into visual images, enabling shape feature extraction and time series clustering (Han et al. 2019). However, these methods are often domain-specific and lack of universality. Moreover, the separation of feature extraction and clustering impedes effective guidance of feature extraction by clustering loss(Ma et al. 2021b).

Joint Clustering Joint clustering optimizes both feature extraction and clustering simultaneously to improve their compatibility. For example, Zhang *et al.* (Zhang and Sun 2023) learned representations and class labels using multivariate shapelets of various lengths under the assumption that time series in homogeneous clusters share similar subsequences. Ma *et al.* (Ma et al. 2021b) proposed a self-supervised time series clustering network that optimized feature extraction and clustering iteratively. Another approach by Ma *et al.* (Ma et al. 2019) aimed to minimize clustering errors using a discriminator to align the distribution of interpolated values with true values in the feature extraction.

In this paper, we propose a cross-domain contrastive clustering method that belongs to the joint clustering type. It achieves end-to-end joint clustering by instance-level and cluster-level contrastive constraints. The main distinction is that we learn the representation from both temporal and frequency domains, enhancing representation quality through cross-domain contrastive constraints. The entire process can be optimized using gradient back propagation to obtain a superior model.

Contrastive Learning

Contrastive learning, a self-supervised learning paradigm, has been popular in fields of natural language processing, computer vision, and recommendation systems (Zhang et al. 2021; Chen and He 2021; Xie et al. 2022). The core idea is to learn data representations or features by modeling the similarity and dissimilarity between samples.

In the context of time series, contrastive learning has been explored in many ways. Franceschi *et al.* (Franceschi, Dieuleveut, and Jaggi 2019) proposed an unsupervised representation learning model for multivariate time series using a time-based sampling strategy and triple loss to ensure that similar time series have similar representations. To consider neighborhood information, Tonekaboni *et al.* (Tonekaboni, Eytan, and Goldenberg 2021) assumed that signals from neighboring areas should have distinguishable distributions from non-neighborhood signals, and learned representations by a debiased contrastive objective. Eldele *et al.* (Eldele et al. 2021) introduced an unsupervised time series contrastive learning framework (TSTCC) based on temporal and context to capture contextual representations. Additionally, Yue *et al.* (Yue et al. 2022) developed a general framework for comparing time series across instances and time scales. However, these methods only focus on contrastive learning in temporal domain but ignore the frequency domain.

In this paper, we leverage the FFT of the time-frequency transform technique in signal processing to obtain frequency domain information, which enables cross-domain comparison of time series between temporal and frequency domains to incorporate crucial information into the representation.

Cross-Domain Contrastive Clustering

In this section, we introduce the proposed CDCC method that consists of three main components: data augmentation, encoding network, and contrastive constraints, incorporating both temporal and frequency domains to enhance clustering performances for time series data. The model framework is illustrated in Figure 1.

Data Augmentation

Temporal Domain Data Augmentation Given a set of time series dataset $\mathbf{X} = \{x_i\}_{i=1}^n$, x_i represents the i-th time series in the dataset. For each time series x_i, random mixing operations are applied on the library $\mathcal{T}$ of time-domain data augmentation methods, including jittering, scaling, and permutation operations, to generate the augmented data $\tilde{\mathbf{X}}^t = \{\tilde{x}_i^t\}_{i=1}^n$. To distinguish the domains, we denote the original

temporal domain data with superscript t (e.g. x_i^t), the corresponding frequency domain with superscript f (e.g. x_i^f), and the augmented data with ˜ (e.g. $\tilde{x}_i^t$, $\tilde{x}_i^f$). The temporal data augmentation process can be defined as follows:

$$\begin{gathered} T^a(x_i^t, \alpha),\ T^b(x_i^t, \beta),\ T^c(x_i^t, \gamma) \sim \mathcal{T}, \\ \tilde{x}_i^t = T^j(x_i^t),\ j \in \{a, b, c\}, \end{gathered} \tag{1}$$

where T^a, T^b, and T^c represent the data augmentation operations of jittering, scaling, and permutation respectively, α, β, γ correspond to the jittering rate, scaling rate, and permutation rate, respectively. We set $\alpha = 0.8$, $\beta = 1.1$, and $\gamma = 0.8$. Through extensive experiments, these parameter settings demonstrates favorable results.

Frequency Domain Data Augmentation Most existing data augmentation methods for time series focus on enhancing the data in temporal domain, while few approaches targets the frequency domain. In this study, we introduce a data augmentation technique on frequency domain inspired by the method proposed by Zhang *et al.* (Zhang et al. 2022). First, we employ FFT converting temporal data to frequency spectrum, $X^f = \{x_i^f\}_{i=1}^n$, where

$$x_i^f = FFT(x_i^t). \tag{2}$$

Subsequently, random mixing is applied on a library $\mathcal{F}$ of frequency domain data augmentation methods, including the addition and removal of frequency components. To add frequency components, we first calculate the maximum amplitude A_m in the spectrum. Then we randomly select θ frequency components with amplitudes smaller than ωA_m and increase their amplitudes to ωA_m, where ω and θ are predefined scaling factors and perturbation rates, respectively. To remove frequency components, a masking operation is used to randomly discard frequency components in the spectrum, with the masking rate ϵ.

