Learning Without Augmenting: Unsupervised Time Series Representation Learning via Frame Projections

Berken Utku Demirel
Department of Computer Science
ETH Zürich, Switzerland
berken.demirel@inf.ethz.ch

Christian Holz
Department of Computer Science
ETH Zürich, Switzerland
christian.holz@inf.ethz.ch

Abstract

Self-supervised learning (SSL) has emerged as a powerful paradigm for learning representations without labeled data. Most SSL approaches rely on strong, well-established, handcrafted data augmentations to generate diverse views for representation learning. However, designing such augmentations requires domain-specific knowledge and implicitly imposes representational invariances on the model, which can limit generalization. In this work, we propose an unsupervised representation learning method that replaces augmentations by generating views using orthonormal bases and overcomplete frames. We show that embeddings learned from orthonormal and overcomplete spaces reside on distinct manifolds, shaped by the geometric biases introduced by representing samples in different spaces. By jointly leveraging the complementary geometry of these distinct manifolds, our approach achieves superior performance without artificially increasing data diversity through strong augmentations. We demonstrate the effectiveness of our method on nine datasets across five temporal sequence tasks, where signal-specific characteristics make data augmentations particularly challenging. Without relying on augmentation-induced diversity, our method achieves performance gains of up to 15–20% over existing self-supervised approaches. Source code: https://github.com/eth-siplab/Learning-with-FrameProjections

1 Introduction

Sample efficient unsupervised representation learning is a critical open challenge in deep learning. While recent self-supervised techniques have shown strong performance in several tasks, they typically rely on handcrafted, aggressive data augmentations that expose the model to different versions of input samples in each training epoch to increase data diversity [1–4]. Moreover, these techniques only perform well when augmentations are carefully optimized for the downstream task; otherwise, model performance drops significantly [1, 5]. Even when the downstream task is known, designing effective augmentations can be challenging for data types lacking a well-defined structure, i.e., text, tabular, signals [6–9], as the overly strong augmentations can cause model collapse [10].

Simply increasing sample diversity through random augmentations does not guarantee improved performance in representation learning. Augmentations are only effective if the augmented views of a sample have sufficient representational similarity with views of other intra-class samples [11, 12]. Without adequate representational similarity, the model struggles to generalize across classes, leading to degraded performance on downstream tasks where intra-class variability is not fully captured through augmentations [13–15]. Since achieving representational similarity in high-dimensional spaces is challenging [16], a key limitation of augmentation-based self-supervised learning lies in its reliance on handcrafted transformations. These transformations can distort critical class-level structure, leading to severe performance degradation in complex or heterogeneous data regimes.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Another important limitation of augmentation-based SSL is the inductive bias and feature suppression introduced by both the augmentation process and the optimization objective [17, 18]. When trained with strong augmentations, models tend to focus on a subset of predictive features, typically those aligned with the enforced invariances, while suppressing other features that may be critical for downstream performance [19]. This often leads to reliance on a subset of features, harming generalization [17, 20]. Moreover, the inductive bias introduced by augmentations acts as a double-edged sword: promoting invariance to certain transformations may benefit some tasks but harm others [21, 22]. For instance, rotation-based augmentations are commonly used in activity recognition from inertial measurement units to promote robustness across sensor placements [23]. However, they can obscure orientation-dependent features needed to distinguish between fine-grained activities such as standing and sitting, where subtle differences in sensor orientations are important. These effects are more problematic in signals where the semantic relevance of augmentations varies across tasks.

In this work, we introduce a novel SSL method that generates views by projecting data onto an orthonormal base and an overcomplete frame, and then performs instance discrimination across these spaces. We demonstrate that the learned representations from instance discrimination lie on distinct manifolds, each shaped by the inherent geometric biases of its corresponding projections. Building on this observation, we propose to learn mapping functions that transform the original data's space into alternative latent representations. This mapping enables us to obtain multiple representations using a single encoder augmented with mapping functions to use collections of manifolds.

Our method leverages the inductive bias introduced by projecting data into an orthonormal basis and an overcomplete frame to learn representations. We summarize our contributions as follows:

- We propose a novel self-supervised learning method that projects data into an orthonormal basis and an overcomplete frame to perform instance discrimination across these fixed transformations without increasing the data diversity using handcrafted augmentations.
- We empirically and theoretically show that embeddings from these transformations lie on distinct manifolds shaped by domain-specific geometric biases. We then jointly leverage these complementary structures to improve performance on downstream tasks.
- We demonstrate that our method achieves up to 15–20% performance gains over existing methods on nine datasets across five temporal sequence tasks while using fixed transformations across datasets unlike existing approaches that rely on task-specific augmentations.

2 Method

2.1 Notations

We use lowercase letters (e.g., x) to denote scalar quantities, and bold lowercase letters (e.g., $\mathbf{x}$) to represent vectors, such as time series, while bold uppercase letters (e.g., $\boldsymbol{X}$) are used for matrices. The parametric function is represented as $f_\theta(.)$ where θ is the parameter. The discrete Fourier transformation is denoted as $\mathcal{F}(.)$, yielding a complex variable as $\mathcal{F}_{\mathrm{x}}(k) \in \mathbb{C}^k$, where k is the frequency. The detailed calculations for each operation are given in the Appendix A.

2.2 Setup

We follow the common SSL setup. Given an unlabeled dataset $\mathcal{D} = \{(\mathbf{x}_i)\}_{i=1}^{K}$ where each $\mathbf{x}_i$ is a real-valued sequence of length L with C channels, the goal is to train a learner f_θ that maps inputs to representations $\boldsymbol{h}_i = f_\theta(\mathbf{x}_i)$. To evaluate the learned representations, we train a linear classifier on top of the frozen encoder using a labeled set $\mathcal{D}_l = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{M}$ with $M \ll K$ and $\mathbf{y}_i \in \{1, \ldots, N\}$.

2.3 Orthonormal Bases and Overcomplete Frames

Analyzing signals in different domains is useful for detecting desired patterns, as each domain is tailored to capture specific aspects of the data [24, 25]. Our method employs the Fourier, $\mathcal{F}_x(k)$, and the Gabor wavelet transform, $\mathcal{W}_{\mathrm{x}}(a, b)$, to construct, respectively, an orthonormal basis (using a tight frame) and an overcomplete frame. Equation 1 defines their corresponding discrete transformations.

$$\mathcal{F}_{\mathrm{x}}(k) = \frac{1}{\sqrt{L}} \sum_n \mathrm{x}(n)\, e^{-j\frac{2\pi}{L}kn}, \quad \mathcal{W}_{\mathrm{x}}(a, b) = \frac{1}{\sqrt{a}} \sum_n \mathrm{x}(n)\, \psi\left(\frac{n-b}{a}\right), \tag{1}$$

where ψ is the Gabor frame. We use these two transformations as they are complementary. Specifically, the Fourier transform provides a global overview of the signal's frequency content, while the Gabor wavelet enables localized frequency analysis by zooming on specific time intervals [26].

2.4 Instance Discrimination

Data representations in the Fourier and Gabor wavelet domains are inherently unique, i.e., distinct samples yield distinct transforms. Consequently, the instance discrimination task between the views in different domains is well-defined. Moreover, since these transformations are isometric, the resulting views are not only unique for each sample but also preserve the underlying geometry. These properties offer significant advantages over existing SSL methods relying on strong augmentations, which may distort samples, causing different classes to appear similar or losing task-relevant information [27].

We use the normalized temperature-scaled cross-entropy (NT-Xent) loss [1, 28, 29], based on cosine similarity, with separate encoders and projection heads for each domain. For example, the Fourier branch takes the transformed input $\mathcal{F}_{\mathrm{x}}$ and produces an embedding via $\boldsymbol{z}^{\mathcal{F}} = g_{\mathcal{F}}(f_{\mathcal{F}}(\mathcal{F}_{\mathrm{x}}))$, as shown in Figure 1. The instance discrimination loss calculated across three domains is defined in Equation 2.

$$\ell\big(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_j^{(d')}\big) = -\log \frac{\exp\big(\mathrm{sim}(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_j^{(d')})/\tau\big)}{\sum_{k\neq i} \exp\big(\mathrm{sim}(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_k^{(d')})/\tau\big)}, \quad \text{where } d \neq d', \quad \text{and} \quad d, d' \in \{t, \mathcal{F}, \mathcal{W}\}, \tag{2}$$

where t is the time domain. We compute the final loss as the unweighted sum of all pairwise terms between the time, Fourier, and wavelet domains over a batch of N samples, as shown in Equation 3.

$$\mathcal{L}_{\mathrm{ID}} = \sum_{\substack{d,d' \in \{t,\mathcal{F},\mathcal{W}\} \\ d \neq d'}} \frac{1}{2N} \sum_{k=1}^{N} \Big[\ell\big(\boldsymbol{z}_{k-1}^{(d)}, \boldsymbol{z}_k^{(d')}\big) + \ell\big(\boldsymbol{z}_k^{(d)}, \boldsymbol{z}_{k-1}^{(d')}\big)\Big] \tag{3}$$

2.4.1 Invariances

Strong augmentations in instance discrimination can cause encoders to become overly reliant on a subset of features or invariant to some transformations that discard task-relevant information [17, 30]. Since the designed augmentations implicitly assume a particular set of representational invariances (e.g., invariance to rotation), and can perform poorly when a downstream task violates this assumption (e.g., distinguishing sitting vs standing) [8]. Proposition 2.1 shows that our method avoids this issue.

