Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency

Anonymous Author(s)
Affiliation
Address
email

Abstract

Time series datasets pose a unique challenge for pre-training due to various kinds of discrepancy between pre-training and target domains, such as shifts in temporal dynamics, fast-evolving trends, long-range and short cyclic effects. While domain adaptation methods can mitigate these shifts, most methods need examples directly from the target domain, making them suboptimal for pre-training on time series. To fill this gap, methods need to accommodate a set of target domains with different temporal dynamics and be capable to do so without seeing any target examples during pre-training. Relative to other fields, in time series we expect that time-based and frequency-based representations of the same example and their local augmentations are located close together in the time-frequency space. To this end, we posit that time-frequency consistency (TF-C) — embedding a time-based neighborhood of a particular example close to its frequency-based neighborhood and back — is desirable for pre-training. Motivated by TF-C, we optimize a decomposable pre-training model, where the self-supervised signal is provided by the distance between time and frequency components, each individually trained by contrastive estimation. We evaluate the new method on eight datasets including electrodiagnostic testing, human daily activity recognition, mechanical fault detection, and physical status monitoring. Experiments against eight state-of-the-art methods show our method outperforms the baselines by 15.4% (F1 score) on average in one-to-one settings (*e.g.*, fine-tuning an EEG-pretrained model on EMG data) and by up to 8.4% (F1 score) in challenging one-to-many settings (*e.g.*, fine-tuning an EEG-pretrained model for either hand-gesture recognition or mechanical fault prediction), reflecting the breadth of scenarios that arise in real-world applications. The source code and all datasets are available at https://anonymous.4open.science/r/TFC-pretraining-6B07.

1 Introduction

Time series plays important roles in many areas, including clinical diagnosis, traffic analysis, and climate science [1–6]. While representation learning has considerably advanced analysis of time series [7, 8] and sequences more broadly [9], learning broadly generalizable representations for time series remains a fundamentally challenging problem [8, 10]. There are numerous immediate benefits from being able to generate such representations, of which pre-training capability is particularly desirable and of great practical importance [11, 12]. Central to pre-training is a question of how to process time series in a diverse dataset so as to greatly improve generalization on new time series coming from different datasets [13, 14, 9]. By training a neural network model on a diversified dataset and transferring it to a new target dataset for fine-tuning, *i.e.*, without explicit retraining on that target data, we expect the resulting performance be at least as good as that of state-of-the-art models tailored to the target dataset.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

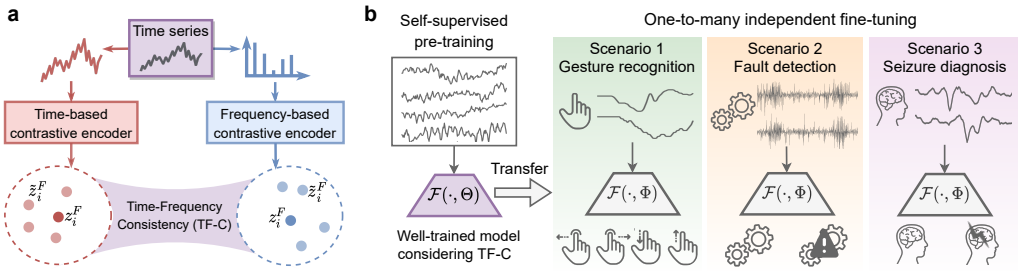

Figure 1: a. Illustration of Time-Frequency Consistency (TF-C). Time-based embedding $\boldsymbol{z}_i^{\mathrm{T}}$ and frequency-based embedding $\boldsymbol{z}_i^{\mathrm{F}}$ for a time series sample $\boldsymbol{x}_i^{\mathrm{T}}$, along with $\widetilde{\boldsymbol{z}}_i^{\mathrm{T}}$ and $\widetilde{\boldsymbol{z}}_i^{\mathrm{F}}$ learned from augmentations of $\boldsymbol{x}_i^{\mathrm{T}}$, should be close to each other in a latent time-frequency space. **b.** Leveraging TF-C property in time series to learn a pre-training model $\mathcal{F}$ with parameters Θ, which get fine-tuned to Φ using a small problem-specific dataset.

However, unfortunately, the expected performance gains are often not realized for a variety of reasons (*e.g.*, distribution shifts, properties of the target dataset unknown during pre-training) [15, 16] that get compounded by the complexity of time series: large variations of temporal dynamics across datasets, varying semantic meaning, irregular sampling, system factors (*e.g.*, different devices or subjects), etc. [17, 16]. This complexity of time series limits the utility of knowledge transfer for pre-training [18, 19]. For example, pre-training a model on a diverse time series dataset with mostly low-frequency components (smooth trends) may not lead to positive transfer on downstream tasks with high-frequency components (transient events) [16]. Examining these challenges can provide clues to what kind of inductive biases could facilitate generalizable representations of time series – this paper offers a strategy for that through a novel time-frequency consistency principle.

In addition, target dataset are not available in pre-training phase (different from domain adaption [20]; Appendix A), requiring that pre-training model captures some latent shared property in time series that can apply to unseen datasets. The principle of a shared property is universal in different time series datasets and thus allows knowledge transfer from a pre-training dataset to a fine-tune dataset. To make it more clear, a key principle for pre-training in computer vision (CV) is that the prior neural network layers learn object edges and shapes regardless of image style or tasks [21]. The shared foundation for pre-training in natural language processing (NLP) is the linguistic assumption of semantics and grammars that is retrained across different languages [22]. Nevertheless, it is hard to put forward such a principle for time series due to aforementioned temporal characteristics.

Moreover, supervised pre-training requires large annotated datasets, creating an obstacle to deep learning deployment [23, 24]. For example, in medical domains, experts often disagree on ground-truth labeling [25, 26], which can introduce unintended biases, *e.g.*, ECG signals falling outside of normal and abnormal rhythms [27, 28]. Self-supervised contrastive learning can overstep the lack of annotations by providing self-supervision signal through contrastive instance discrimination [29].

Present work. We introduce a strategy for self-supervised pre-training in time series by modeling Time-Frequency Consistency (TF-C). TF-C means the time-based representation and the frequency-based representation, which are learned from the same time series sample, are closer to each other in a joint time-frequency space, and farther apart if the representations are associated with different time series samples. Specifically, we adopt contrastive learning in time space to generate a time-based representation. In parallel, we propose a set of novel augmentations based on the characteristic of frequency spectrum and produce a frequency-based embedding through contrastive instance discrimination. To the best of our knowledge, this is the first work that implements augmentation in frequency domain to leverage rich information in frequency. The pre-training objective is to minimize the distance between the time-based and frequency-based embeddings through a dedicated TF-C loss (Figure 1 (a)). The TF-C loss provides self-supervising signals to optimize the pre-training model and captures the latent relationships between time domain and frequency domain. The learned relationship (*i.e.*, TF-C) encoded in model parameters are transferred to initialize the fine-tuning model and improve performance in datasets of interest (Figure 1 (b)).

We evaluate the proposed pre-training model on eight time series datasets under two evaluation settings (one-to-one and one-to-many). The eight datasets cover a large set of variations: different number of channels (from univariate to 9-channel multivariate), varying time series length (from 128 to 5,120), different sampling rate (from 16 Hz to 4,000 Hz), different scenarios (neurological healthcare, human activity recognition, mechanical fault detection, physical status monitoring, etc.)

and diverse types of signals (EEG, EMG, ECG, acceleration, and vibration). We compare TF-C approach to eight state-of-the-art baselines. Results show that TF-C achieves positive transfer, outperforming baselines by a large margin of 15.4% (F1 score) on average. Further, the approach outperforms the strongest baselines with an improvement of up to 7.2% in F1 score. Finally, the TF-C approach improves prior work by 8.4% in precision (when pre-training the model on sleep EEG signals and fine-tuning it on hand-gesture recognition) in a challenging one-to-many setup that applies the same pre-trained model to multiple independent fine-tuning datasets.