It is important to note that excessive perturbation in frequency spectrum may lead to significant changes in temporal domain. Therefore it is crucial to avoid excessively large values for ω, θ, or ϵ. In our experiments, we set $\omega = 0.1$, $\theta = 0.1$, and $\epsilon = 0.1$. The frequency domain data augmentation process can be summarized as follows:

$$\begin{gathered} F^a(x_i^f, \omega, \theta), F^b(x_i^f, \epsilon) \sim \mathcal{F}, \\ \tilde{x}_i^f = F^j(x_i^f), \quad j \in \{a, b\}, \end{gathered} \tag{3}$$

where F^a and F^b refer to the operations of adding and removing frequency components, respectively. The frequency domain augmented data is denoted as $\tilde{X}^f = \{\tilde{x}_i^f\}_{i=1}^n$.

Encoding Network

The encoding network plays a pivotal role in our deep learning model, directly influencing its ability to capture the structure of time series data. To leverage the distinct characteristics of the temporal and frequency domains, we employ a bidirectional long short-term memory network (BiLSTM) (Kong et al. 2023) and a three-layer convolutional block as the encoders for the temporal and frequency domains, respectively. Choosing BiLSTM is because it considers both past and future information within the time series and extract abstract features across different time periods effectively. By feeding $X^t/\tilde{X}^t$ into BiLSTM, we obtain the corresponding temporal domain representations $H^t/\tilde{H}^t$ as follows:

$$H^t = BiLSTM(X^t), \quad \tilde{H}^t = BiLSTM(\tilde{X}^t). \tag{4}$$

We employ a three-layer convolutional block as the spectrum encoder. Specifically, each convolutional block consists of a convolutional layer ($Cv1d$), a batch normalization layer ($BN1d$), a rectified linear unit ($relu$) activation function, and a one-dimensional max pooling layer ($MaxPool1d$). Following the first convolutional block, a dropout layer is added after the max pooling layer to randomly deactivate pooled results, thus mitigating overfitting concerns. The three-layer convolutional block ($CB^3(\cdot)$) can be defined as follows:

$$\begin{gathered} CB^3(x_i^f) = CB(CB(Dropout(CB(x_i^f)))), \\ CB(x) = MaxPool1d(relu(BN1d(Cv1d(x)))), \end{gathered} \tag{5}$$

where $CB(x)$ is one layer convolutional block. By feeding $X^f/\tilde{X}^f$ into $CB^3(\cdot)$, we obtain the corresponding frequency domain representation $H^f/\tilde{H}^f$ as follows:

$$H^f = CB^3(X^f), \quad \tilde{H}^f = CB^3(\tilde{X}^f). \tag{6}$$

Contrastive Constraints

According to the optimization strategy of contrastive learning, contrastive constraints aim to maximize the similarity of representations between the original sample and the augmented data, while also enhancing the discriminability of different sample representations. We adopt the InfoNCE for contrastive loss function, as it effectively preserves the underlying data clusters (Parulekar et al. 2023). This type of contrast function, acting on individual samples, is referred to as the instance-level contrast constraint in this paper. Additionally, drawing inspiration from the contrastive clustering model (Li et al. 2021), in order to achieve end-to-end clustering results, we perform classification on sample representations to obtain pseudo-labels. These pseudo-labels are then used to construct cluster-level contrastive constraints. The cluster-level contrastive can facilitate the aggregation of similar samples, aiding the model in learning more discriminative representation in class-level. The instance-level and cluster-level contrastive losses will be discussed in the following.

The instance-level loss function does not directly act on the encoding results $\{H^t, \tilde{H}^t, H^f, \tilde{H}^f\}$ for model's robustness. Instead, it operates on the transformed instance-level representations $\{Z^t, \tilde{Z}^t, Z^f, \tilde{Z}^f\}$. Specifically, let z_i^I and $\tilde{z}_i^I$ represent the i-th row of the original view Z^I and its augmented view $\tilde{Z}^I$ in the temporal or frequency domain. In a training dataset of size n, each sample representation z_i^I is paired positively with its corresponding augmented view $\tilde{z}_i^I$ and negatively with other samples. The instance-level contrastive loss for sample z_i^I can be formulated as:

$$\begin{aligned}\mathcal{L}_{z_i^I} &= -log\frac{exp(s(z_i^I,\tilde{z}_i^I)/\tau^I)}{\sum_{j=1,j\neq i}^{n} exp(s(z_i^I,z_j^I)/\tau^I)+exp(s(z_i^I,\tilde{z}_j^I)/\tau^I)},\\ z_i^I &= g_\phi^I(h_i), \quad \tilde{z}_i^I = g_\phi^I(\tilde{h}_i),\\ g_\phi^I(h) &= normalize(MLP^2(h)), \quad h \in \{h_i, \tilde{h}_i\},\end{aligned} \tag{7}$$

where $MLP^2(\cdot)$ is a two-layer perception, $normalize(\cdot)$ is a normalization operation, τ^I denotes the instance-level temperature parameter. $s(z_i, \tilde{z}_i)$ is the cosine similarity between samples z_i and $\tilde{z}_i$, which is defined as:

$$s(z_i, \tilde{z}_i) = \frac{(z_i^T \tilde{z}_i)}{||z_i||||\tilde{z}_i||}. \tag{8}$$

Considering the symmetry between the original view z_i^I and the augmented view $\tilde{z}_i^I$, the instance-level contrastive loss can be expressed as:

$$\mathcal{L}^{ins}(Z^I, \tilde{Z}^I) = \frac{1}{2n}\sum_{i=1}^{n} \mathcal{L}_{z_i^I} + \mathcal{L}_{\tilde{z}_i^I}, \tag{9}$$

where $\mathcal{L}^{ins}(Z^I, \tilde{Z}^I)$ can be applied to the temporal and frequency domain.