Proposition 2.1. *Let f_d^* denote an optimal encoder under NT-Xent for domain $d \in \{t, \mathcal{F}, \mathcal{W}\}$. If for some unintended transformation W, the encoder is invariant, i.e., $f_d^*(W\mathrm{x}) = f_d^*(\mathrm{x})$, then for any anchor sample x the NT-Xent loss across domains is lower bounded by the number of negatives.*

$$\ell\Big(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_j^{(d')}\Big) \;\geq\; \log(K+1) > 0,$$

where $K \geq 1$ is the number of negatives that become near-positives due to the invariance.

Proof. At the NT-Xent optimum, positive pairs align perfectly [31, 32],

$$f_d^*(\mathcal{T}(\mathrm{x})) = f_d^*(\mathrm{x}), \quad \forall d,$$

for the domain transformations $\mathcal{T} \in \{t, \mathcal{F}, \mathcal{W}\}$. If f_d^* is also invariant to W, then at least $K \geq 1$ negatives satisfy $\boldsymbol{z}_i = f_d^*(W\mathrm{x}_i)$ with $(f_d^*(\mathrm{x}), \boldsymbol{z}_i) \approx 1$. Even if all other negatives are dissimilar, the denominator of NT-Xent contains at least $K+1$ large terms, yielding

$$\ell(\mathrm{x}) \geq -\log \frac{e^{1/\tau}}{(K+1)\, e^{1/\tau}} = \log(K+1).$$

Thus the loss admits a nontrivial lower bound in the presence of unintended invariance. □

Proposition 2.1 states that, unlike augmentation-based contrastive learning setups which may encourage spurious invariances [8, 30], our method ensures that such invariances cannot minimize the overall optimization objective. The detailed derivation of the proposition is provided in Appendix A.

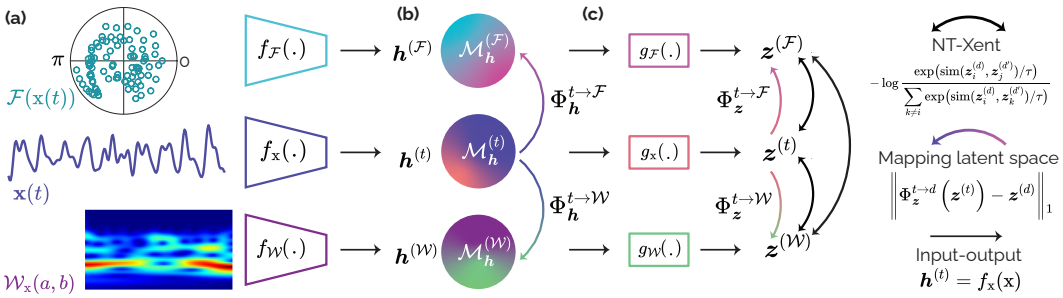


Figure 1: Overview of our method. **(a)** Original data and its transformed versions. The Fourier transformation is given in polar coordinates where we feed to the model magnitude and angle of each harmonic separately. **(b)** Representations $\boldsymbol{h}$ from each encoder lie on distinct manifolds $\mathcal{M}$, with latent mappers $\Phi_{\boldsymbol{h}}^{t\to d}$ translating across domain-specific spaces. **(c)** Embeddings $\boldsymbol{z}$ from each projection, with non-linear mappers used during pre-training to improve predictability across spaces.

2.5 Collections of Manifolds

Transforming data into well-known bases or overcomplete frames has clear benefits for pattern recognition, but doing this at inference time is costly due to the extra encoders and transformations. When a transformation operates in the same space as convolutional filters, prior works showed that applying transformations to the filters instead of the input reduces inference cost while still producing diverse representations (e.g., rotating filters instead of rotating the image and reprocessing it [33]).

However, our method transforms data into the complex space where applying equivalent transformations to neural networks is not straightforward. We therefore propose lightweight latent space mappers Φ that transform the original representations into other spaces to leverage their geometry in downstream tasks. Specifically, we train two mappers, $\Phi_{\boldsymbol{h}}^{t\to d} : \mathcal{M}_{\boldsymbol{h}}^{(t)} \to \mathcal{M}_{\boldsymbol{h}}^{(d)}$, where $d \in \{\mathcal{F}, \mathcal{W}\}$, to approximate the representations produced by the corresponding domain-specific encoders.

Our approach focuses on learning pairwise relationships, such as relative angles between samples across manifolds, rather than mapping individual points between latent spaces. Therefore, unlike prior methods based on affine latent-space mappings [34], we use non-linear mappers. This is motivated by Proposition 2.2, which shows that in high dimensions, representations of the same sample across spaces can become orthogonal, while pairwise angle variation can span the full range. This suggests that preserving pairwise geometry is more challenging than aligning individual points.

Proposition 2.2 (Angle Concentration vs. Pairwise Spread)**.** *Let* $\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \sim \mathrm{Unif}(S^{d-1})$*, where* $\boldsymbol{h}^{(\mathcal{F})} = f_{\mathcal{F}}(\mathcal{F}(\mathrm{x}))$*. Although individual samples across latent spaces tend toward orthogonality, the pairwise angular difference* Δ_{ij} *between distinct samples can span the full range up to* π*.*

$$\arccos\left(\langle \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \rangle\right) = \frac{\pi}{2}, \quad \textit{while} \quad \arccos(\langle \boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)} \rangle) - \arccos(\langle \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle) = \Delta_{ij} \leq \pi \quad (4)$$

Proof.

$$\dim\left(\boldsymbol{h}_i^{(t)\perp} \cap \boldsymbol{h}_j^{(t)\perp}\right) = d - 2 \geq 1 \implies \exists\, \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} : \langle \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle = \cos\phi, \quad \forall \phi \in [0, \pi], \quad (5)$$

Therefore, $|\Delta_{ij}| = \left|\theta_{ij}^{(t)} - \theta_{ij}^{(\mathcal{F})}\right| \leq \pi$. □

Figure 2 illustrates Proposition 2.2 by showing the angle densities between the same ($i = j$, $\arccos\left(\langle \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \rangle\right)$) and different samples ($i \neq j$, $|\Delta_{ij}|$) across domains. We also provide supporting details in Appendix C, and the full proof in Appendix A. The key point of Proposition 2.2 is that preserving near-orthogonality for many pairs does *not* guarantee a global isometry between latent spaces. Because higher-order relations (e.g., triplet geometry and curvature) can change, the spaces can differ globally despite pairwise near-orthogonality.

In our method, we aim to take advantage of these different geometries, shaped by inductive bias of data and architectures, to improve the performance in the downstream tasks. Therefore, we employ two mapper functions ($\Phi_{\boldsymbol{h}}^{t\to d}$) to capture the geometry of latent spaces for other domains.

To improve predictability across latent spaces and reduce estimation error while preserving distinct geometries, we employ lightweight mappers over the embeddings, $\Phi_{\boldsymbol{z}}^{t\to d} : \mathcal{M}_{\boldsymbol{z}}^{(t)} \to \mathcal{M}_{\boldsymbol{z}}^{(d)},\ d \in \{\mathcal{F}, \mathcal{W}\}$, and optimize them jointly using the loss defined in Equation 6.

The overall pre-training loss for encoders is the sum of $\mathcal{L}_{\text{map}}$ and $\mathcal{L}_{\text{ID}}$, with no additional weighting. After pre-training, we freeze the encoders and train the latent space mappers using the same $\mathcal{L}_{\text{map}}$ loss. During inference, we only use the main encoder f_{x} and the latent space mappers $\Phi_{\boldsymbol{h}}^{t\to d}$, excluding all projectors $g(\cdot)$, auxiliary encoders $(f_{\mathcal{F}}, f_{\mathcal{W}})$, and mappers on projected embeddings $\Phi_{\boldsymbol{z}}^{t\to d}$. We provide pseudocode implementation of our method in Appendix B.