2 Related Work

Pre-training for time series. Although there are some studies about self-supervised representation learning for time series [7, 8, 30, 31] and many works on self-supervised pre-training on image data [32–34, 23], their intersection, self-supervised pre-training on time series data, still remains insufficiently explored. In time series, it's not obvious that what assumptions are reasonable that can bridge pre-training and target datasets. Hence, pre-training models in CV [35, 36, 13] and NLP [9, 14, 37] are not directly applicable due to data modality mismatch, and results leave room for improvement [30, 38, 39]. So far, only Shi et al. proposed a model for self-supervised time series pre-training explicitly [11]. The model captures the local and global temporal pattern but it's not convincing why the designed pretext task can capture generalizable representations. Although several studies applies to transfer setting in time series [7, 8, 17, 40], there is no devoted conceptual assumptions or dedicated architectural modules to justify why their model empowers knowledge transfer across datasets. Addressing this gap, we posit the assumption of TF-C which is invariant to different time series datasets and can guide the development of effective pre-training models.

Different from domain adaptation models [20, 41] that implies target datasets are required to in model training, our task only have access to the pre-training dataset which demands more generalizeable property in time series. Self-supervised domain adaptation means no labels in target dataset but still require labels in model training [42, 43] while our model does not rely on labels in model pre-training.

Contrastive learning with time series. Contrastive learning, a popular type of self-supervised learning, aims to learn an encoder that maps inputs into an embedding space such that positive sample pairs (original augmentation and another alternative augmentation/view of the same input sample) are pulled closer and negative sample pairs (original augmentation and an alternative input sample augmentation) are pushed apart [29, 44]. Contrastive learning in time series is less investigated in comparison, partly due to the challenge of identifying augmentations that capture key invariance properties in time series data. For example, CLOCS defines adjacent segments of a time series as positive pairs [40], and TNC assumes overlapping neighborhoods of time series should have similar representations [45]. These methods leverage temporal invariance to define positive pairs which are used to calculate contrastive loss, but other forms of invariance- based, such as transformation (*e.g.*, SimCLR [39]) or contextual invariance (*e.g.*, TS2vec [46], TS-TCC [47]), augmentations are also possible. In this work, we propose an augmentation bank that exploits multiple invariances to generate diverse augmentations (Sec. 4.1), which adds richness into the pre-training model [47]. Importantly, we propose frequency-based augmentations by perturbing the frequency spectrum of time series (*e.g.*, adding or removing the frequency components and manipulating the their amplitude; more details in Sec. 4.2) to learn better representations by exposing the model to local range of frequency variations. A previous work CoST processes sequential signals through frequency domain but the augmentations are still implemented in time space [48]. To the best of our knowledge, this is the first work that develops augmentations in frequency domain to leverage frequency-invariance for contrastive learning. Further, we propose a pre-training model that subjects to TF-C upon two individual contrastive encoders.

3 Problem Formulation

We are given a pre-training dataset $\mathcal{D}^{\text{pret}} = \{\boldsymbol{x}_i^{\text{pret}} \mid i = 1, \ldots, N\}$ of unlabeled time series samples where sample $\boldsymbol{x}_i^{\text{pret}}$ has K^{pret} channels and L^{pret} timestamps. Let $\mathcal{D}^{\text{tune}} = \{(\boldsymbol{x}_i^{\text{tune}}, y_i) \mid i = 1, \ldots, M\}$ be a fine-tuning (*i.e.*, target; target and fine-tuning are used interchangeably) dataset of labeled time series samples, each having K^{tune} channels and L^{tune} timestamps. Further, every sample $\boldsymbol{x}_i^{\text{tune}}$ is associated with label $y_i \in \{1, \ldots, C\}$, where C is the number of classes. Without loss of generality, in the following, we describe the approach for univariate (single-channel) time series, noting that the

approach can accommodate multivariate time series, different numbers of channels across samples, and varying lengths (shown in experiments in Sec. 5.2). We use superscript symbol $\tilde{}$ to denote contrastive augmentations. We note that $\boldsymbol{x}_i^{\text{T}} \equiv \boldsymbol{x}_i$ denotes an input time series sample, and $\boldsymbol{x}_i^{\text{F}}$ denotes the frequency spectrum of $\boldsymbol{x}_i$.

Problem (Self-Supervised Contrastive Pre-Training For Time Series). *Given are an unlabeled pre-training dataset $\mathcal{D}^{pret}$ with N samples and a target dataset $\mathcal{D}^{tune}$ with M samples ($M \ll N$). We want to use $\mathcal{D}^{pret}$ to pre-train a neural model $\mathcal{F}$ so that by fine-tuning model parameters on $\mathcal{D}^{tune}$, the fine-tuned model can generate a generalizable representation $\boldsymbol{z}_i^{tune} = \mathcal{F}(\boldsymbol{x}_i^{tune})$ for every $\boldsymbol{x}_i^{tune}$.*

We follow established pre-training setup, *e.g.*, [40]; for pre-training, only unlabeled dataset $\mathcal{D}^{\text{pret}}$ is available while, for fine-tuning, a small labeled dataset $\mathcal{D}^{\text{tune}}$ can be used. In short, a model $\mathcal{F}$ is pre-trained on unlabeled time series dataset $\mathcal{D}^{\text{pret}}$ and its optimized model parameters Θ are fine-tuned to go from $\mathcal{F}(\cdot, \Theta)$ to $\mathcal{F}(\cdot, \Phi)$ using dataset $\mathcal{D}^{\text{tune}}$. The Φ denotes fine-tuned model parameters. Note that this problem (*i.e.*, $\mathcal{D}^{\text{pret}}$ is independent of target dataset) is different than domain adaptation because we don't access the fine-tuning dataset $\mathcal{D}^{\text{tune}}$ during pre-training, meaning that a successful pre-trained model should be applicable to multiple independent fine-tuning datasets without necessitating re-training.

Rationale for Time-Frequency Consistency (TF-C). For effective per-training on time series, our central idea is to identify a broadly generalizable property that is preserved across different time series datasets. In time series in particular, the time domain shows how readouts change with time whereas frequency domain tells us how much of the signal lies within each frequency band over a range of frequencies (*e.g.*, frequency spectrum) [49]. Explicitly considering the frequency domain can provide a deeper understanding of time series behavior that can be challenging to capture solely in time domain [50]. However, most existing time-series methods focus exclusively on modeling time domain, *e.g.*, [46, 47], and ignore frequency domain altogether. One could argue that is sufficient for high-capacity methods as time and frequency domains describe the same data, just from different aspects [51], and time series can be converted between time and frequency domains using transformation, such as Fourier [52, 50]. While we are not interested in modeling the two domains individually, the relationship between them, grounded in signal processing theory, provides an invariance that is valid regardless of time series distribution [53, 54] and thus can serve as an inductive bias for pre-training to promote generalizability. Approaching this invariance through the lens of representation learning, we formulate Time-Frequency Consistency (TF-C) by postulating that for every time series sample $\boldsymbol{x}_i$, there exists a latent time-frequency space in which time-based representation $\boldsymbol{z}_i^{\text{T}}$ and frequency-based representation $\boldsymbol{z}_i^{\text{F}}$ of the same sample, together with their local augmentations (defined later) are close to each other.

Representational Time-Frequency Consistency (TF-C). *Let be given a time series sample $\boldsymbol{x}_i$. Then in a model $\mathcal{F}$ satisfying TF-C, time-based representation $\boldsymbol{z}_i^{\text{T}}$ and frequency-based representation $\boldsymbol{z}_i^{\text{F}}$ learned from $\boldsymbol{x}_i$ by $\mathcal{F}$, and representations learned from local augmentations of $\boldsymbol{x}_i$ are close together in the latent time-frequency space.*

Our pre-training strategy is to use a time series dataset $\mathcal{D}^{\text{pret}}$ to induce TF-C in $\mathcal{F}$'s model parameters Θ, which, in turn, are used to initialize the target model on $\mathcal{D}^{\text{pret}}$ and produce generalizable representations for downstream prediction. The invariant nature of TF-C means that our approach can bridge $\mathcal{D}^{\text{pret}}$ and $\mathcal{D}^{\text{pret}}$ even when large discrepancies exist between them (in terms of temporal dynamics, semantic meaning, etc.), providing a vehicle for a generalizable pre-training strategy for time series.