The cluster-level constraint is also not directly applied to the output of the encoder, but rather on the results of the clustering. Let $Z^C_{\cdot,i}$ be the *i*-th column of category assignment $Z^C \in \mathbb{R}^{n\times k}$ (n is the number of samples, k is the number of clusters, Z^C can be Z^{C_t} or Z^{C_f}) . Similarly, $\tilde{Z}^C_{\cdot,i}$ is for the augmented data. Then, the cluster-level contrastive loss of $Z^C_{\cdot,i}$ can be written as:

$$\begin{aligned}\mathcal{L}_{Z^C_{\cdot,i}} &= -log\frac{exp(s(Z^C_{\cdot,i},\tilde{Z}^C_{\cdot,i})/\tau^C))}{\sum_{j=1,j\neq i}^{n} exp(s(Z^C_{\cdot,i},\tilde{Z}^C_{\cdot,j}))+exp(s(Z^C_{\cdot,i},Z^C_{\cdot,j}))},\\ z_i^C &= g_\phi^C(h_i), \quad \tilde{z}_i^C = g_\phi^C(\tilde{h}_i),\\ g_\phi^C(h) &= softmax(MLP^2(h)), h \in \{h_i, \tilde{h}_i\},\end{aligned} \tag{10}$$

where $g_\phi^C(h_i)$ function is calculated with a two-layer perception and a classification function $softmax(\cdot)$, τ^C is the cluster-level temperature parameter. In the clustering process, in order to prevent degenerate solutions, cross-entropy constraints are introduced:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{k} P_i^C log P_i^C - \tilde{P}_i^C log \tilde{P}_i^C, \tag{11}$$

where $P_i^C = \sum_{j=1}^{n} Z^C_{j,i}/n$, $\tilde{P}_i^C = \sum_{j=1}^{n} \tilde{Z}^C_{j,i}/n$. Through this constraint, the occurrence of empty clusters can be prevented. Considering the symmetry of the original view and the augmented view, the cluster-level contrastive loss can be expressed as:

$$\mathcal{L}^{cls}(Z^C, \tilde{Z}^C) = \frac{1}{2n}\sum_{i=1}^{k} \mathcal{L}_{Z^C_{\cdot,i}} + \mathcal{L}_{\tilde{Z}^C_{\cdot,i}} + \mathcal{L}_{ce}. \tag{12}$$

Cross-Domain Contrastive Constraints

To achieve information fusion between the temporal and frequency domain, firstly, instance-level and cluster-level contrastive constraints are separately applied to the two domains. Then, a third contrastive constraint between the two domains is established for information fusion. By employing Equation 9 and Equation 12 on the temporal domain representations $\{Z^{I_t}, \tilde{Z}^{I_t}, Z^{C_t}, \tilde{Z}^{C_t}\}$ and frequency domain representations $\{Z^{I_f}, \tilde{Z}^{I_f}, Z^{C_f}, \tilde{Z}^{C_f}\}$, we obtain the following two intra-domain loss functions:

$$\mathcal{L}_t = \mathcal{L}^{ins}(Z^{I_t}, \tilde{Z}^{I_t}) + \mathcal{L}^{cls}(Z^{C_t}, \tilde{Z}^{C_t}), \tag{13}$$

$$\mathcal{L}_f = \mathcal{L}^{ins}(Z^{I_f}, \tilde{Z}^{I_f}) + \mathcal{L}^{cls}(Z^{C_f}, \tilde{Z}^{C_f}). \tag{14}$$

Given that the frequency domain representation of any sample is derived from its temporal domain counterpart, it is expected that the representations of the same sample in different domains should exhibit analogous structural characteristics. To address this, we employ instance-level and cluster-level contrastive constraints for the augmented samples between domains. Specifically, the cross-domain contrastive loss function is:

$$\mathcal{L}_{tf} = \mathcal{L}^{ins}(\tilde{Z}^{I_t}, \tilde{Z}^{I_f}) + \mathcal{L}^{cls}(\tilde{Z}^{C_t}, \tilde{Z}^{C_f}). \tag{15}$$

It is essential to note that this cross-domain contrastive constraint is solely applied to the augmented samples, excluding the original data. Based on our experiments, it has been observed that conducting cross-domain constraint on the original data often results in model overfitting. Because the human labels are marked using the temporal domain, integrating too much frequency domain information into temporal domain will deteriorate the quality of clustering.

The final loss function is established by combining above three losses:

$$\mathcal{L} = \lambda(\mathcal{L}_t + \mathcal{L}_f) + (1-\lambda)\mathcal{L}_{tf}, \tag{16}$$

where λ is employed to balance the significance of the intra-domain constraint and cross-domain constraint. In our experimental setup, we set its value to 0.5. Finally, we adopt the Adam optimizer to optimize the proposed framework.