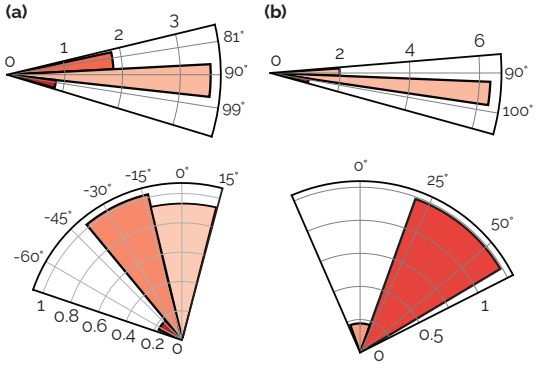


Figure 2: Radial histograms illustrating angle distributions. **(a)** Top: Angle density of $\arccos\left(\langle \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \rangle\right)$; Bottom: Angle density of Δ_{ij}. **(b)** Same illustration from the Gabor wavelet $\boldsymbol{h}^{(\mathcal{W})}$. In both cases, representations of the same samples across domains approach orthogonality, while pairwise angle differences remain widely distributed.

$$\mathcal{L}_{\text{map}} = \frac{1}{N} \left\| \Phi_{\boldsymbol{z}}^{t\to\mathcal{F}} \left(\boldsymbol{z}^{(t)} \right) - \boldsymbol{z}^{(\mathcal{F})} \right\|_1 + \frac{1}{N} \left\| \Phi_{\boldsymbol{z}}^{t\to\mathcal{W}} \left(\boldsymbol{z}^{(t)} \right) - \boldsymbol{z}^{(\mathcal{W})} \right\|_1 \tag{6}$$

3 Experiments

3.1 Datasets

We conducted experiments on nine datasets across five tasks, including heart rate (HR) estimation from photoplethysmography (PPG), step counting and activity recognition using inertial measurements (IMUs), cardiovascular disease classification from electrocardiogram (ECG) and sleep stage classification from electroencephalography (EEG). We provide brief descriptions of each dataset below. Additional details, including pre-training and fine-tuning settings are available in Appendix E.

Heart rate We used the IEEE Signal Processing Cup in 2015 (IEEE SPC) [35], and DaLia [36] for PPG-based heart rate prediction from wrist. We used the leave-one-session-out (LOSO) cross-validation, which evaluates models on subjects/sessions that were not used for training.

Activity recognition We used HHAR [37], and USC [38] for activity recognition from inertial measurement units from smartphones or wearable devices. We evaluated the cross-person generalization performance of the models, that is, the model was evaluated on previously unseen subjects.

Cardiovascular disease (CVD) classification We conducted experiments on China Physiological Signal Challenge (CPSC2018) [39] and Chapman University, Shaoxing People's Hospital (Chapman) datasets [40]. We selected the same four specific leads as in [41] while treating each dataset as a single domain with a small portion of the remaining dataset used for fine-tuning. We split the dataset for fine-tuning and testing based on patients (each patient's recordings appear in only one set).

Step counting We used the Clemson dataset [42], which released for pedometer evaluation. We conducted experiments using wrist IMUs where labels are available through videos.

Sleep stage classification We used the Sleep-EDF dataset, from PhysioBank [43], which includes 197 whole-night PSG sleep recordings, where we used a single EEG channel (i.e., Fpz-Cz) with a sampling rate of 100 Hz, following the same setup as in [44] while using only 10% for fine-tuning.

3.2 Baselines

Fundamentals We compare our method to core SSL approaches in the linear evaluation setting [1]. These include SimCLR [1], BYOL [2], VICReg [3], and Barlow Twins [45]. We also include CLIP [46], since data transformations in our method can be interpreted as different data domains.

Temporal sequences We also compare our method with SSL techniques for temporal data, including TS-TCC [44], TF-C [47], simMTM [48], and TS2Vec [49]. These methods are designed specifically for temporal sequences; for instance, TF-C uses a Fourier encoder during both training and inference, while TS-TCC employs task-specific augmentations with transformer architectures.

3.3 Implementation

We employed ResNet [50] with eight blocks [51], as the backbone for the encoder, with the projector consisting of two fully connected layers. For latent space mapping, we use a lightweight 1D convolutional that downsamples and reconstructs the input with transposed convolutions, preserving dimensions while enabling non-linear transformations with fewer than 1k parameters. Similarly, we use two convolutional encoders for the Fourier- and wavelet-transformed inputs. To ensure a fair comparison, baselines use 384-dimensional encoders as output, while ours uses 128 per encoder.

We train models with a batch size of 1024 for 256 epochs and decay the learning rate using the cosine decay schedule. After pre-training, we train a single linear layer classifier on features extracted from the frozen pre-trained network. The models were optimized using Adam [52] with a learning rate of 0.003, while the linear layer was fine-tuned with a learning rate of 0.03. Reported results are mean and standard deviation values across three independent runs with different seeds. For each dataset, we set the Fourier transform length equal to the signal length while excluding negative frequencies. For the wavelet transform, we use 48 logarithmically spaced scales ranging from 1 to 128 for all datasets. More details about the implementation and architectures are given in Appendix E.3.

4 Results

We report the performance of our method against baselines across nine datasets spanning five tasks in Tables 1 to 3. Overall, our approach achieves up to 20—30% improvements in some tasks, with an average gain of 10—15% across all datasets compared to both general SSL and sequence-specific techniques. Moreover, unlike prior approaches that employ different augmentations, our method uses the same transformation across tasks without increasing the diversity of the training set.

Our main results show that our method outperforms previous techniques by a significant margin in several datasets. To further investigate, we explore whether prior methods can close this gap with comparable modifications. Specifically, we ask the following questions.

1. Our model includes lightweight mappers with additional backbones. We ask if prior methods can match our performance by increasing backbone capacity, that is, by brute-force scaling.
2. Our model employs instance discrimination task with more than one view. Thus, we also ask if existing techniques with multiple positive views can match our performance.

To answer the first question, we increased the backbone capacity of previous fundamental techniques, especially SimCLR, BYOL and Barlow Twins as they represent unique approaches, by 2x compared to our method by adding additional residual blocks to the backbone. Results are given in Figure 3. We omit the error bars for figures where the standard deviations were negligible relative to the means.

Table 1: Performance comparison of our method with other methods for *HR estimation*

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑
Supervised									
FCN	15.13±0.50	21.63±0.48	52.09±5.43	16.57±0.91	26.20±0.60	55.98±0.78	12.45±0.12	18.35±0.24	56.98±0.78
ResNet	7.08±0.20	13.60±0.38	79.60±1.10	9.90±1.47	16.67±1.60	67.58±2.98	5.50±0.05	10.84±0.03	82.10±0.06
Self-Supervised									
SimCLR	12.42±0.05	20.96±0.30	73.62±0.52	16.41±0.22	22.62±0.39	52.16±1.12	16.88±0.19	22.64±0.22	56.37±0.21
BYOL	18.71±0.93	25.01±1.50	69.82±4.36	19.44±0.57	26.66±0.90	46.74±5.02	15.59±0.38	21.04±0.36	57.11±0.06
VICReg	13.17±0.82	20.38±1.27	73.65±0.02	16.78±0.47	23.10±0.75	54.10±1.26	15.70±0.15	21.83±0.18	55.32±0.62
Barlow Twins	13.22±0.34	20.42±0.88	67.51±2.01	22.08±0.85	29.35±0.56	35.65±3.40	11.87±0.57	19.20±0.29	62.20±0.40
CLIP	10.31±0.35	17.10±0.30	76.00±1.35	16.73±1.08	26.39±0.44	41.59±2.80	13.20±0.18	20.88±0.22	50.42±1.49
TS-TCC	11.56±0.41	18.04±0.66	78.38±1.41	16.52±0.34	24.86±0.59	44.93±3.40	10.23±0.01	18.19±0.04	62.77±0.04
SimMTM	13.20±0.11	18.27±0.22	73.78±1.10	17.03±0.63	25.18±0.98	51.33±2.62	13.61±0.05	20.12±0.07	55.47±0.09
TF-C	12.10±0.15	20.12±0.37	66.01±1.14	14.12±0.36	22.86±0.44	52.74±1.40	16.15±0.43	23.47±0.13	50.12±1.45
TS2Vec	9.80±0.49	16.64±0.55	75.30±1.10	24.57±0.30	24.83±0.16	50.40±2.31	12.65±0.32	20.04±0.33	58.17±0.53
Ours	**8.84**±0.50	**14.37**±0.95	**82.67**±1.30	**14.06**±1.09	**21.48**±2.01	**54.88**±1.89	**9.13**±0.20	**16.92**±0.56	**63.72**±0.06

Table 2: Performance comparison of our method with other techniques for *Activity* and *Step*