To realize TF-C, our model $\mathcal{F}$ has four components (Figure 2): a time encoder G_{T}, a frequency encoder G_{F}, and two cross-space projectors R_{T} and R_{F} that map time-based and frequency-based representations, respectively, to the same time-frequency space. Together, the four components provide a way to embed $\boldsymbol{x}_i$ to the latent time-frequency space such that time-based embedding $\boldsymbol{z}_i^{\text{T}} = R_{\text{T}}(G_{\text{T}}(\boldsymbol{x}_i^{\text{T}}))$ and frequency-based embedding $\boldsymbol{z}_i^{\text{F}} = R_{\text{F}}(G_{\text{F}}(\boldsymbol{x}_i^{\text{F}}))$ are close together.

4 Our Approach

Next, we present the architecture of the proposed self-supervised contrastive pre-training model $\mathcal{F}$. Without specification, the data mentioned in this section are from pre-training dataset and the superscript ^{pret} is omitted for simplification. Here we describe the model using univariate time series as an example, but our model can be straightforwardly applied to multivariate time series (Sec 5).

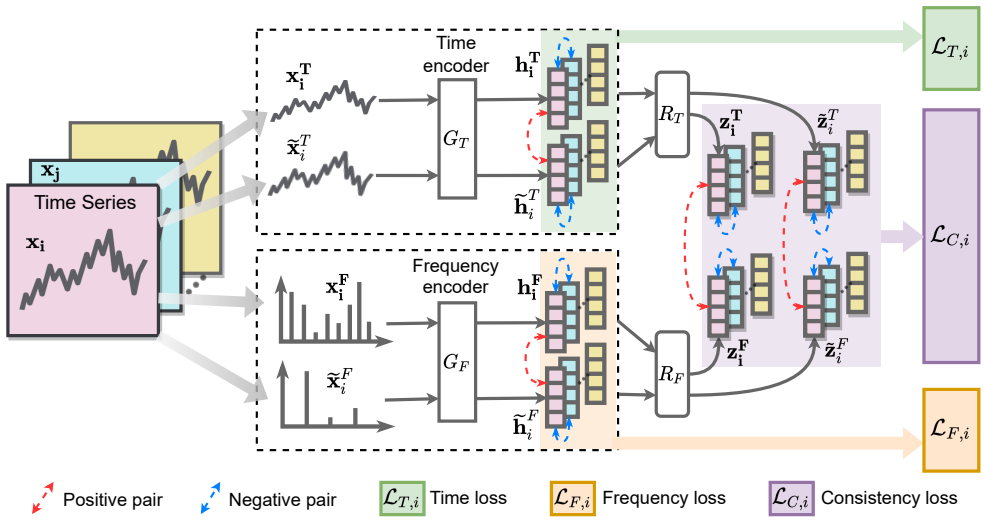

Figure 2: Overview of TF-C approach. Our TF-C pre-training model $\mathcal{F}$ has four components: a time encoder G_{T}, a frequency encoder G_{F}, and two cross-space projectors R_{T} and R_{F}. For input time series $\boldsymbol{x}_i$, the model produces time-based representations (*i.e.*, $\boldsymbol{z}_i^{\text{T}}$ and $\widetilde{\boldsymbol{z}}_i^{\text{T}}$ of input $\boldsymbol{x}_i$ and its augmented version, respectively) and frequency-based representations (*i.e.*, $\boldsymbol{z}_i^{\text{F}}$ and $\widetilde{\boldsymbol{z}}_i^{\text{F}}$ of input $\boldsymbol{x}_i$ and its augmented version, respectively). The TF-C property is realized by promoting the alignment of time- and frequency-based representations in the latent time-frequency space, providing a vehicle for transferring $\mathcal{F}$ to a target dataset not seen before.

4.1 Time-based contrastive encoder

For input time series sample $\boldsymbol{x}_i$, we generate an augmentation set $\mathcal{X}_i^{\text{T}}$ through an augmentation bank $\mathcal{B}^{\text{T}} : \boldsymbol{x}_i^{\text{T}} \rightarrow \mathcal{X}_i^{\text{T}}$. Each element $\widetilde{\boldsymbol{x}}_i^{\text{T}} \in \mathcal{X}_i^{\text{T}}$ is a sample that is augmented from $\boldsymbol{x}_i$ based on the temporal characteristics. Augmentation bank includes jittering, scaling, time-shifts, and neighborhood segments, all established in contrastive learning [39, 47, 40]. We develop the augmentation bank to produce diverse augmentations (rather than a single types of augmentations) and expose the model to complex temporal dynamics, producing robust time-based embeddings [47].

We randomly select an element $\widetilde{\boldsymbol{x}}_i^{\text{T}}$ as an augmented time series and use the time encoder G_{T} to map samples to embeddings $\boldsymbol{h}_i^{\text{T}} = G_{\text{T}}(\boldsymbol{x}_i^{\text{T}})$. As $\widetilde{\boldsymbol{x}}_i^{\text{T}}$ is generated based on $\boldsymbol{x}_i^{\text{T}}$, after passing through G_{T}, we assume the embedding of $\boldsymbol{x}_i^{\text{T}}$ is close to the embedding of $\widetilde{\boldsymbol{x}}_i^{\text{T}}$ but far away from the embedding of $\boldsymbol{x}_j^{\text{T}}$ and $\widetilde{\boldsymbol{x}}_j^{\text{T}}$ that are derived from another sample $\boldsymbol{x}_j^{\text{T}} \in \mathcal{D}^{\text{pret}}$. In specific, we select the positive pair as $(\boldsymbol{x}_i^{\text{T}}, \widetilde{\boldsymbol{x}}_i^{\text{T}})$ and negative pairs as $(\boldsymbol{x}_i^{\text{T}}, \boldsymbol{x}_j^{\text{T}})$ and $(\boldsymbol{x}_i^{\text{T}}, \widetilde{\boldsymbol{x}}_j^{\text{T}})$.

Contrastive time loss. To maximize the similarity within a positive pair and minimize the similarity within a negative pair, we adopt the NT-Xent (the normalized temperature-scaled cross entropy loss) as distance function d which is widely used in contrastive learning [33, 39]. In specific, we measure the loss of time-based contrastive encoder in terms of sample $\boldsymbol{x}_i^{\text{T}}$ by:

$$\mathcal{L}_{\text{T},i} = d(\boldsymbol{h}_i^{\text{T}}, \widetilde{\boldsymbol{h}}_i^{\text{T}}, \mathcal{D}^{\text{pret}}) = -\log \frac{\exp(\text{sim}(\boldsymbol{h}_i^{\text{T}}, \widetilde{\boldsymbol{h}}_i^{\text{T}})/\tau)}{\sum_{\boldsymbol{x}_j \in \mathcal{D}^{\text{pret}}} \mathbb{1}_{i \neq j} \exp(\text{sim}(\boldsymbol{h}_i^{\text{T}}, G_{\text{T}}(\boldsymbol{x}_j))/\tau)}, \tag{1}$$

where $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$ denotes the cosine similarity, the $\mathbb{1}_{i \neq j} \in \{0, 1\}$ is a binary indicator that equals to 0 when $i = j$, and τ is a temporal parameter. Here, the $\boldsymbol{x}_j \in \mathcal{D}^{\text{pret}}$ refers to both the original sample $\boldsymbol{x}_j^{\text{T}} \in \mathcal{D}^{\text{pret}}$ and the corresponding augmented sample $\widetilde{\boldsymbol{x}}_j^{\text{T}} \in \mathcal{X}_j^{\text{T}}$. This loss urges the time encoder G_{T} to generate closer embeddings for positive pairs, and push the embeddings for negative pairs apart from each other.