Clustering

The CDCC integrates representation learning and clustering process, where the clustering-level representation Z^c can serve as the basis for category assignments Y that can be determined as:

$$Y = \arg\max(g_\phi^C(BiLSTM(X))), \tag{17}$$

where $g_\phi^C(\cdot)$ represents the mapping function at the clustering level in the temporal domain. It is worth noting that time series data is typically labeled based on temporal information. Therefore, we employ the temporal category output as the final results. If frequency domain data is utilized for labeling, one may consider employing frequency domain.

Experiments

Experimental Setup

Dataset and Evaluation Metrics To validate the effectiveness of the proposed model, experiments were con-

Dataset	NMI							RI						
	TSTCC	TST	FeatTS	STCN	R-Clust	TCGAN	**CDCC**	TSTCC	TST	FeatTS	STCN	R-Clust	TCGAN	**CDCC**
ACSF1	0.477	0.491	0.364	0.314	0.545	0.331	**0.557**	0.820	0.742	0.680	0.760	0.867	0.562	**0.884**
Adiac	0.527	0.609	0.450	0.531	**0.708**	0.536	0.567	0.937	0.946	0.911	0.891	**0.956**	0.930	0.953
ArrowHead	0.270	0.324	0.326	0.324	**0.332**	0.288	0.310	0.656	0.677	0.695	0.692	0.661	0.621	**0.705**
Beef	0.295	0.283	0.277	0.173	0.270	0.290	**0.378**	0.679	0.671	0.612	0.695	0.670	0.639	**0.770**
Car	0.286	0.254	0.350	0.302	**0.562**	0.243	0.387	0.701	0.687	0.738	0.717	**0.792**	0.679	0.754
CBF	0.578	0.673	0.767	0.498	0.947	0.452	**0.993**	0.773	0.821	0.907	0.768	0.984	0.741	**0.998**
CricketX	0.360	0.276	0.272	0.133	0.323	0.291	**0.457**	0.871	0.864	0.736	0.833	0.864	0.863	**0.895**
CricketY	0.365	0.334	0.258	0.156	0.360	0.320	**0.439**	0.873	0.868	0.847	0.839	0.871	0.862	**0.889**
CricketZ	0.317	0.283	0.234	0.212	0.331	0.292	**0.388**	0.870	0.865	0.797	0.849	0.862	0.864	**0.885**
DSR	0.878	0.878	0.643	**0.896**	0.613	0.864	0.766	0.939	0.939	0.839	**0.941**	0.810	0.936	0.901
DPOAG	**0.472**	0.428	0.388	0.398	0.426	0.330	0.422	**0.752**	0.743	0.713	0.743	0.742	0.724	0.741
DPTW	0.594	0.552	0.565	**0.601**	0.550	0.499	0.577	**0.905**	0.901	0.804	0.888	0.808	0.786	0.883
ECG200	0.208	0.172	0.321	0.268	0.180	0.128	**0.381**	0.655	0.644	0.679	0.690	0.633	0.618	**0.696**
ECGFiveDays	0.358	0.006	0.586	0.233	0.018	0.002	**0.832**	0.727	0.504	0.844	0.652	0.511	0.501	**0.940**
EOGVS	0.252	**0.349**	0.175	0.221	0.132	0.219	0.319	0.848	0.859	0.784	0.846	0.833	0.785	**0.874**
FaceFour	**0.649**	0.448	0.376	0.505	0.646	0.434	0.557	**0.836**	0.756	0.707	0.810	0.828	0.725	0.828
FiftyWords	**0.689**	0.666	0.400	0.454	0.646	0.659	0.638	**0.957**	0.955	0.908	0.926	0.953	0.955	0.953
Fish	0.340	0.301	0.327	0.443	**0.555**	0.345	0.507	0.793	0.789	0.810	0.842	0.858	0.784	**0.863**
Fungi	**1.000**	0.983	0.730	0.861	**1.000**	0.926	0.969	**1.000**	0.995	0.918	0.960	**1.000**	0.972	0.992
GPMVF	0.341	0.421	0.477	0.384	0.000	0.341	**0.923**	0.617	0.696	0.786	0.737	0.499	0.617	**0.978**
GPOVY	0.565	**1.000**	0.705	0.979	**1.000**	0.343	**1.000**	0.782	**1.000**	0.897	0.995	**1.000**	0.619	**1.000**
HouseTwenty	0.403	0.202	**0.658**	0.115	0.246	0.065	0.568	0.751	0.633	**0.881**	0.575	0.661	0.540	0.838
Large.Kit.App	0.056	0.132	0.211	0.262	0.008	0.044	**0.271**	0.583	0.595	0.657	0.685	0.551	0.576	**0.693**
MPOAG	0.399	0.389	0.379	0.393	0.400	0.395	**0.403**	0.736	0.734	0.725	0.737	0.732	0.733	**0.740**
MPTW	0.430	0.427	**0.449**	0.445	0.412	0.405	0.432	**0.851**	0.849	0.791	0.841	0.795	0.785	0.833
OSULeaf	0.307	0.261	0.301	0.302	0.447	0.238	**0.524**	0.767	0.764	0.675	0.761	0.806	0.747	**0.841**
PAwP	0.597	0.586	0.170	0.408	**0.625**	0.581	0.613	0.968	0.964	0.217	0.908	0.967	0.965	**0.969**
PAP	0.696	0.677	0.723	0.589	**0.934**	0.687	0.833	0.971	0.971	0.955	0.926	**0.991**	0.968	0.983
PigCVP	0.596	0.534	0.306	0.532	0.714	0.572	**0.727**	0.960	0.952	0.693	0.931	0.972	0.959	**0.975**
Plane	0.932	0.932	0.617	0.946	0.981	0.830	**0.989**	0.954	0.954	0.847	0.959	0.994	0.917	**0.997**
PowerCons	0.683	0.727	0.447	0.351	0.465	0.568	**0.779**	0.889	0.894	0.766	0.717	0.725	0.837	**0.930**
PPTW	0.546	0.521	0.564	**0.631**	0.557	0.449	0.624	0.803	0.797	0.796	0.880	0.772	0.742	**0.883**
SHMC2	**0.266**	0.247	0.250	0.120	0.250	0.245	0.264	0.750	0.728	0.679	0.734	0.739	0.663	**0.770**
ShapeletSim	0.061	0.032	**1.000**	0.605	**1.000**	0.001	0.713	0.528	0.519	**1.000**	0.827	**1.000**	0.498	0.904
ShapesAll	0.724	0.714	0.205	0.508	0.751	0.698	**0.769**	0.979	0.978	0.615	0.908	0.981	0.972	**0.982**
SwedishLeaf	0.649	0.615	0.506	0.467	0.724	0.537	**0.770**	0.911	0.916	0.900	0.863	0.933	0.890	**0.953**
S.C.	**0.895**	0.784	0.626	0.671	0.810	0.739	0.884	0.929	0.880	0.857	0.856	0.900	0.862	**0.964**
TS1	0.004	0.002	0.202	**0.297**	0.020	0.000	0.261	0.500	0.499	0.629	**0.686**	0.512	0.498	0.668
Trace	0.558	0.800	0.591	0.804	**1.000**	0.500	0.750	0.762	0.843	0.759	0.898	**1.000**	0.749	0.874
WS	**0.506**	0.470	0.318	0.293	0.469	0.435	0.497	0.904	0.901	0.882	0.875	0.899	0.894	**0.905**
#Best↑	7	2	3	4	10	0	**18**	6	1	2	2	7	0	**26**
AVG NMI/RI↑	0.478	0.470	0.438	0.441	0.524	0.403	**0.601**	0.812	0.807	0.773	0.816	0.823	0.764	**0.877**
AVG RANK↓	3.425	4.238	4.800	4.688	3.213	5.563	**2.075**	3.213	3.925	5.288	4.363	3.588	5.825	**1.800**
P-Value	5E-04	7E-06	1E-07	1E-06	6E-03	6E-11		1E-03	3E-06	1E-10	4E-07	6E-05	2E-11	