Method	HHAR			USC			Clemson		
	Acc ↑	W-F1 ↑	F1 ↑	Acc ↑	W-F1 ↑	F1 ↑	MAPE ↓	MAE ↓	RMSE ↓
Supervised									
FCN	74.21±1.56	72.88±2.06	71.58±1.81	48.87±0.74	46.02±0.95	45.33±0.82	5.02±0.26	2.86±0.15	4.05±0.13
ResNet	69.85±2.32	68.61±2.81	67.29±2.52	52.17±1.22	49.38±0.84	48.01±1.22	6.55±2.37	3.78±1.44	5.04±1.43
Self-Supervised									
SimCLR	40.55±0.62	39.21±0.64	39.41±0.66	29.16±0.69	29.02±0.67	28.99±0.79	8.70±0.22	4.36±0.13	6.30±0.24
BYOL	49.64±2.48	48.63±2.75	48.02±2.59	28.40±1.23	28.23±1.42	28.23±0.96	9.35±0.19	4.72±0.12	6.79±0.24
VICReg	38.05±3.01	37.12±2.66	37.38±3.02	23.75±1.00	23.16±1.03	22.92±1.21	10.87±0.61	5.47±0.35	7.78±0.14
Barlow Twins	38.97±0.65	37.75±1.00	38.21±1.12	27.24±0.19	26.84±0.20	26.25±0.77	9.89±0.35	4.95±0.15	7.03±0.21
CLIP	43.78±0.89	42.53±0.90	43.07±0.98	25.55±0.63	25.78±1.25	25.17±0.75	8.52±0.46	4.26±0.23	6.73±0.63
TS-TCC	68.56±1.19	66.90±1.22	68.10±1.30	33.61±0.72	33.11±1.09	33.91±0.79	5.61±0.15	2.70±0.06	4.69±0.38
SimMTM	44.78±0.62	42.48±0.37	43.60±0.62	22.34±0.28	25.68±0.41	29.72±1.78	8.77±0.18	4.61±0.32	6.90±0.18
TF-C	31.13±0.42	30.57±0.40	31.00±0.31	30.78±0.39	28.16±0.23	30.82±1.41	12.47±0.72	6.31±0.37	7.93±0.30
TS2Vec	67.13±0.11	65.56±0.21	64.13±0.21	35.40±0.96	32.17±1.26	35.47±1.42	5.92±0.93	3.01±0.28	5.02±0.42
Ours	**70.67**±0.06	**67.74**±0.29	**68.79**±0.25	**52.21**±1.09	**48.64**±1.52	**48.22**±1.11	**5.16**±0.44	**2.50**±0.13	**4.65**±0.45

Table 3: Performance comparison of our method with other techniques for *CVD* and *Sleep*

Method	Chapman			CPSC			Sleep		
	Acc ↑	AUC ↑	F1 ↑	Acc ↑	AUC ↑	F1 ↑	Acc ↑	W-F1 ↑	Kappa ↑
Supervised									
FCN	84.63±2.13	95.40±0.57	82.41±2.40	63.64±1.12	91.30±0.02	60.43±1.04	71.98±0.86	63.33±0.84	62.01±1.30
ResNet	93.16±0.41	98.59±0.05	92.02±0.42	75.21±1.73	95.02±0.03	71.70±1.90	76.94±0.97	67.52±1.95	69.14±0.61
Self-Supervised									
SimCLR	75.28±0.57	93.55±0.25	74.04±0.50	50.10±0.41	**87.20**±0.07	50.10±0.24	72.45±2.32	58.93±1.59	59.47±3.20
BYOL	76.08±0.40	93.54±0.18	74.80±0.45	51.90±0.30	87.05±0.22	50.89±0.38	70.77±0.27	58.23±0.55	55.90±1.20
VICReg	70.10±1.90	89.35±0.93	67.84±1.79	46.21±1.29	84.70±0.50	42.51±0.96	68.72±1.03	57.24±1.04	57.13±1.42
Barlow Twins	72.43±1.45	91.17±0.60	70.42±1.53	48.67±0.51	85.78±0.19	44.57±0.53	70.10±0.62	57.72±0.81	57.88±0.82
CLIP	82.98±0.96	95.15±0.42	81.00±1.03	50.01±0.89	86.40±0.32	47.99±0.89	73.16±0.81	62.06±0.91	63.75±1.23
TS-TCC	73.50±0.55	90.65±0.07	71.10±0.57	51.59±1.22	86.32±0.16	50.27±1.32	62.80±1.13	52.43±1.05	48.98±1.68
SimMTM	84.29±1.29	95.87±0.18	83.31±1.25	51.70±0.23	87.08±0.21	50.62±0.55	74.69±1.84	63.53±1.21	65.31±2.76
TF-C	85.84±0.39	96.10±0.10	84.71±0.40	47.86±0.69	86.27±0.05	45.42±0.66	64.50±1.80	56.77±2.21	52.61±2.41
TS2Vec	78.87±1.03	90.23±0.24	81.32±0.47	48.73±0.85	85.49±0.37	46.57±1.10	65.71±1.06	55.32±1.77	56.81±1.90
Ours	**87.21**±0.80	**96.50**±0.21	**85.30**±0.98	**52.10**±0.90	87.11±0.40	**51.26**±1.18	**77.30**±1.04	**68.05**±0.86	**69.16**±1.32

As shown in the results, simply increasing model size does not close the performance gap. In fact, larger models often exhibit stable or decreased performance, particularly in heterogeneous low-data regimes, likely due to overfitting even with strong augmentations. In contrast, our method consistently outperforms them across all tasks, achieving 10–15% higher performance with approximately half the number of parameters.

One observation from this comparison is that TF-C performs worse than others for some datasets, despite using two encoders, one for the time and one for the Fourier domain. We hypothesize that this performance drop may stem from the strong augmentations applied in TF-C framework, which changes the magnitude of frequency components and degrade representations for noisy signals.

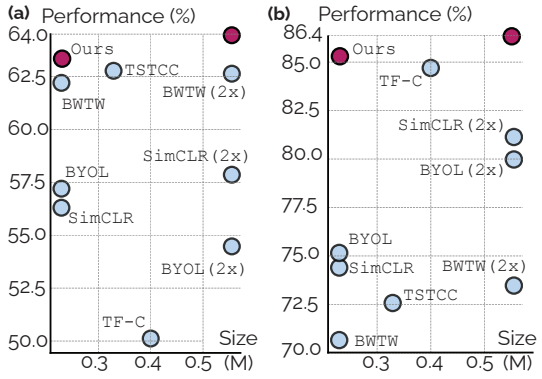


Figure 3: Performance of our method on DaLiA **(a)** using correlation (ρ) and on Chapman **(b)** using F1 score, compared to other self-supervised learning techniques. Barlow Twins is abbreviated as (BWTW), and backbone sizes are shown in millions of parameters. The red circle in each plot denotes our method, which achieves higher performance with fewer parameters.

For the second question, we generated a third view using a task-specific random augmentation and applied the instance discrimination loss across all views, following the same setup as our method. Figure 4 presents the results, including comparisons with larger backbone for prior techniques.

These results show that adding a third view improves performance for prior methods, but they still fall short of ours by about 10%. Notably, introducing a third view increases data diversity for these

methods, while in our case, using a third transformation does not. This empirical evidence highlights the data-hungry nature of prior SSL approaches and the efficiency of our method further.

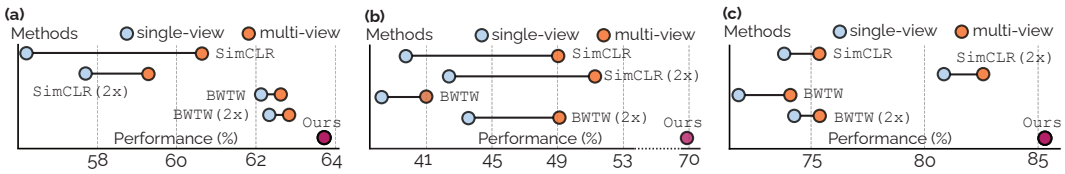


Figure 4: Performance comparison across datasets, **(a)** DaLiA (ρ), **(b)** HHAR (Acc), and **(c)** Chapman (F1), when adding a third view for instance discrimination similar to our method. While prior methods benefit from the additional view, their performance still lags behind our approach by a large margin.

4.1 Ablation Experiments

Our method consists of multiple components, so we conduct many ablation studies to assess the contribution of each. First, we evaluate the impact of each data transformation by selectively removing them individually. We exclude the orthonormal basis transformation (w/o OB) and retain only the overcomplete frame, then exclude the overcomplete frame transformation (w/o OF) and retain only the orthonormal basis in the method. We present the results across tasks in Tables 4, 5, and 6.

Table 4: Ablation on proposed method in *PPG* datasets for HR estimation

Method	IEEE SPC12			IEEE SPC22			DaLiA$_{PPG}$		
	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑
Ours	**8.84**±0.50	**14.37**±0.95	**82.67**±1.30	**14.06**±1.09	**21.48**±2.01	**54.88**±1.89	9.13±0.20	16.92±0.56	63.72±0.06
w/o OB	11.23 (+2.39)	17.26 (+2.89)	75.99 (-6.68)	15.88 (+1.82)	24.82 (+3.34)	53.96 (-0.92)	**8.87** (-0.26)	**16.13** (-0.79)	**65.34** (+1.62)
w/o OF	9.80 (+0.96)	15.18 (+0.81)	81.19 (-1.48)	15.33 (+1.27)	24.29 (+2.81)	53.05 (-1.83)	9.60 (+0.47)	17.31 (+0.39)	63.06 (-0.66)
w/o $\Phi_{\boldsymbol{h}}^{t\to d}$	9.62 (+0.78)	15.46 (+1.09)	80.21 (-2.46)	14.16 (+0.10)	22.46 (+0.98)	56.23 (+2.35)	8.90 (-0.23)	16.27 (-0.65)	64.65 (+0.93)

These results show that the best performance is mostly achieved when both transformations are used together, though the degree of performance change varies across tasks. One interesting result from our ablation study is that using only the Gabor wavelet transform sometimes outperforms the combination with Fourier transformation. Specifically, we observe that the Gabor wavelet is more effective for tasks involving sudden signal changes (e.g., abnormal heart rhythms, neural activity during sleep), while the Fourier transform better captures global structures such as periodic patterns in heart rate or step counting from inertial measurements. This empirical evidence supports our motivation for using complementary, principled transformations to jointly capture diverse signal characteristics.