4.2 Frequency-based contrastive encoder

Next, we generate the frequency spectrum $\boldsymbol{x}_i^{\text{F}}$ from time series $\boldsymbol{x}_i^{\text{T}}$ through a transform operator, *e.g.*, [52, 55]. We consider augmentation methods to perturb $\boldsymbol{x}_i^{\text{F}}$ based on characteristics of frequency spectra. As every frequency component in the frequency spectrum denotes a base function (*e.g.*, sinusoidal function for Fourier transformation) with the corresponding frequency and amplitude. In our augmentation methods, we perturb the frequency spectrum by adding or removing frequency

components. The small change of frequency spectrum will largely affect the temporal patterns in time domain [53]. To make sure the perturbed time series is still similar to the original sample, we use a small budget Δ in the perturbations where Δ denotes the number of frequency components we perturb. While removing frequency components, we randomly select Δ frequency components and set their amplitudes as 0. While adding frequency components, we randomly choose Δ frequency components from the ones have smaller amplitude than $\alpha \cdot A_m$, and increase their amplitude to $\alpha \cdot A_m$. The A_m is the maximum amplitude in the frequency spectrum and α is a pre-defined coefficient to adjust the sale of the perturbed frequency component ($\alpha = 0.5$ in this work). We produce an augmentation set $\mathcal{X}_i^{\mathrm{F}}$ for $\boldsymbol{x}_i^{\mathrm{F}}$ through frequency-augmentation bank $\mathcal{B}^{\mathrm{F}} : \boldsymbol{x}_i^{\mathrm{F}} \rightarrow \mathcal{X}_i^{\mathrm{F}}$. We have two augmentation methods (*i.e.*, removing/adding frequency components) in $\mathcal{B}^{\mathrm{F}}$, $|\mathcal{X}_i^{\mathrm{F}}| = 2$.

We utilize a frequency encoder G_{F} to map the frequency spectrum (*e.g.*, $\boldsymbol{x}_i^{\mathrm{F}}$) to a frequency-based embedding (*e.g.*, $\boldsymbol{h}_i^{\mathrm{F}} = G_{\mathrm{F}}(\boldsymbol{x}_i^{\mathrm{F}})$). We assume the frequency encoder G_{F} can learn similar embedding for the original frequency spectrum $\boldsymbol{x}_i^{\mathrm{F}}$ and a slightly perturbed frequency spectrum $\widetilde{\boldsymbol{x}}_i^{\mathrm{F}} \in \mathcal{X}_i^{\mathrm{F}}$. Thus, we set the positive pair as $(\boldsymbol{x}_i^{\mathrm{F}}, \widetilde{\boldsymbol{x}}_i^{\mathrm{F}})$ and the negative pairs as $(\boldsymbol{x}_i^{\mathrm{F}}, \boldsymbol{x}_j^{\mathrm{F}})$ and $(\boldsymbol{x}_i^{\mathrm{F}}, \widetilde{\boldsymbol{x}}_j^{\mathrm{F}})$.

Contrastive frequency loss. We calculate frequency-based contrastive loss for sample $\boldsymbol{x}_i$ as:

$$\mathcal{L}_{\mathrm{F},i} = d(\boldsymbol{h}_i^{\mathrm{F}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{F}}, \mathcal{D}^{\text{pret}}) = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{h}_i^{\mathrm{F}}, \widetilde{\boldsymbol{h}}_i^{\mathrm{F}})/\tau)}{\sum_{\boldsymbol{x}_j \in \mathcal{D}^{\text{pret}}} \mathbb{1}_{i \neq j} \exp(\mathrm{sim}(\boldsymbol{h}_i^{\mathrm{F}}, G_{\mathrm{F}}(\boldsymbol{x}_j))/\tau)}. \tag{2}$$

In preliminary work, we find that the value of τ has little effect on performance and use the same τ throughout all experiments. The $\mathcal{L}_{\mathrm{F},i}$ yield a frequency encoder G_{F} producing embeddings invariant to frequency spectrum perturbations.

4.3 Time-frequency consistency

Our consistency loss $\mathcal{L}_{\mathrm{C}}$ urges learned embeddings to satisfy TF-C: for a given sample, its time-based and frequency-based embeddings and their local neighborhoods should be close to each other. We first map the $\boldsymbol{h}_i^{\mathrm{T}}$ from latent time domain and the $\boldsymbol{h}_i^{\mathrm{F}}$ from latent frequency domain to a time-frequency domain through projector R_{T} and R_{F}, respectively. Particularly, for every input sample $\boldsymbol{x}_i$, we have four embeddings, which are $\boldsymbol{z}_i^{\mathrm{T}} = R_{\mathrm{T}}(\boldsymbol{h}_i^{\mathrm{T}})$, $\widetilde{\boldsymbol{z}}_i^{\mathrm{T}} = R_{\mathrm{T}}(\widetilde{\boldsymbol{h}}_i^{\mathrm{T}})$, $\boldsymbol{z}_i^{\mathrm{F}} = R_{\mathrm{F}}(\boldsymbol{h}_i^{\mathrm{F}})$, and $\widetilde{\boldsymbol{z}}_i^{\mathrm{F}} = R_{\mathrm{F}}(\widetilde{\boldsymbol{h}}_i^{\mathrm{F}})$.

To enforce the embeddings in time-frequency domain subject to TF-C, we design a novel TF-C loss that measures the distance between a time-frequency embedding pair. The time-frequency embedding pair refers to the pair of a time-based embedding and a frequency-based embedding. Thus, for a sample $\boldsymbol{x}_i$, we have four time-frequency embedding pairs, which are $(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}})$, $(\boldsymbol{z}_i^{\mathrm{T}}, \widetilde{\boldsymbol{z}}_i^{\mathrm{F}})$, $(\widetilde{\boldsymbol{z}}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}})$, and $(\widetilde{\boldsymbol{z}}_i^{\mathrm{T}}, \widetilde{\boldsymbol{z}}_i^{\mathrm{F}})$. We use $S_i^{\mathrm{TF}} = d(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}}, \mathcal{D}^{\text{pret}})$ to denote the distance for embedding pair $(\boldsymbol{z}_i^{\mathrm{T}}, \boldsymbol{z}_i^{\mathrm{F}})$. Similarly, we define $S_i^{\mathrm{T}\widetilde{\mathrm{F}}}$, $S_i^{\widetilde{\mathrm{T}}\mathrm{F}}$, and $S_i^{\widetilde{\mathrm{T}}\widetilde{\mathrm{F}}}$. Note, in the time-frequency domain, we don't consider the pair $(\boldsymbol{z}_i^{\mathrm{T}}, \widetilde{\boldsymbol{z}}_i^{\mathrm{T}})$ where the two embeddings are from from the same domain (*i.e.*, time domain). It's the same for pair $(\boldsymbol{z}_i^{\mathrm{F}}, \widetilde{\boldsymbol{z}}_i^{\mathrm{F}})$. We have already considered the above two pairs in the calculation $\mathcal{L}_{\mathrm{T},i}$ and $\mathcal{L}_{\mathrm{F},i}$.

Next, let's closely observe S_i^{TF} and $S_i^{\mathrm{T}\widetilde{\mathrm{F}}}$ that involve three embeddings: $\boldsymbol{z}_i^{\mathrm{T}}$, $\boldsymbol{z}_i^{\mathrm{F}}$, and $\widetilde{\boldsymbol{z}}_i^{\mathrm{F}}$. Here, $\boldsymbol{z}_i^{\mathrm{T}}$ and $\boldsymbol{z}_i^{\mathrm{F}}$ are learned from the original sample ($\boldsymbol{x}_i^{\mathrm{T}}$ and $\boldsymbol{x}_i^{\mathrm{F}}$) while $\widetilde{\boldsymbol{z}}_i^{\mathrm{F}}$ is learned from the augmented $\widetilde{\boldsymbol{x}}_i^{\mathrm{F}}$. Thus, intuitively, $\boldsymbol{z}_i^{\mathrm{T}}$ should be closer to $\boldsymbol{z}_i^{\mathrm{F}}$ in comparison to $\widetilde{\boldsymbol{z}}_i^{\mathrm{F}}$. Motivated by the relative relationship, we encourage the proposed model to learn a S_i^{TF} that is smaller than $S_i^{\mathrm{T}\widetilde{\mathrm{F}}}$. Inspired by the triplet loss [56, 57], we design $(S_i^{\mathrm{TF}} - S_i^{\mathrm{T}\widetilde{\mathrm{F}}} + \delta)$ as a term of TF-C loss where δ is a given constant margin. This term optimize the model towards a smaller S_i^{TF} and relatively larger $S_i^{\mathrm{T}\widetilde{\mathrm{F}}}$.