Table 1: Overall performance comparison. #Best indicates the number best results on all datasets. AVG NMI/RI indicates average of NMI or RI over all datasets. AVG RANK indicates average rank. P-Value indicates the significance tests.

ducted on 40 time series datasets from UCR[1] (Dau et al. 2019). The statistical information of the datasets is presented in Appendix. The training and testing sets from the UCR were merged for evaluation. Normalized Mutual Information (NMI) and Rand Index (RI) are considered as metrics (Li, Qian, and Wang 2021; Aghabozorgi, Seyed Shirkhorshidi, and Ying Wah 2015).

[1]DSR: DiatomSizeReduction, DPOAG: DistalPhalanxOutlineAgeGroup, DPTW: DistalPhalanxTW, EOGVS: EOGVerticalSignal, GPMVF: GunPointMaleVersusFemale, GPOVY: GunPointOldVersusYoung, Large.Kit.App: LargeKitchenAppliances, MPOAG: MiddlePhalanxOutlineAgeGroup, MPTW: MiddlePhalanxTW, PAwP: PigAirwayPressure, PAP: PigArtPressure, PPTW: ProximalPhalanxTW, SHMC2: SemgHandMovementCh2, S.C.: SyntheticControl, TS1: ToeSegmentation1, WS: WordSynonyms

Baseline Methods Six models, including a semi-supervised models, a self-supervised model, and four unsupervised representation learning models (two-stage clustering) were chosen for performance evaluation[2]:

TSTCC (Eldele et al. 2021): A contrastive learning model that introduces strong augmentation and weak augmentation. Similar to TST, K-means clustering is applied for clustering tasks.

TST (Zerveas et al. 2021): An unsupervised representation learning model for time series based on transformers. It achieves better performance than supervised methods in regression, classification, and prediction. K-means is applied to the representations for clustering tasks.