Second, we retain both transformations and their NT-Xent loss but exclude the latent space mappers $\Phi_{\boldsymbol{h}}^{t\to d}$ during inference, performing linear probing solely on the original latent space $\boldsymbol{h}^{(t)}$. Results are reported in the same tables with the first ablation experiment under "w/o $\Phi_{\boldsymbol{h}}^{t\to d}$". As the results indicate, removing the mappers from our method degrades performance compared to the best case.

We conducted additional ablations to evaluate the performance of latent space mappers $\Phi_{\boldsymbol{h}}^{t\to d}$ and to assess the impact of embedding mappers $\Phi_{\boldsymbol{z}}^{t\to d}$ on the performance. Results are given in Appendix D.

4.2 Discussion of results

Do we need data augmentations? Data augmentations are widely viewed as essential for learning representations from unlabeled data [1, 53]. A celebrated theory, InfoMin [27], argues that effective augmentations reduce mutual information between views while retaining task-relevant features. However, our method does not rely on asymmetric views but instead uses unitary transformations that preserve all information. Our results suggest that asymmetric views are not essential for SSL.

A recent study [12] highlights the importance of strong augmentations in instance discrimination by showing that augmentations can cause different intra-class samples to align when their augmented views overlap (i.e., two different cars appear similar when both are cropped to show only the wheels).

Table 5: Ablation on proposed method in *IMU* datasets for Activity and Step

Method	HHAR			USC			Clemson		
	Acc ↑	W-F1 ↑	F1 ↑	Acc ↑	W-F1 ↑	F1 ↑	MAPE ↓	MAE ↓	RMSE ↓
Ours	**70.67**±0.06	**67.74**±0.29	**68.79**±0.25	**52.21**±1.09	**48.64**±1.52	**48.22**±1.11	**5.16**±0.44	**2.50**±0.23	**4.65**±0.50
w/o OB	67.45 (-3.22)	65.10 (-2.64)	86.50 (-2.09)	44.70 (-7.51)	41.20 (-7.44)	41.95 (-6.27)	7.50 (+2.34)	3.65 (+1.15)	5.79 (+1.14)
w/o OF	69.74 (-0.93)	67.62 (-1.12)	68.01 (-0.78)	49.06 (-3.15)	45.17 (-3.47)	45.61 (-2.61)	5.59 (+0.43)	2.72 (+0.22)	5.07 (+0.42)
w/o $\Phi_{\boldsymbol{h}}^{t\to d}$	67.11 (-3.56)	66.24 (-1.50)	67.05 (-1.74)	51.26 (-0.95)	47.81 (-0.83)	47.70 (-0.52)	5.20 (+0.04)	2.45 (-0.05)	4.53 (-0.12)

Table 6: Ablation on proposed method in *ECG* and *EEG* datasets for CVD and Sleep

Method	Chapman			CPSC			Sleep		
	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	Kappa (κ) ↑
Ours	**87.21**±0.80	96.50±0.21	85.30±0.98	52.10±0.90	87.11±0.40	51.26±1.18	77.30±1.04	68.05±0.86	69.16±1.32
w/o OB	86.79 (-0.42)	**96.51** (+0.01)	**85.90** (+0.60)	**58.81** (+6.71)	**89.35** (+2.24)	**55.91** (+ 4.65)	**81.15** (+3.85)	**70.73** (+2.68)	**74.37** (+5.21)
w/o OF	81.82 (-5.39)	94.15 (-2.35)	79.26 (-6.04)	45.83 (-6.27)	84.47 (-2.64)	44.08 (-7.18)	73.91 (-3.39)	63.98 (-4.07)	64.75 (-4.41)
w/o $\Phi_{\boldsymbol{h}}^{t\to d}$	84.98 (-2.23)	95.70 (-0.80)	82.69 (-2.61)	51.44 (-0.66)	86.59 (-0.52)	49.63 (-1.63)	79.30 (+2.00)	68.40 (+0.35)	71.67 (-2.51)

However, this explanation does not extend to our method. Since our transformations are unitary, the views preserve the structure of the original sample without introducing overlaps across instances. Our findings show that instance-discrimination–based self-supervised learning can succeed without relying on hand-crafted strong augmentations for temporal signals. This opens the door for future work to test whether the same holds in other modalities, such as images and audio.

Implicit bias An interesting finding from our experiments is that, although all encoders are trained with the same loss at the same time and without stop-gradient operations (commonly used to prevent collapse), the geometries of the learned representation differ significantly across encoders. We attribute this to the applied data transformations, which introduce strong implicit biases that shape each latent space in ways that emphasize different characteristics, such as global or local features.

5 Limitations

While our work demonstrates that instance discrimination-based SSL can be effective without aggressive data augmentations, we do not provide a theoretical explanation for the observed performance gains. We hypothesize that the representational improvements arise from implicit biases of the transformations, though this remains unverified by formal theoretical analysis.

In terms of scope, our experiments primarily focused on classification tasks, as this setting involves a wide variety of augmentations, including task-specific ones [5], making it a natural testbed for our augmentation-free approach. Nevertheless, since our method leverages both global (FFT) and local (Gabor) representations for representation learning, it also holds potential for forecasting tasks where capturing both long-range and localized temporal patterns is critical. Exploring this direction in both self-supervised and supervised paradigms remains an important avenue for future work.

Finally, while our method applies Fourier and wavelet transformations, it is worth noting that although the Fourier transform is computationally efficient [54], computing wavelet coefficients is expensive. In our experiments, we mitigated this by caching the coefficients after a one-time computation. We quantitatively evaluate the computational overhead of our method and other techniques in Appendix F, showing that our approach maintains competitive runtime compared to state-of-the-art works.

6 Related Work

Data augmentation in SSL Learning representations through data transformations dates back to early self-supervised learning methods, which introduced pretext tasks that reformulated the problem into a supervised one, such as predicting image rotations [55] or spatial contexts [56, 57]. More recently, data augmentations have become a central component of SSL, with stronger transformations applied to boost sample diversity [1, 12, 44]. However, learning representations with strong augmentations introduce new challenges [22, 58]. Mainly, strong augmentations can alter the label of a sample [59], leading to model collapse [27], and may suppress informative features [22]. This often causes models to rely on a subset of features aligned with augmentation-induced invariances, potentially ignoring others that are critical for downstream performance [19, 20, 22]. Motivated by these, we propose a method that replaces data augmentations with principled, well-understood transformations. Instead of relying on task-specific, hand-tuned augmentations, our approach performs instance discrimination using views generated from principled transformations.

Mapping between latent spaces Mapping between different representations is a growing area in deep learning as it enables the use of pre or co-trained models across domains [60, 61]. For instance, authors in [62] proposed a zero-shot communication method between latent spaces by projecting them into a shared relative space, constructed from pairwise distances between anchor points. However, this approach relies on anchor points to capture the structure of each latent space. More recent work focuses on directly translating between latent spaces using affine or orthogonal transformations [63]. These methods are inspired by Procrustes analysis for latent space alignment [64, 65], which has been applied in language processing [66, 67]. Yet, orthogonal transformations preserve inner products and therefore cannot capture the full representational differences learned by distinct encoders—an aspect central to our approach. Instead, our method estimates the geometry of the target latent space, shaped by domain-specific transformation biases, and enables its use in downstream tasks.

Multiview contrastive learning Learning representations with instance discrimination using multiple views without strong augmentations was proposed early in SSL [68–70]. Earlier methods [68] used different channels (e.g., L and ab from an RGB image) as views, which act as implicit augmentations and may bias the model toward certain features. In contrast, our method uses orthonormal and overcomplete transformations that are unitary or redundant by design, avoiding feature selection bias. Moreover, unlike prior work, we introduce lightweight mappers that learn the geometry of each representation space to better align them during inference to enable effective linear probing.

Implicit bias from frequency Prior work has leveraged frequency information for representation learning in temporal sequences [47, 71, 9]. Some approaches, in line with ours, use the NT-Xent loss to maximize agreement between time-domain inputs and their frequency-transformed counterparts using the Fourier transform [47] or spectrograms [72, 73]. However, our method differs in two key ways. First, these methods rely on additional encoders for each transformed view [47, 72] and still require task-specific data augmentations. In contrast, we replace augmentations entirely with principled transformations that generalize across datasets without task-specific tuning. Second, previous work typically focuses on either global frequency features (via Fourier transform) [47, 74] or localized frequency content (via spectrograms) [72, 73]. Our method integrates both by jointly using orthonormal and overcomplete representations while remaining more efficient using lightweight latent space mappers instead of using separate modality-specific encoders.