Similarly, S_i^{TF} is supposed to be smaller than $S_i^{\widetilde{\mathrm{T}}\mathrm{F}}$ and $S_i^{\widetilde{\mathrm{T}}\widetilde{\mathrm{F}}}$, respectively. In summary, we calculate the consistency loss $\mathcal{L}_{\mathrm{C},i}$ for sample $\boldsymbol{x}_i$ by:

$$\mathcal{L}_{\mathrm{C},i} = \sum_{S^{\text{pair}}} (S_i^{\mathrm{TF}} - S_i^{\text{pair}} + \delta), \quad S^{\text{pair}} \in \{S_i^{\mathrm{T}\widetilde{\mathrm{F}}}, S_i^{\widetilde{\mathrm{T}}\mathrm{F}}, S_i^{\widetilde{\mathrm{T}}\widetilde{\mathrm{F}}}\}, \tag{3}$$

where S_i^{pair} denotes the time-frequency embedding pair in which at least one embedding that is derived from augmented sample in time domain or frequency domain and δ is a pre-defined constant. By combining three triplet loss items, $\mathcal{L}_{\mathrm{C}}$ stimulates the pre-training model to consider the consistency between time-based and frequency-based embeddings in model optimization. Although the Eq. 3 does not explicitly measure the loss across different time series samples (*e.g.*, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$), the cross-sample relationships are implicitly covered in the calculation of S_i^{TF} and S_i^{pair}.

4.4 Implementation and technical details

The overall loss function in pre-training has three terms. First, time-based contrastive loss $\mathcal{L}_{\text{T}}$ urges the model to learn embeddings invariant to temporal augmentations. Second, frequency-based contrastive loss $\mathcal{L}_{\text{F}}$ promotes learning of embeddings invariant to frequency spectrum-based augmentations. Third, the consistency loss $\mathcal{L}_{\text{C}}$ guides the model to retain the consistency between time-based and frequency-based embeddings. In summary, pre-training loss is the following:

$$\mathcal{L}_{\text{TF-C},i} = \lambda(\mathcal{L}_{\text{T},i} + \mathcal{L}_{\text{F},i}) + (1-\lambda)\mathcal{L}_{\text{C},i} \tag{4}$$

where λ controls the relative importance of contrastive and consistency losses. We calculate the total loss by summing $\mathcal{L}_{\text{TF-C},i}$ across all pre-training samples. In implementation, the contrastive losses are calculated within the batch. Recall our problem definition, the model $\mathcal{F}$ we want to learn is the combination of neural networks G_{T}, R_{T}, G_{F}, and R_{F}. When pre-training is completed, we store parameters of entire model, and denote it as $\mathcal{F}(\cdot, \Theta)$ where Θ represents all trainable parameters. When a test sample $\boldsymbol{x}_i^{\text{tune}}$ is presented, fine-tuned model $\mathcal{F}(\cdot, \Phi)$ generates embedding $\boldsymbol{z}_i^{\text{tune}}$ by concatenating time- and frequency-based embeddings as: $\boldsymbol{z}_i^{\text{tune}} = \mathcal{F}(\boldsymbol{x}_i^{\text{tune}}, \Phi) = [\boldsymbol{z}_i^{\text{tune,T}}; \boldsymbol{z}_i^{\text{tune,F}}]$.

5 Experiments

Datasets. We evaluate the new approach on 8 diverse datasets. (1) **SleepEeg** [58] has 371,055 univariate brainwaves (EEG; 100 Hz) collected from 197 individuals. Each sample associates with one among five sleeping stages. (2) **Epilepsy** [59] monitors the brain activities of 500 subjects with single-channel EEG sensor (174 Hz). Sample is labelled in binary based on whether the subject have epilepsy or not. (3) **FD-A** [60] gathers the vibration signals from rolling bearing from a mechanical system aiming at fault detection. Every sample has 5,120 timestamps and an indicator for one out of three mechanical device states. (4) **FD-B** [60] has the same setting as the **FD-A** but the rolling bearings are performed in different working conditions (*e.g.*, varying rotational speed). (5) **HAR** [61] has 10,299 9-dimension samples from 6 daily activities. (6) **Gesture** [62] includes 640 samples that are collected from 8 hand gestures recorded by accelerometer. (7) **ECG** [27] contains 8,528 single-sensor ECG recordings with sorted into four classes based on human physiology. (8) **EMG** [63] consists of 163 EMG samples with 3-class labels implying muscular diseases. Dataset labels are not used in pre-training. Further dataset statistics are in Appendix B and Appendix Table 3.

Baselines. We consider 8 baseline methods. This includes 6 state-of-the-art methods: TS-SD [11], TS2vec [46], CLOCS [40], Mixing-up [17], TS-TCC [47], and SimCLR [39]. The TS2Vec, TS-TCC, and SimCLR are designed for representation learning on a single dataset (not across datasets), we apply them fit our settings to make the results comparable. To examine utility of pre-training, we consider two approaches (non-deep learning KNN model and the Random Init. approach, which randomly initializes the fine-tuning model) that are applied directly to fine-tuning datasets without any pre-training. The evaluation metrics are accuracy, precision (macro-averaged), recall, F1 score, AUROC, and AUPRC. Method details and hyper-parameter selection are in Appendix D.

Implementation. We use two 3-layer 1-D ResNets [64] as backbones for G_{T} and G_{F} encoders. Datasets contain long series (samples in FD-A and FD-B have 5,120 observations), and preliminary experiments identified ResNet as a better option than a Transformer variant [65]. We use 2 fully-connected layers for R_{T} and R_{F} (no sharing of parameters). We set $\Delta = 1$ and $\alpha = 0.5$ in frequency augmentations and $\tau = 0.2$, $\delta = 1$, $\lambda = 0.5$ in loss functions. Reported are mean and standard deviation values across 5 independent runs (both pre-training and fine-tuning) on the same data split. Results for KNN (K=2) do not change so standard deviation is zero. Details are in Appendix E.

5.1 Results: One-to-one pre-training evaluation

Setup. In one-to-one evaluation, we pre-train a model on *one* pre-training dataset and use it for fine-tuning on *one* target dataset only. Scenario 1 (SleepEeg → Epilepsy): Pre-training is done on SleepEeg and fine-tuning on Epilepsy. While both datasets describe a single-channel EEG, the signals are from different channels/positions on scalp, monitor different physiology (sleep vs. epilepsy), and are collected from different patients. Scenario 2 (FD-A → FD-B): Datasets describe mechanical devices that operate in different working conditions, including rotational speed, load torque, and radial force. Scenario 3 (HAR → Gesture): Datasets record different activities (6 types of human daily activities vs. 8 hand gestures). While both datasets contain acceleration signals,

Table 1: One-to-one pre-training evaluation (Scenario 3). Pre-training is performed on HAR dataset, followed by fine-tuning on GESTURE. Results for other three scenarios are shown in Appendix Tables 4-6.

Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
Non-DL (KNN)	0.6766±0.0000	0.6500±0.0000	0.6821±0.0000	0.6442±0.0000	0.8190±0.0000	0.5231±0.0000
Random Init.	0.4219±0.0865	0.4751±0.0925	0.4963±0.1026	0.4886±0.0967	0.7129±0.1206	0.3358±0.1194
TS-SD	0.6937±0.0533	0.6806±0.0496	0.6883±0.0525	0.6785±0.0495	0.8708±0.0305	0.6261±0.0790
TS2vec	0.6453±0.0260	0.6287±0.0339	0.6451±0.0218	0.6261±0.0294	0.8890±0.0054	0.6670±0.0118
CLOCS	0.4731±0.0229	0.4639±0.0432	0.4766±0.0266	0.4392±0.0198	0.8161±0.0068	0.4916±0.0103
Mixing-up	0.7183±0.0123	0.7001±0.0166	0.7183±0.0123	0.6991±0.0145	**0.9127±0.0018**	0.7654±0.0071
TS-TCC	0.7593±0.0242	0.7668±0.0257	0.7566±0.0231	0.7457±0.0210	0.8866±0.0040	0.7217±0.0121
SimCLR	0.4383±0.0652	0.4255±0.1072	0.4383±0.0652	0.3713±0.0919	0.7721±0.0559	0.4116±0.0971
TF-C (Ours)	**0.7824±0.0237**	**0.7982±0.0496**	**0.8011±0.0322**	**0.7991±0.0296**	0.9052±0.0136	**0.7861±0.0149**

HAR has 9 channels while GESTURE has 1 channel. Scenario 4 (ECG → EMG): While both are physiological datasets, the ECG records the electrical signal from the heart whereas EMG measures muscle response in response to a nerve's stimulation of the muscle. We note that the discrepancy between pre-training and fine-tuning datasets in the above four scenarios is substantial, covering diverse kinds of variation in time series datasets: varying semantic meaning, sampling frequency, time series length, number of classes, and system factors (*e.g.*, number of devices or subjects). The setup is further challenged by relatively small number of samples available for fine-tuning (EPILEPSY: 60; FD-B: 60; GESTURE: 480; EMG: 122). Further details are in Appendix B-C.

Results. The results for four scenarios are shown in Table 1 and Appendix Tables 4-6. Overall, TF-C model has won 16 out of 24 tests (6 metrics in 4 scenarios), and is the second best performer in only 8 other tests. We report all metrics but discuss the F1 score in the following. On average, our TF-C model claims a large margin of 15.4% over all baselines. Although the strongest baseline is varying (such as TS-TCC in Scenario 2; Mixing-up in Scenario 3), our model outperforms the strongest baselines by 1.5% across all scenarios. Specifically, when using HAR-pre-trained model for fine-tuning on GESTURE (Scenario 3) (Table 1), the TF-C approach achieves the highest performance of 79.91% which yields a margin of 7.2% over the best baseline TS-TCC (74.57%). One potential reason that Scenario 3 is the most complex dataset in terms of classes (HAR has 6 classes while GESTURE has 8 classes) that is difficult to model. Thee assumption is proved from side by the relative lower performances in Scenario 3 ($< 80\%$) than other scenarios ($> 90\%$).

5.2 Results: One-to-many pre-training evaluation

Setup. In one-to-many evaluation, pre-training is done using *one* dataset followed by fine-tuning on *multiple* target datasets independently without starting pre-training from scratch. Out of eight datasets, SLEEPEEG has most complex temporal dynamics [66] and is the largest (371,055 samples). For that reason, we pre-train a model on SLEEPEEG and separately fine-tune a well-pre-trained model on EPILEPSY, FD-B, GESTURE, and EMG.

Results. Results are shown in Table 2. Intuitively, there are less common between EEG signals and vibration, acceleration and EMG, we expect that the transfer learning will be less effective compared to the one-to-one evaluation. The pre-training and fine-tuning datasets are largely different in the bottom three blocks (SLEEPEEG → {FD-B, GESTURE, EMG}). The large gap reasonably leads to a deterioration in baseline performances, however, our model has obviously higher tolerance to knowledge transfer across datasets with large difference. Notably, We find that the proposed model with TF-C earned best performance in 14 out of 16 settings in the three challenging settings: indicating our TF-C assumption is universal in time series. For example, our approach outperforms the strongest baseline by 8.4% (in precision) when fine-tuning on GESTURE. Our model has large potential to serve as a universal model when there is no large pre-training dataset that is similar to the small fine-tuning dataset. Further, the TF-C approach consistently outperforms KNN and Random Init. (which are not pre-trained) by a large margin of 42.8% and 25.1% (both in F1 score) on average.

Ablation study. We wanted to evaluate how relevant model component are for effective pre-training. As shown in Appendix Table 7 (SLEEPEEG → GESTURE), removing $\mathcal{L}_{\text{C}}$, $\mathcal{L}_{\text{T}}$, and $\mathcal{L}_{\text{F}}$ result in performance degradation (precision) of 6.1%, 7.2%, and 6.7%, respectively. To validate that the performance increment is not solely brought by a third loss term no matter what consistency it measures, we replaced consistency loss $\mathcal{L}_{\text{C}}$ with a loss term measuring the consistency within time space (named $\mathcal{L}_{\text{TT-C}}$) or within frequency space (named $\mathcal{L}_{\text{FF-C}}$). Results show our consistency loss outperforms $\mathcal{L}_{\text{TT-C}}$ and $\mathcal{L}_{\text{FF-C}}$ by 5.3% and 7.2% (accuracy), respectively. We also find that full

Table 2: One-to-many pre-training evaluation. Pre-training is performed on SLEEPEEG dataset, followed by an independent fine-tuning on EPILEPSY, FD-B, GESTURE, and EMG.