FeatTS (Tiano, Bonifati, and Ng 2021): A novel semi-supervised clustering method that extracts discriminative

[2]https://github.com/JiacLuo/CDCC

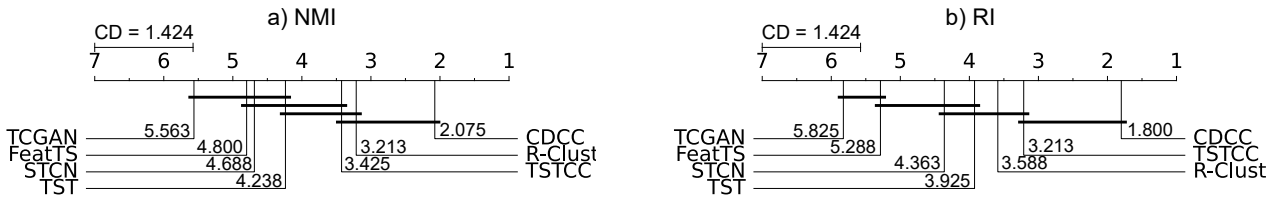

Figure 2: Critical-difference diagram for NMI and RI.

features from time series and then performs clustering.

STCN (Ma et al. 2021b): A self-supervised network for time series clustering, which optimizes feature extraction and clustering in a self-supervised manner.

R-Clust (Marco-Blanco and Cuevas 2023): A pipeline for time series clustering, using random convolutions and Principal Component Analysis (PCA) to extract features.

TCGAN (Huang and Deng 2023): A representation learning framework for time series, using adversarial game to optimize representations.

Following Yang *et al.* (Yang and Hong 2022), all the data are normalized using z-score normalization.

Parameter Settings We conducted a grid search on parameters which may affect the performance based on the recommendations in the corresponding paper and experimental analysis. In CDCC, $\tau^I = 0.5$, and $\tau^C = 1$. The learning rate, the number of layers in BiLSTM, batch size and the dropout rate was searched.

The experiments were conducted on a DCU Z100SM (16GB) computing card using PyTorch environment. The result of each algorithm is reported at their best parameters.

Overall Performance Comparing

The proposed CDCC method was compared with TSTCC, TST, FeatTS, STCN, R-Clustering, and TCGAN. As shown in Table 1, the CDCC achieved 18 best NMIs and 26 best RIs out of 40 datasets. It also achieved the highest average NMI of 0.601, the highest average RI of 0.877, and the highest average rankings of 2.075 (NMI) and 1.800 (RI), respectively. We also conduct pairwise comparisons between CDCC and other method using Wilcoxon signed-rank tests with Bonferroni correction (Demšar 2006). The results of the significance tests are presented in the last row (P-Value) of Table 1. At a significance level of $p < 0.05$, CDCC shows significant superiority over all the compared methods.

Furthermore, post-hoc Nemenyi tests (Demšar 2006) were conducted for accessing the statistical significance. The results, as depicted in Figure 2, reveal that CDCC exhibits a significantly superior performance compared to most baselines at a significance level of $p < 0.05$. The methods TCGAN, FeatTS, STCN, and TST, which are aligned along the horizontal line in the NMI diagram, display similar performance without statistically significant differences. Notably, it is worth mentioning that TSTCC displays better performance than others, owning to its strong and weak augmentation. R-Clustering outperforms others by using a large number of random convolution kernels to extract features.

Ablation Study

We conducted ablation experiments on the frequency domain contrast, temporal domain contrast, and cross-domain contrast modules to analyze their individual contributions. Experimental results are presented in Figure 3, where all ablation operations are listed as follows:

- $w/o\ \mathcal{L}_{tf}^{ins}$: without instance-level cross-domain contrast.
- $w/o\ \mathcal{L}_{tf}^{cls}$: without cluster-level cross-domain contrast.
- $w/o\ \mathcal{L}_f$: without frequency-domain contrast.
- $w/o\ \mathcal{L}_f \& \mathcal{L}_{tf}$: without both cross-domain contrast and frequency-domain contrast.

Figure 3 shows that removing instance-level contrast loss ($w/o\ \mathcal{L}_{tf}^{ins}$) or cluster-level contrast loss ($w/o\ \mathcal{L}_{tf}^{cls}$) from CDCC leads to a noticeable decrease. Likewise, excluding the frequency-domain loss ($w/o\ \mathcal{L}_f$) results in the optimization of data representation solely through the cross-domain contrast loss. Nevertheless, the model's clustering performance remains superior to that achieved when cluster-level of cross domain constraints are excluded. Moreover, when the cross-domain contrast loss is further disregarded ($w/o\ \mathcal{L}_f \& \mathcal{L}_{tf}$), the model's clustering metrics exhibit a significant decline. The above results indicate the effective guidance provided by the cross-domain contrast loss (especially for cluster-level) in learning representation of time series. More details can be found in the Appendix.

Parameter Analysis

CDCC's key parameters, the perturbation rate (θ), masking rate (ϵ), and balancing coefficient (λ), are analyzed to evaluate their impacts on performances using ArrowHead, CBF, Fungi, and SwedishLeaf datasets. θ and ϵ play similar roles in the model, we set them equally ($\theta = \epsilon$) for all datasets.

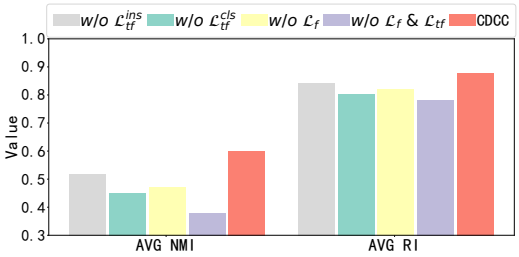

Figure 3: The ablation results. The average of NMI/RI on 40 datasets are used as metrics.