7 Conclusion

In conclusion, we have shown that principled geometric transformations in the form of orthonormal bases and overcomplete frames are effective for self-supervised representation learning. By generating views through unitary and frame-based projections, our method uses complementary manifolds without perturbing or enlarging the data. Crucially, our approach achieves up to 15–20% performance gains across nine datasets in five tasks, without relying on larger backbone architectures or handcrafted, domain-specific augmentations. Our results underscore the importance of exploiting intrinsic geometric biases in data representations, opening a new avenue for SSL methods that prioritize mathematical structure over empirical trial and error. We believe this work paves the way toward more generalizable, augmentation-free self-supervision across a wide range of domains.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[2] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[5] Berken Utku Demirel and Christian Holz. Finding order in chaos: A novel data augmentation method for time series in contrastive learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[6] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2021.

[7] Soyoung Yoon, Gyuwan Kim, and Kyumin Park. Ssmix: Saliency-based span mixup for text classification. *ArXiv*, abs/2106.08062, 2021.

[8] Yi Sui, Tongzi Wu, Jesse C. Cresswell, Ga Wu, George Stein, Xiao Shi Huang, Xiaochen Zhang, and Maksims Volkovs. Self-supervised representation learning from random data projectors. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] Berken Utku Demirel and Christian Holz. An unsupervised approach for periodic source detection in time series. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[10] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

[11] Vlad Sobal, Mark Ibrahim, Randall Balestriero, Vivien Cabannes, Diane Bouchacourt, Pietro Astolfi, Kyunghyun Cho, and Yann LeCun. $\mathbb{X}$-sample contrastive loss: Improving contrastive learning with sample similarity graphs. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[13] Yifei Wang, Qi Zhang, Tianqi Du, Jiansheng Yang, Zhouchen Lin, and Yisen Wang. A message passing perspective on learning dynamics of contrastive learning. In *International Conference on Learning Representations*, 2023.

[14] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2022.

[15] Lu Han, Han-Jia Ye, and De-Chuan Zhan. Augmentation component analysis: Modeling similarity via the augmentation overlaps. In *The Eleventh International Conference on Learning Representations*, 2023.

[16] Advait Parulekar, Liam Collins, Karthikeyan Shanmugam, Aryan Mokhtari, and Sanjay Shakkottai. Infonce loss provably learns cluster-preserving representations. In *Annual Conference Computational Learning Theory*, 2023.

[17] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations*, 2021.

[18] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[19] Joshua David Robinson, Li Sun, Ke Yu, kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[20] Zihu Wang, Yu Wang, Zhuotong Chen, Hanbin Hu, and Peng Li. Contrastive learning with consistent representations. *Transactions on Machine Learning Research*, 2024.

[21] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[22] Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? On the role of simplicity bias in class collapse and feature suppression. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 38938–38970. PMLR, 2023.

[23] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. Exploring contrastive learning in human activity recognition for healthcare. *ArXiv*, abs/2011.11542, 2020.

[24] Leon Cohen. *Time-frequency analysis: theory and applications*. Prentice-Hall, Inc., USA, 1995.

[25] F. Hlawatsch and G.F. Boudreaux-Bartels. Linear and quadratic time-frequency signal representations. *IEEE Signal Processing Magazine*, 9(2):21–67, 1992.

[26] Stéphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., USA, 3rd edition, 2008.

[27] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[28] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[29] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[30] Kimia Hamidieh, Haoran Zhang, Swami Sankaranarayanan, and Marzyeh Ghassemi. Views can be deceiving: Improved SSL through feature space augmentation. In *The Twelfth International Conference on Learning Representations*, 2024.

[31] AL Goldberger. Is the normal heartbeat chaotic or homeostatic? *Physiology*, 6(2):87–91, 1991. PMID: 11537649.

[32] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[33] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

[34] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Zhilin Zhang, Zhouyue Pi, and Benyuan Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, 2015.

[36] Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19, 2019.

[37] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. SenSys, 2015.

[38] Mi Zhang and Alexander A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. Association for Computing Machinery, 2012.

[39] Eddie Y. K. Ng, Feifei Liu, Chengyu Liu, Lina Zhao, X. Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, and Jianqing Li. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 2018.

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

[41] Erick A Perez Alday, Annie Gu, Amit J Shah, Chad Robichaux, An-Kwok Ian Wong, Chengyu Liu, Feifei Liu, Ali Bahrami Rad, Andoni Elola, Salman Seyedi, Qiao Li, Ashish Sharma, Gari D Clifford, and Matthew A Reyna. Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020. *Physiological Measurement*, 41(12):124003, dec 2020.

[42] Ryan Mattfeld, Elliot Jesch, and Adam Hoover. A new dataset for evaluating pedometer performance. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 865–869, 2017.

[43] 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. *Circulation*, 101(23):e215–e220, 2000.

[44] 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. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.

[45] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[47] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *Proceedings of Neural Information Processing Systems, NeurIPS*, 2022.

[48] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. SimMTM: A simple pre-training framework for masked time-series modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[49] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, Jun. 2022.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[51] Shenda Hong, Yanbo Xu, Alind Khare, Satria Priambada, Kevin Maher, Alaa Aljiffry, Jimeng Sun, and Alexey Tumanov. Holmes: Health online model ensemble serving for deep learning models in intensive care units. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1614–1624, 2020.

[52] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[53] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):857–876, 2023.

[54] An algorithm for the machine calculation of complex fourier series. In *Papers on Digital Signal Processing*. The MIT Press, 11 1969.

[55] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[56] Carl Doersch, Abhinav Kumar Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[57] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544. IEEE Computer Society, 2016.

[58] Junbo Zhang and Kaisheng Ma. Rethinking the Augmentation Module in Contrastive Learning: Learning Hierarchical Augmentation Invariance with Expanded Views . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[59] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5549–5560, 2023.

[60] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[61] Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[62] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023.

[63] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[64] Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[65] Chang Wang and Sridhar Mahadevan. Manifold alignment without correspondence. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, 2009.

[66] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[67] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*, 2017.

[68] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 776–794, Cham, 2020. Springer International Publishing.

[69] Vladan Stojnic and Vladimir Risojevic. Self-supervised learning of remote sensing scene representations using contrastive multiview coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1182–1191, June 2021.

[70] Yijie Lin, Yuanbiao Gou, Xiaotian Liu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. Dual contrastive prediction for incomplete multi-view representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4447–4461, 2023.

[71] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022.

[72] Luyu Wang and Aäron van den Oord. Multi-format contrastive learning of audio representations. *ArXiv*, abs/2103.06508, 2021.

[73] Luyu Wang, Pauline Luc, Adrià Recasens, Jean-Baptiste Alayrac, and Aäron van den Oord. Multimodal self-supervised learning of general audio representations. *ArXiv*, abs/2104.12807, 2021.

[74] Zhen Liu, Qianli Ma, Peitian Ma, and Linghao Wang. Temporal-frequency co-training for time series semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 2023.

[75] Peter G. Casazza and Gitta Kutyniok, editors. *Finite Frames: Theory and Applications*. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, 2013.

[76] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[77] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, October 2016.

[78] A. N. Gorban and I. Y. Tyukin. Blessing of dimensionality: mathematical foundations of the statistical physics of data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2118):20170237, 2018.

[79] Isaac Chavel. *Eigenvalues in Riemannian Geometry*. Academic Press, November 1984. Google-Books-ID: 0v1VfTWuKGgC.

[80] Isaac Chavel. *Isoperimetric Inequalities: Differential Geometric and Analytic Perspectives*. Cambridge University Press, July 2001. Google-Books-ID: LtPaxe18ukoC.

[81] Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. CBramod: A criss-cross brain foundation model for EEG decoding. In *The Thirteenth International Conference on Learning Representations*, 2025.

[82] Jiexi Liu and Songcan Chen. Timesurl: self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024.

[83] Leandro Giacomini Rocha, Dwaipayan Biswas, Bram-Ernst Verhoef, Sergio Bampi, Chris Van Hoof, Mario Konijnenburg, Marian Verhelst, and Nick Van Helleputte. Binary cornet: Accelerator for hr estimation from wrist-ppg. *IEEE Transactions on Biomedical Circuits and Systems*, 2020.

[84] Dani Kiyasseh, Tingting Zhu, and David A. Clifton. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International Conference on Machine Learning*, 2020.

[85] Hangwei Qian, Tian Tian, and Chunyan Miao. What makes good contrastive learning on small-scale wearable-based tasks? In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3761–3771, New York, NY, USA, 2022. Association for Computing Machinery.

[86] Horace Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12(3):241, aug 2001.

[87] Dwaipayan Biswas, Luke Everson, Muqing Liu, Madhuri Panwar, Bram-Ernst Verhoef, Shrishail Patki, Chris H. Kim, Amit Acharyya, Chris Van Hoof, Mario Konijnenburg, and Nick Van Helleputte. Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment. *IEEE Transactions on Biomedical Circuits and Systems*, 2019.

[88] Valentin Bieri, Paul Streli, Berken Utku Demirel, and Christian Holz. Beliefppg: uncertainty-aware heart rate estimation from ppg signals via belief propagation. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI '23. JMLR.org, 2023.

[89] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, Lille, France, 07–09 Jul 2015. PMLR.