Scenarios	Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
SLEEPEEG ↓ EPILEPSY	**Non-DL (KNN)**	0.8525±0.0000	0.8639±0.0000	0.6431±0.0000	0.6791±0.0000	0.6434±0.0000	0.6279±0.0000
	Random Init.	0.8983±0.0656	0.9213±0.1369	0.7447±0.1135	0.7959±0.1208	0.8578±0.2153	0.6489±0.1926
	TS-SD	0.8952±0.0522	0.8018±0.2244	0.7647±0.1485	0.7767±0.1855	0.7677±0.2452	0.7940±0.1825
	TS2vec	0.9395±0.0044	0.9059±0.0116	0.9039±0.0118	0.9045±0.0067	0.9587±0.0086	0.9430±0.0103
	CLOCS	**0.9507±0.0027**	0.9301±0.0067	**0.9127±0.0165**	**0.9206±0.0066**	0.9803±0.0023	0.9609±0.0116
	Mixing-up	0.8021±0.0000	0.4011±0.0000	0.5000±0.0000	0.4451±0.0000	0.9743±0.0081	0.9618±0.0104
	TS-TCC	0.9253±0.0098	0.9451±0.0049	0.8181±0.0257	0.8633±0.0215	0.9842±0.0034	0.9744±0.0043
	SimCLR	0.9071±0.0344	0.9221±0.0166	0.7864±0.1071	0.8178±0.0998	0.9045±0.0539	0.9128±0.0205
	TF-C (Ours)	0.9495±0.0249	**0.9456±0.0108**	0.8908±0.0216	0.9149±0.0534	**0.9811±0.0237**	**0.9703±0.0199**
SLEEPEEG ↓ FD-B	**Non-DL (KNN)**	0.4473±0.0000	0.2847±0.0000	0.3275±0.0000	0.2284±0.0000	0.4946±0.0000	0.3308±0.0000
	Random Init.	0.4736±0.0623	0.4829±0.0529	0.5235±0.1023	0.4911±0.0590	0.7864±0.0349	0.7528±0.0254
	TS-SD	0.5566±0.0210	0.5710±0.0535	0.6054±0.0272	0.5703±0.0328	0.7196±0.0113	0.5693±0.0532
	TS2vec	0.4790±0.0113	0.4339±0.0092	0.4842±0.0197	0.4389±0.0107	0.6463±0.0130	0.4442±0.0162
	CLOCS	0.4927±0.0310	0.4824±0.0316	0.5873±0.0387	0.4746±0.0485	0.6992±0.0099	0.5501±0.0365
	Mixing-up	0.6789±0.0246	0.7146±0.0343	**0.7613±0.0198**	0.7273±0.0228	0.8209±0.0035	0.7707±0.0042
	TS-TCC	0.5499±0.0220	0.5279±0.0293	0.6396±0.0178	0.5418±0.0338	0.7329±0.0203	0.5824±0.0468
	SimCLR	0.4917±0.0437	0.5446±0.1024	0.4760±0.0885	0.4224±0.1138	0.6619±0.0219	0.5009±0.0477
	TF-C (Ours)	**0.6938±0.0231**	**0.7559±0.0349**	0.7202±0.0257	**0.7487±0.0268**	**0.8965±0.0135**	**0.7871±0.0267**
SLEEPEEG ↓ GESTURE	**Non-DL (KNN)**	0.6833±0.0000	0.6501±0.0000	0.6833±0.0000	0.6443±0.0000	0.8190±0.0000	0.5232±0.0000
	Random Init.	0.4219±0.0629	0.4751±0.0175	0.4963±0.0679	0.4886±0.0459	0.7129±0.0166	0.3358±0.1439
	TS-SD	0.6922±0.0444	0.6698±0.0472	0.6867±0.0488	0.6656±0.0443	0.8725±0.0324	0.6185±0.0966
	TS2vec	0.6917±0.0333	0.6545±0.0358	0.6854±0.0349	0.6570±0.0392	0.8968±0.0123	0.6989±0.0346
	CLOCS	0.4433±0.0518	0.4237±0.0794	0.4433±0.0518	0.4014±0.0602	0.8073±0.0109	0.4460±0.0384
	Mixing-up	0.6933±0.0231	0.6719±0.0232	0.6933±0.0231	0.6497±0.0306	0.8915±0.0261	0.7279±0.0558
	TS-TCC	0.7188±0.0349	0.7135±0.0352	0.7167±0.0373	0.6984±0.0360	0.9099±0.0085	0.7675±0.0201
	SimCLR	0.4804±0.0594	0.5946±0.1623	0.5411±0.1946	0.4955±0.1870	0.8131±0.0521	0.5076±0.1588
	TF-C (Ours)	**0.7642±0.0196**	**0.7731±0.0355**	**0.7429±0.0268**	**0.7572±0.0311**	**0.9238±0.0159**	**0.7961±0.0109**
SLEEPEEG ↓ EMG	**Non-DL (KNN)**	0.4390±0.0000	0.3772±0.0000	0.5143±0.0000	0.3979±0.0000	0.6025±0.0000	0.4084±0.0000
	Random Init.	0.7780±0.0729	0.5909±0.0625	0.6667±0.0135	0.6238±0.0267	0.9109±0.1239	0.7771±0.1427
	TS-SD	0.4606±0.0000	0.1545±0.0000	0.3333±0.0000	0.2111±0.0000	0.5005±0.0126	0.3775±0.0110
	TS2vec	0.7854±0.0318	**0.8040±0.0750**	0.6785±0.0396	0.6766±0.0501	**0.9331±0.0164**	**0.8436±0.0372**
	CLOCS	0.6985±0.0323	0.5306±0.0750	0.5354±0.0291	0.5139±0.0409	0.7923±0.0573	0.6484±0.0680
	Mixing-up	0.3024±0.0534	0.1099±0.0126	0.2583±0.0456	0.1541±0.0204	0.4506±0.1718	0.3660±0.1635
	TS-TCC	0.7889±0.0192	0.5851±0.0974	0.6310±0.0991	0.5904±0.0952	0.8851±0.0113	0.7939±0.0386
	SimCLR	0.6146±0.0582	0.5361±0.1724	0.4990±0.1214	0.4708±0.1486	0.7799±0.1344	0.6392±0.1596
	TF-C (Ours)	**0.8171±0.0287**	0.7265±0.0353	**0.8159±0.0289**	**0.7683±0.0311**	0.9152±0.0211	0.8329±0.0137

fine-tuning (optimizing the whole model) performs slightly better than partial fine-tuning (updating only final layers in R_{T} and R_{F}), by 5.4% in F1 score.

6 Conclusion

We introduce a pre-training model using the TF-C inductive bias in time series to support knowledge transfer across time-series datasets. The approach uses self-supervised contrastive estimation to inject TF-C in a pre-training model, bringing the encoded time-based and frequency-based representations and their local neighborhoods closer together in the latent space. Experiments on eight datasets and four pre-training scenarios show the TF-C approach can outperform eight state-of-the-art baselines. Our findings have implications for building broadly generalizable pre-training models for time series.

Limitations and future directions. We posit that TF-C can serve as a generalizable property to support pre-training on diverse time series datasets. There might exist additional generalizable properties, such as temporal autoregressive processes, that could also proved useful for pre-training on time series. Our method expects the input time series to be regularly sampled, however, our model can handle irregularly sampled time series with an encoder (such as Raindrop [67] and SeFT [68]) that can learn time-based representations from irregular time series. For frequency encoder inputs $\boldsymbol{x}_i^{\mathrm{F}}$, possible alternatives include resampling or interpolation to obtain regularly sampled signals and using regular or non-uniform FFT operations.

References

[1] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. Scientific Data, 6(1):96, December 2019.

[2] Shahbaz Rezaei and Xin Liu. Deep Learning for Encrypted Traffic Classification: An Overview. IEEE Communications Magazine, 57(5):76–81, May 2019.

[3] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. Skilful precipitation nowcasting using deep generative models of radar. Nature, 597(7878):672–677, September 2021.

[4] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing, 90:106181, May 2020.

[5] Bing Su and Ji-Rong Wen. Temporal alignment prediction for supervised representation learning and few-shot sequence classification. In ICLR, 2022.

[6] Yixiang Deng, Lu Lu, Laura Aponte, Angeliki M Angelidi, Vera Novak, George Em Karniadakis, and Christos S Mantzoros. Deep transfer learning and data augmentation improve glucose levels prediction in type 2 diabetes patients. NPJ Digital Medicine, 4(1):1–13, 2021.

[7] Quentin Rebjock, Baris Kurt, Tim Januschowski, and Laurent Callot. Online false discovery rate control for anomaly detection in time series. NeurIPS, 34, 2021.

[8] Fan-Keng Sun, Chris Lang, and Duane Boning. Adjusting for autocorrelated errors in neural networks for time series. NeurIPS, 34, 2021.

[9] Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, and Qun Liu. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. ICLR, 2022.

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

[11] Pengxiang Shi, Wenwen Ye, and Zheng Qin. Self-Supervised Pre-training for Time Series Classification. In 2021 International Joint Conference on Neural Networks (IJCNN), pages 1–8, Shenzhen, China, July 2021. IEEE.

[12] Weixia Dang, Biyu Zhou, Lingwei Wei, Weigang Zhang, Ziang Yang, and Songlin Hu. Ts-bert: Time series anomaly detection via pre-training model bert. In International Conference on Computational Science, pages 209–223. Springer, 2021.

[13] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In CVPR, pages 3558–3568, 2021.

[14] Kailai Sun, Zuchao Li, and Hai Zhao. Multilingual pre-training with universal dependency learning. NeurIPS, 34, 2021.

[15] Rui Ye and Qun Dai. Implementing transfer learning across different datasets for time series forecasting. Pattern Recognition, 109:107617, 2021.

[16] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Transfer learning for time series classification. In 2018 IEEE international conference on big data (Big Data), pages 1367–1376. IEEE, 2018.

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

[18] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. Transfer learning for clinical time series analysis using deep neural networks. Journal of Healthcare Informatics Research, 4(2):112–137, 2020.

[19] Amiel Meiseles and Lior Rokach. Source model selection for deep learning in the time series domain. IEEE Access, 8:6190–6200, 2020.

[20] Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. NeurIPS, 34, 2021.

[21] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. 2018. Publisher: arXiv Version Number: 2.

[22] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[23] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In NeurIPS, 2020.

[24] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In NeurIPS, 2020.

[25] Simon Rogers, Derek Sleeman, and John Kinsella. Investigating the disagreement between clinicians' ratings of patients in icus. IEEE Journal of Biomedical and Health Informatics, 17(4):843–852, 2013.