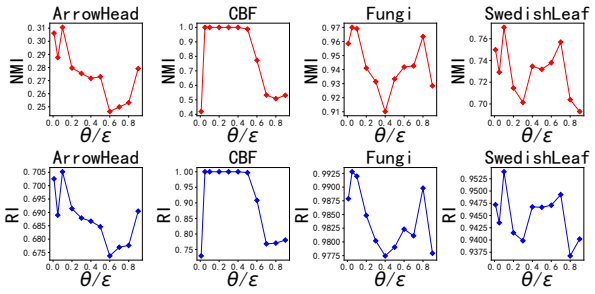

Figure 4: The impact of parameter θ/ϵ.

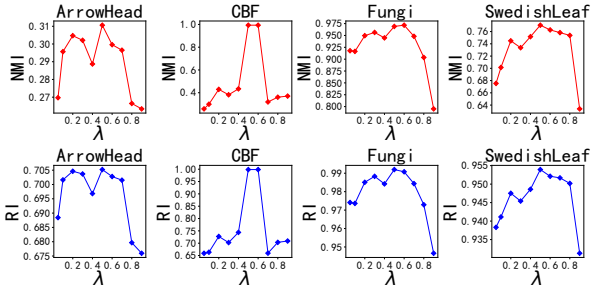

Figure 5: The impact of parameter λ.

Figure 4 depicts NMI and RI as a function of θ/ϵ. It can be observed that as θ/ϵ increases, the clustering metrics fluctuate continuously. However, smaller values of θ/ϵ generally result in better clustering performance. This is due to the sensitivity of frequency domain information to data augmentation methods, where excessive removal or addition of frequency components can disrupt useful features.

The impact of the balancing coefficient λ on the model's performance is illustrated in Figure 5. Different datasets exhibit varying sensitivities to λ. Generally, a trend can be observed where optimal clustering results are achieved when λ is around 0.5. This suggests that the cross-domain contrast constraint plays a crucial role in the overall model's constraints.

Visualization

Representation Visualization We visualize the distribution of representations on CBF and S.C. datasets by t-SNE (van der Maaten and Hinton 2008). The following observations can be made from Figure 6:

- The original data X are dispersed as depicted in the first column. T-SNE is unable to reveal the data structure as labeled by humans.

- The representations in the frequency domain (H^f) are also dispersed in the case of CBF , but exhibit some clustering tendencies in the case of S.C., as seen in the second column.

- The temporal domain (H^t) demonstrate clear clustering structures, as illustrated in the 3rd. The distribution of H^t demonstrates that CDCC can discover data structures in a manner similar to humans. Therefore, we primarily select temporal domain to generate cluster categories.

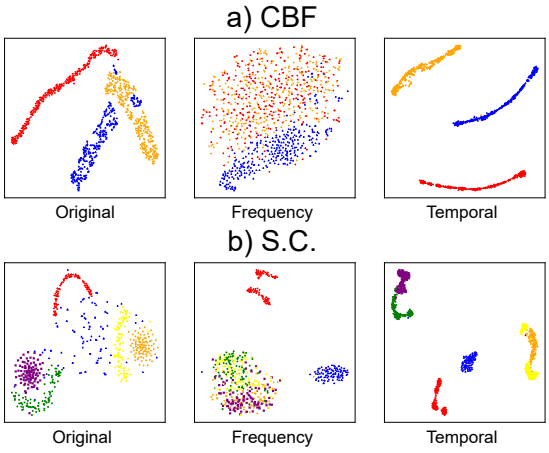

Figure 6: Visualization of the representations for different domains with t-SNE. The samples from the same class are marked in the same color.

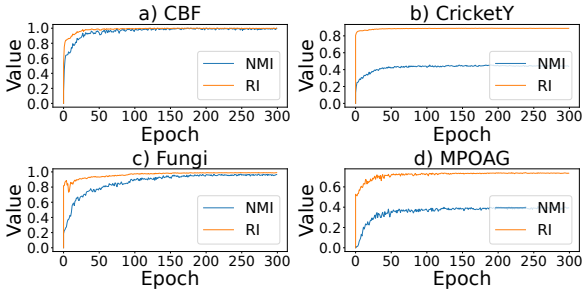

Figure 7: The convergence of clustering performance.

Convergence Analysis The clustering quality convergence of CDCC was analyzed on CBF, CricketY, Fungi, and MOAG datasets, as illustrated in Figure 7. It reveals that with an increasing number of epochs, the model's clustering performance, steadily improves until reaching convergence. These findings underscore the desirable convergence behavior exhibited by the proposed clustering model.

Conclusion

This work proposes a cross-domain contrastive learning model CDCC, for time series clustering. It utilizes intra-domain and cross-domain contrastive constraints to enhance the representation capability in both the temporal and frequency domains. By incorporating instance-level and cluster-level contrastive constraint, the model not only optimizes sample representations but also obtains clustering outputs. Extensive experiments demonstrate that the overall performance of the model is superior to existing models. Ablation experiments show that incorporating frequency domain information and cross-domain contrast can improve the clustering performance effectively. However, CDCC for aperiodic data stills need more explorations (e.g. image and device data), which is our future work.