[90] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[91] Viswanath Sivakumar, Jeffrey Seely, Alan Du, Sean R Bittner, Adam Berenzweig, Anuoluwapo Bolarinwa, Alexandre Gramfort, and Michael I Mandel. emg2qwerty: A large dataset with baselines for touch typing using surface electromyography. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[92] A Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

[93] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, April 2019.

[94] Jaemyung Yu, Jaehyun Choi, Dong-Jae Lee, HyeongGwon Hong, and Junmo Kim. Self-supervised transformation learning for equivariant representations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[95] T. Anderson Keller, Xavier Suau, and Luca Zappella. Homomorphic self-supervised learning. *Transactions on Machine Learning Research*, 2023.

[96] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022.

[97] Berken Utku Demirel and Christian Holz. Shifting the paradigm: A diffeomorphism between time series data manifolds for achieving shift-invariancy in deep learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[98] Akira Hasegawa, Kazuyoshi Itoh, and Yoshiki Ichioka. Generalization of shift invariant neural networks: Image processing of corneal endothelium. *Neural Networks*, 9(2):345–356, 1996.

[99] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7372–7380, Jun. 2022.

[100] Haifeng Li, Jun Cao, Jiawei Zhu, Qinyao Luo, Silu He, and Xuying Wang. Augmentation-free graph contrastive learning of invariant-discriminative representations. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):11157–11167, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have provided the empirical results with extensive ablations to show the contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Our theoretical results are given in Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the implementation details in the main manuscript Section 3.3 and Appendix E.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: For double-blind review, we include our code in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided details in Appendix E.1 and E.2.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviation across three runs with different random seeds to reflect the variability and statistical reliability of our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the computer resources in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPS Code of Ethics, and no ethical concerns or violations were identified.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work proposes a general self-supervised learning framework for temporal data. It does not target a specific application area, and as such, it does not have direct or immediate societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not include any pretrained language models, image generators or scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have added the relevant references in the appropriate sections of the manuscript.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the source code accompanying our method as part of the supplementary material under a CC BY-NC-SA: Attribution-NonCommercial-ShareAlike - license. The code is documented and sufficient to reproduce the results presented in the paper. No new datasets or personally identifiable data are introduced.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No large language models (LLMs) were used in the development or implementation of the core methods in this research. Any LLM use was limited to minor writing or editing support and did not impact the scientific contributions of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Theoretical Analysis

Here, we present complete proofs of our theoretical study, starting with notations. We assume all the samples are absolutely summable, and finite.

A.1 Representations and Notations

A.1.1 Orthonormal Bases and Overcomplete Frames

An orthonormal basis in a Hilbert space provides a complete set of mutually orthogonal unit-norm vectors. The Fourier transform forms such a basis for square-integrable signals over finite intervals. For a discrete signal x of length L, the discrete Fourier transform (DFT) is defined as:

$$\mathcal{F}_{\mathrm{x}}(k) = \frac{1}{\sqrt{L}} \sum_{n=0}^{L-1} x(n)\, e^{-j\frac{2\pi}{L}kn},$$

where k indexes the discrete frequencies. In this normalized form, the Fourier basis satisfies:

$$\sum_{k} |\langle \mathrm{x}, e_k \rangle|^2 = \|\mathrm{x}\|^2,$$

making it a tight frame (also known as a Parseval frame [75]). A frame can be a generalization of a basis that allows for redundancy. A set $\{\phi_i\} \subset \mathcal{H}$ is a frame for Hilbert space $\mathcal{H}$ if:

$$A\|\mathrm{x}\|^2 \leq \sum_{i} |\langle \mathrm{x}, \phi_i \rangle|^2 \leq B\|\mathrm{x}\|^2,$$

for all $x \in \mathcal{H}$ and constants $0 < A \leq B < \infty$. Frames provide stability and flexibility, especially for non-stationary or noisy signals.

The Gabor wavelet transform is an example of such a redundant frame. It uses Gaussian-modulated sinusoids to achieve localized time-frequency decomposition, trading off time and frequency resolution optimally [76]. The discrete Gabor wavelet transform for a signal x is defined as in below.

$$\mathcal{W}_{\mathrm{x}}(a, b) = \frac{1}{\sqrt{a}} \sum_{n=0}^{L-1} \mathrm{x}\, \psi\left(\frac{n-b}{a}\right),$$

where a controls the scale and b controls the translation (time shift); we use 48 log-spaced scales and apply shifts at single time-step intervals. The detailed implementations for the wavelet calculations are given in Appendix E.3.3. The Gabor wavelet $\psi(t)$ is defined as:

$$\psi(t) = e^{-\frac{t^2}{2\sigma^2}}\, e^{j2\pi\xi t},$$

where σ controls the width of the Gaussian envelope and ξ is the center frequency. The Gabor wavelet captures localized oscillations and is well-suited for analyzing transient or non-stationary features in wide-range of signals. In our work, the Fourier transform provides a global view of signal frequency content, while the Gabor wavelet transform enables fine-grained, localized analysis. Combining these complementary perspectives improves the expressiveness of the learned representations.

The complete list of notations used throughout this manuscript is provided in Table A.1.3.

A.1.2 Manifold

In our method, we define each latent representation space as a manifold $\mathcal{M}$, similar to the manifold hypothesis [77, 78], which states that high-dimensional data often lies on low-dimensional structures

within the ambient space. Since we generate three different latent representations (from time, Fourier, and wavelet domains), we consider a collection of manifolds, each corresponding to a specific transformation. In our implementation, each manifold is assigned a fixed latent dimension of 128, resulting in a total latent dimension of 384 across the three manifolds. For a fair comparison, we set the latent dimension to 384 for baseline methods, ensuring the linear classifier has the same number of parameters during fine-tuning.

A.1.3 Notation List

Notation	Description
$\mathbf{x}$	Temporal sequence represented as a bold lowercase symbol
$\mathcal{F}_{\mathrm{x}}[k]$	Fourier transformation of the temporal sequence with k frequencies
$\mathcal{W}_{\mathrm{x}}(a,b)$	Gabor wavelet transformation of the temporal sequence
$f_{\mathrm{x}}(.)$	The encoder to obtain representations for the temporal sequence
$f_{\mathcal{F}}(.)$	The encoder to obtain representations for the Fourier transformed temporal sequence
$f_{\mathcal{W}}(.)$	The encoder to obtain representations for the wavelet transformed temporal sequence
$g_{\mathrm{x}}(.)$	The projector to obtain embeddings for the temporal sequence
$g_{\mathcal{F}}(.)$	The projector to obtain embeddings for the Fourier transform of the temporal sequence
$g_{\mathcal{W}}(.)$	The projector to obtain embeddings for the wavelet transformation of the temporal sequence
$\boldsymbol{h}^{(t)}$	The representations obtained from temporal sequence, i.e., $\boldsymbol{h}^{(t)} = f_{\mathrm{x}}(\mathrm{x})$
$\boldsymbol{h}^{(\mathcal{F})}$	The representations obtained from the Fourier transformation of the temporal sequence
$\boldsymbol{h}^{(\mathcal{W})}$	The representations obtained from the Gabor wavelet transformation of the temporal sequence
$\boldsymbol{z}^{(t)}$	The embeddings of the sequence obtained from the projected representations, i.e. $\boldsymbol{z}^{(t)} = g_t(\boldsymbol{h}^t)$
$\boldsymbol{z}^{(\mathcal{F})}$	The embeddings of the Fourier transformed temporal sequence
$\boldsymbol{z}^{(\mathcal{W})}$	The embeddings of the wavelet transformed temporal sequence
$\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}$	The representation mapping function from time to Fourier domain
$\Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}$	The representation mapping function from time to Wavelet domain
$\Phi_{\boldsymbol{z}}^{t\to\mathcal{F}}$	The embedding mapping function from time to Fourier domain
$\Phi_{\boldsymbol{z}}^{t\to\mathcal{W}}$	The embedding mapping function from time to Wavelet domain
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle$	The inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$\mathrm{sim}(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_j^{(d')})$	Cosine similarity between the embeddings
$\perp$	Perpendicular
τ	Temperature coefficient for NT-Xent loss
$\mathcal{M}$	Manifold notation
$\mathcal{L}$	A loss function, i.e., cross-entropy.