[26] Leonard M Horowitz, Rita de Sales French, Kirk D Wallis, David L Post, and Ellen Y Siegelman. The prototype as a construct in abnormal psychology: Ii. clarifying disagreement in psychiatric judgments. Journal of Abnormal Psychology, 90(6):575, 1981.

[27] Gari Clifford, Chengyu Liu, Benjamin Moody, Li-wei Lehman, Ikaro Silva, Qiao Li, Alistair Johnson, and Roger Mark. AF Classification from a Short Single Lead ECG Recording: the Physionet Computing in Cardiology Challenge 2017. September 2017.

[28] Mitchell L Gordon, Kaitlyn Zhou, Kayur Patel, Tatsunori Hashimoto, and Michael S Bernstein. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In CHI, pages 1–14, 2021.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. 2018. Publisher: arXiv Version Number: 2.

[30] Pritam Sarkar and Ali Etemad. Self-supervised learning for ecg-based emotion recognition. In ICASSP, 2020.

[31] Joseph Y. Cheng, Hanlin Goh, Kaan Dogrusoz, Oncel Tuzel, and Erdrin Azemi. Subject-Aware Contrastive Learning for Biosignals. arXiv:2007.04871 [cs, eess, stat], June 2020. arXiv: 2007.04871.

[32] Sriram Ravula, Georgios Smyrnis, Matt Jordan, and Alexandros G Dimakis. Inverse problems leveraging pre-trained contrastive representations. NeurIPS, 34, 2021.

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709 [cs, stat], June 2020. arXiv: 2002.05709.

[34] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised Pre-training for Object Detection with Transformers. arXiv:2011.09094 [cs], April 2021. arXiv: 2011.09094.

[35] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised Representation Learning by Sorting Sequences. arXiv:1708.01246 [cs], August 2017. arXiv: 1708.01246.

[36] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In ICCV, pages 2959–2968, 2019.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs], May 2019. arXiv: 1810.04805.

[38] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. In arXiv:2001.08317, 2020.

[39] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. Exploring Contrastive Learning in Human Activity Recognition for Healthcare. arXiv:2011.11542 [cs, eess], February 2021. arXiv: 2011.11542.

[40] Dani Kiyasseh, Tingting Zhu, and David A Clifton. CLOCS: Contrastive Learning of Cardiac Signals Across Space, Time, and Patients. page 10, 2020.

[41] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. ICLR, 2022.

[42] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: Task-oriented alignment for unsupervised domain adaptation. NeurIPS, 34, 2021.

[43] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. ICLR, 2022.

[44] Bernd Illing, Jean Ventura, Guillaume Bellec, and Wulfram Gerstner. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. NeurIPS, 34, 2021.

[45] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. arXiv:2106.00750 [cs, stat], June 2021. arXiv: 2106.00750.

[46] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. TS2Vec: Towards Universal Representation of Time Series. arXiv:2106.10466 [cs], February 2022. arXiv: 2106.10466.

[47] 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:2106.14112 [cs], June 2021. arXiv: 2106.14112.

[48] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. 2022. Publisher: arXiv Version Number: 3.

[49] Rob J. Hyndman and George Athanasopoulos. Forecasting: principles and practice. Otexts, online, open-access textbook, Lexington, Ky., 2nd edition edition, 2018.

[50] Ronald Newbold Bracewell and Ronald N Bracewell. The Fourier transform and its applications, volume 31999. McGraw-hill New York, 1986.

[51] Leon Cohen. Time-frequency analysis. Prentice Hall signal processing series. Prentice Hall PTR, Englewood Cliffs, N.J, 1995.

[52] Henri J Nussbaumer. The fast fourier transform. In Fast Fourier Transform and Convolution Algorithms, pages 80–111. Springer, 1981.

[53] Patrick Flandrin. Time-Frequency/Time-Scale Analysis: Time-Scale Analysis. Elsevier, Burlington, 1998. OCLC: 476160927.

[54] Antonia Papandreou-Suppappola, editor. Applications in Time-Frequency Signal Processing. CRC Press, 1 edition, October 2018.

[55] Ryan Soklaski, Michael Yee, and Theodoros Tsiligkaridis. Fourier-based augmentations for improved robustness and uncertainty calibration. NeurIPS'W, 2021.

[56] Elad Hoffer and Nir Ailon. Deep metric learning using Triplet network. 2014. Publisher: arXiv Version Number: 4.

[57] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Procedings of the British Machine Vision Conference 2016, pages 119.1–119.11, York, UK, 2016. British Machine Vision Association.

[58] B. Kemp, A.H. Zwinderman, B. Tuk, H.A.C. Kamphuisen, and J.J.L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. IEEE Transactions on Biomedical Engineering, 47(9):1185–1194, September 2000.

[59] Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E, 64(6):061907, November 2001.

[60] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification. In European Conference of the Prognostics and Health Management Society, 2016.

[61] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. Computational Intelligence, page 6, 2013.

[62] Jiayang Liu, Zhen Wang, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. In 2009 IEEE International Conference on Pervasive Computing and Communications, pages 1–9, Galveston, TX, USA, March 2009. IEEE.

[63] Ary L. Goldberger, Luis A. N. 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), June 2000.

[64] Amrutha Ramanathan and James McDermott. Fall detection with accelerometer data using residual networks adapted to multi-variate time series classification. In IJCNN, pages 1–8, 2021.

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

[66] Xiang Zhang and Lina Yao. Deep Learning for EEG-Based Brain–Computer Interfaces: Representations, Algorithms and Applications. World Scientific, 2021.

[67] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In ICLR, 2022.

[68] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In ICML, pages 4353–4363, 2020.

Broader Impacts

Our proposed novel framework for time series unsupervised pre-training improves classification performance on target datasets in different application scenarios. The recognition of time-frequency consistency as a universal property specific to time series data is a weak assumption that enables effective, task- and domain-agnostic transfer learning. We believe our work will inspire the research community to uncover other universal properties for transfer learning. We also hope our work will also attract more researchers to the more general problem of time series representation learning which is still underappreciated relative to problems from CV and NLP fields.

On the society level, our work, along the line of transfer learning, can facilitate more efficient use of time series data in various settings. For example, in medical settings, some diseases of clinical interest may have very small labelled dataset. In this case, unlabelled data from patients of different diseases but with similar underlying physiological conditions can be used to pre-train the model. However, practitioners need to be aware of the limitations of the model, including that it may make biased predictions. Specifically, bias may exist in the source dataset used for pre-training due to an imbalance of samples from subjects of different demographic attributes. Also, the standardized medical protocols for collecting these datasets might be unsuitable for subjects with certain physiological attributes, creating unforseen bias that may be transferred to the finetuning phase.

All datasets in this paper are publicly available and are not associated with any privacy or security concern. Furthermore, we have followed guidelines on responsible use specified by primary authors of the datasets used in the current work.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] In abstract and introduction, we claim that TF-C is a generalizable property of time series that can support pre-training, which is well-justified in Sec. 3 and experimentally demonstrated in Sec. 5 (our model consistently performs comparatively to or above baseline methods).
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Broader Impact on Page 10.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Yes, we include an anonymous link (see Abstract) that provides the source codes with all implementation details, implementation of baselines, and eight datasets. The link will be updated to an non-anonymous link after acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyper-parameters, how they were chosen)? [Yes] See implementation details in Sec. 5. See Appendix E for baseline architectures and hyper-parameter settings. More details can be found in the included URL.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We run experiments for 5 times and report the average value with standard deviation. See Table 1, Table 2, and Appendix Tables 4-6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix E.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We used eight existing datasets and 6 state-of-the-art baselines in contrastive learning and pre-training for time series. We cited the creators for every exist asset we used. See Sec. 5.

(b) Did you mention the license of the assets? [Yes] All dataset licenses are mentioned in the Appendix E.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See the anonymous link in Abstract.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All data we use is freely available for download, without any requirement to re-contact the data curator.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Our datasets are public, well-established, and do not contain PII or offensive content

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]