Acknowledgments

This work was supported by the Science and Technology Innovation 2030-"New Generation of Artificial Intelligence" Major Program (No.2021ZD0112400), the National Natural Science Foundation of China (Nos. 62276162, 62106132, 62136005, 62272286), the Fundamental Research Program of Shanxi Province (No. 202203021222016), and the Science and Technology Major Project of Shanxi (No. 202201020101006).

References

Aghabozorgi, S.; Seyed Shirkhorshidi, A.; and Ying Wah, T. 2015. Time-series clustering – A decade review. *Information Systems*, 53: 16–38.

Alqahtani, A.; Ali, M.; Xie, X.; and Jones, M. W. 2021. Deep time-series clustering: A review. *Electronics*, 10(23): 3001.

Baytas, I. M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A. K.; and Zhou, J. 2017. Patient Subtyping via Time-Aware LSTM Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 65–74.

Brigham, E. O.; and Morrow, R. E. 1967. The fast Fourier transform. *IEEE Spectrum*, 4(12): 63–70.

Chen, I. Y.; Krishnan, R. G.; and Sontag, D. 2022. Clustering interval-censored time-series for disease phenotyping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6211–6221.

Chen, X.; and He, K. 2021. Exploring Simple Siamese Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15750–15758.

Dau, H. A.; Bagnall, A.; Kamgar, K.; Yeh, C.-C. M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C. A.; and Keogh, E. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6): 1293–1305.

Demšar, J. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7: 1–30.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C. K.; Li, X.; and Guan, C. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2352–2359.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Górecki, T. 2014. Using derivatives in a longest common subsequence dissimilarity measure for time series classification. *Pattern Recognition Letters*, 45: 99–105.

Han, L.; Zheng, K.; Zhao, L.; Wang, X.; and Shen, X. 2019. Short-Term Traffic Prediction Based on DeepCluster in Large-Scale Road Networks. *IEEE Transactions on Vehicular Technology*, 68(12): 12301–12313.

Huang, F.; and Deng, Y. 2023. TCGAN: Convolutional Generative Adversarial Network for time series classification and clustering. *Neural Networks*, 165: 868–883.

Kong, F.; Li, J.; Jiang, B.; Wang, H.; and Song, H. 2023. Integrated Generative Model for Industrial Anomaly Detection via Bidirectional LSTM and Attention Mechanism. *IEEE Transactions on Industrial Informatics*, 19(1): 541–550.

Li, F.; Qian, Y.; and Wang, J. 2021. GoT: a Growing Tree Model for Clustering Ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8349–8356.

Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Jing, L. 2019. Clustering ensemble based on sample's stability. *Artificial Intelligence*, 273: 37–55.

Li, F.; Qian, Y.; Wang, J.; Dang, C.; and Liu, B. 2018. Cluster's quality evaluation and selective clustering ensemble. *ACM Transactions on Knowledge Discovery From Data*, 12(5): 60.

Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J. T.; and Peng, X. 2021. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8547–8555.

Ma, Q.; Chen, C.; Li, S.; and Cottrell, G. W. 2021a. Learning Representations for Incomplete Time Series Clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10): 8837–8846.

Ma, Q.; Li, S.; Zhuang, W.; Li, S.; Wang, J.; and Zeng, D. 2021b. Self-Supervised Time Series Clustering With Model-Based Dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9): 3942–3955.

Ma, Q.; Zheng, J.; Li, S.; and Cottrell, G. W. 2019. Learning Representations for Time Series Clustering. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Marco-Blanco, J.; and Cuevas, R. 2023. Time Series Clustering With Random Convolutional Kernels. arXiv:2305.10457.

Parulekar, A.; Collins, L.; Shanmugam, K.; Mokhtari, A.; and Shakkottai, S. 2023. InfoNCE Loss Provably Learns Cluster-Preserving Representations. arXiv:2302.07920.

Rodgers, J. L.; and Nicewander, W. A. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42: 59–66.

Tiano, D.; Bonifati, A.; and Ng, R. 2021. FeatTS: Feature-Based Time Series Clustering. In *Proceedings of the 2021 International Conference on Management of Data*, 2784–2788. ISBN 9781450383431.

Tonekaboni, S.; Eytan, D.; and Goldenberg, A. 2021. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *9th International Conference on Learning Representations*.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Wang, W.; Lyu, G.; Shi, Y.; and Liang, X. 2018. Time Series Clustering Based on Dynamic Time Warping. In *2018 IEEE*

9th International Conference on Software Engineering and Service Science, 487–490.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive Learning for Sequential Recommendation. In *2022 IEEE 38th International Conference on Data Engineering*, 1259–1273.

Yang, L.; and Hong, S. 2022. Unsupervised Time-Series Representation Learning with Iterative Bilinear Temporal-Spectral Fusion. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, 25038–25054.

Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; and Xu, B. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8980–8987.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A Transformer-Based Framework for Multivariate Time Series Representation Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2114–2124.

Zhang, D.; Nan, F.; Wei, X.; Li, S.-W.; Zhu, H.; McKeown, K.; Nallapati, R.; Arnold, A. O.; and Xiang, B. 2021. Supporting Clustering with Contrastive Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5419–5430.

Zhang, N.; and Sun, S. 2023. Multiview Unsupervised Shapelet Learning for Multivariate Time Series Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4981–4996.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency. In *Advances in Neural Information Processing Systems*, volume 35, 3988–4003.