Table 7: Detailed list of notations used in this work

A.2 Proof for Proposition 2.2

Proposition A.1 (Angle Concentration vs. Pairwise Spread)**.** *Let* $\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \sim \mathrm{Unif}(S^{d-1})$*, where* $\boldsymbol{h}^{(\mathcal{F})} = f_{\mathcal{F}}(\mathcal{F}(\mathrm{x}))$*. Although individual samples across latent spaces tend toward orthogonality, the pairwise angular difference* Δ_{ij} *between distinct samples can span the full range up to* π*.*

$$\arccos\left(\langle \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \rangle\right) = \frac{\pi}{2}, \quad \textit{while} \quad \arccos(\langle \boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)} \rangle) - \arccos(\langle \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle) = \Delta_{ij} \leq \pi \quad (7)$$

Proof. The first part follows the Chernoff on the spherical cap for angle concentration, Let $\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(S^{d-1})$ and write $\xi := \langle \boldsymbol{h}^{(t)}, \boldsymbol{h}^{(\mathcal{F})} \rangle$.

$$\mathbb{E}[\xi] = 0, \qquad \mathrm{Var}[\xi] = \frac{1}{d}. \tag{8}$$

Lévy's concentration [79, 80] gives, for every $\varepsilon \in (0, 1)$,

$$\Pr\big(|\xi| > \varepsilon\big) \;\leq\; 2\exp\Big(-\tfrac{(d-2)\varepsilon^2}{2}\Big) \;\xrightarrow[d\to\infty]{}\; 0 \tag{9}$$

Hence $\xi \xrightarrow{p} 0$ and, by continuity of $\arccos$ at 0,

$$\arccos \xi \xrightarrow{p} \frac{\pi}{2}. \tag{10}$$

Let distinct indices $i \neq j$ while setting,

$$V := \boldsymbol{h}_i^{(t)\perp} \cap \boldsymbol{h}_j^{(t)\perp}, \qquad \dim V = d - 2 \geq 1 \;\; (d \geq 3). \tag{11}$$

Choose orthonormal $u, u_\perp \in V$. For any $\phi \in [0, \pi]$ define

$$\boldsymbol{h}_i^{(\mathcal{F})} := u, \qquad \boldsymbol{h}_j^{(\mathcal{F})} := \cos\phi\, u + \sin\phi\, u_\perp \tag{12}$$

Then

$$\langle \boldsymbol{h}_i^{(t)}, \boldsymbol{h}_i^{(\mathcal{F})} \rangle = \langle \boldsymbol{h}_j^{(t)}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle = 0, \qquad \langle \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle = \cos\phi, \tag{13}$$

Let

$$\theta_{ij}^{(t)} := \arccos\langle \boldsymbol{h}_i^{(t)}, \boldsymbol{h}_j^{(t)} \rangle, \qquad \theta_{ij}^{(\mathcal{F})} := \arccos\langle \boldsymbol{h}_i^{(\mathcal{F})}, \boldsymbol{h}_j^{(\mathcal{F})} \rangle = \phi. \tag{14}$$

Thus

$$\Delta_{ij} := \theta_{ij}^{(t)} - \theta_{ij}^{(\mathcal{F})} = \theta_{ij}^{(t)} - \phi, \tag{15}$$

and varying ϕ over $[0, \pi]$ makes Δ_{ij} sweep the full interval $[\theta_{ij}^{(t)} - \pi,\ \theta_{ij}^{(t)}] \subset [-\pi, \pi]$. In particular, $\phi = 0$ or π yields $|\Delta_{ij}| = \pi$. □

Therefore, even though the angles from same samples concentrate at $\pi/2$, pairwise discrepancies can reach any value up to the maximal π.

In practice, although $\boldsymbol{h}^{(t)}$ and $\boldsymbol{h}^{(\mathcal{F})}$ are coupled via the same loss, we conjecture that the high-dimensional geometry and symmetric repulsion of negatives make their joint distribution approximate to the independent uniforms on S^{d-1} [32]. This justifies modeling both latent spaces as samples from $\mathrm{Unif}(S^{d-1})$ when analyzing angular gaps between non-matched pairs.

A.3 Proof for Proposition 2.1

Proposition A.2. *Let f_d^* denote an optimal encoder under NT-Xent for domain $d \in \{t, \mathcal{F}, \mathcal{W}\}$. If for some unintended transformation W the encoder is invariant, i.e., $f_d^*(W\mathrm{x}) = f_d^*(\mathrm{x})$, then for any anchor sample x the NT-Xent loss across domains is lower bounded by the number of negatives.*

$$\ell\Big(\boldsymbol{z}_i^{(d)}, \boldsymbol{z}_j^{(d')}\Big) \;\geq\; \log(K+1) > 0,$$

where $K \geq 1$ is the number of negatives that become near-positives due to the invariance.

Proof. For each domain $d \in \{t, \mathcal{F}, \mathcal{W}\}$, let $f_d : \mathcal{X} \to \mathbb{R}^m$ be the encoder and g_d the projection head. We write the (unit-normalized) embedding as

$$\boldsymbol{z}^{(d)}(\mathrm{x}) \coloneqq \frac{g_d(f_d(\mathrm{x}))}{\|g_d(f_d(\mathrm{x}))\|_2} \in \mathbb{S}^{m-1}, \quad \text{and } \operatorname{sim}(u, v) \coloneqq u^\top v \in [-1, 1] \tag{16}$$

For an anchor x in domain d and its positive view in domain $d' \neq d$ (same underlying sample), the NT-Xent loss is

$$\ell_{d,d'}(\mathrm{x}) = -\log \frac{\exp\big(\operatorname{sim}(\boldsymbol{z}^{(d)}(\mathrm{x}), \boldsymbol{z}^{(d')}(\mathrm{x}))/\tau\big)}{\exp\big(\operatorname{sim}(\boldsymbol{z}^{(d)}(\mathrm{x}), \boldsymbol{z}^{(d')}(\mathrm{x}))/\tau\big) + \sum_{k \neq i} \exp\big(\operatorname{sim}(\boldsymbol{z}^{(d)}(\mathrm{x}), \boldsymbol{z}^{(d')}(\mathrm{x}_k))/\tau\big)}, \tag{17}$$

with temperature $\tau > 0$. The full instance-discrimination objective $\mathcal{L}_{\mathrm{ID}}$ (Equation 3) averages/sums $\ell_{d,d'}$ over all ordered domain pairs and batch samples.

Suppose there exists an unintended transformation W and a domain $d \in \{t, \mathcal{F}, \mathcal{W}\}$ such that

$$f_d^*(Wx) = f_d^*(\mathrm{x}) \tag{18}$$

In a batch, let $S \subset \{k \neq i\}$ be indices of *near-positive* negatives created by the invariance:

$$\operatorname{sim}\big(\boldsymbol{z}^{(d)}(\mathrm{x}),\, \boldsymbol{z}^{(d')}(\mathrm{x}_k)\big) \geq 1 - \delta \quad \text{for all } k \in S, \tag{19}$$

with $|S| = K \geq 1$ and some $\delta \in [0, 1)$. In the exact invariance case, $\delta = 0$.

Per-pair lower bound. Fix (d, d') and an anchor sample x. We can define the positive similarity between embeddings as $s_{\mathrm{pos}} \coloneqq \operatorname{sim}(\boldsymbol{z}^{(d)}, \boldsymbol{z}^{(d')})$. If we split the negative set into S (the K near-positives) and R (the rest). From (17) and (19), we can write the loss function as in Equation 20

$$\ell_{d,d'}(\mathrm{x}) = -\log \frac{e^{s_{\mathrm{pos}}/\tau}}{e^{s_{\mathrm{pos}}/\tau} + \sum_{k \in S} e^{\operatorname{sim}(\boldsymbol{z}^{(d)}(\mathrm{x}), \boldsymbol{z}^{(d')}(\mathrm{x}_k))/\tau} + \sum_{r \in R} e^{\operatorname{sim}(\cdot)/\tau}} \;\geq\; -\log \frac{e^{s_{\mathrm{pos}}/\tau}}{e^{s_{\mathrm{pos}}/\tau} + K\, e^{(1-\delta)/\tau}} \tag{20}$$

Factor $e^{s_{\mathrm{pos}}/\tau}$ from the denominator,

$$\ell_{d,d'}(\mathrm{x}) \geq -\log \frac{1}{1 + K\, e^{\frac{1-\delta-s_{\mathrm{pos}}}{\tau}}} = \log\Big(1 + K\, e^{\frac{1-\delta-s_{\mathrm{pos}}}{\tau}}\Big)$$

At optimum, $s_{\mathrm{pos}} = 1$ (or $s_{\mathrm{pos}} \geq 1 - \varepsilon$ in the approximate case). Since the RHS is decreasing in s_{pos}, the weakest bound (i.e., smallest lower bound that still holds) is obtained at $s_{\mathrm{pos}} = 1$, giving

$$\ell_{d,d'}(\mathrm{x}) \;\geq\; \log\Big(1 + K\, e^{-\delta/\tau}\Big) \tag{21}$$

In particular, for exact near-positives ($\delta = 0$), $\ell_{d,d'}(\mathrm{x}) \geq \log(K+1)$.

As is standard, we assume that at the NT-Xent optimum positive pairs align [31, 32], i.e., $\operatorname{sim}(\boldsymbol{z}^{(d)}(\mathrm{x}), \boldsymbol{z}^{(d')}(\mathrm{x})) = 1$. To extend this proof, we have also provided a case which also covers the approximate case. If $s_{\mathrm{pos}} \geq 1 - \varepsilon$ for some small $\varepsilon \geq 0$, then

$$\ell_{d,d'}(x) \;\geq\; \log\Big(1 + K\, e^{\frac{\varepsilon-\delta}{\tau}}\Big) \;\geq\; \log\Big(1 + K\, e^{-\delta/\tau}\Big),$$

because $\varepsilon \geq 0$. Thus, Equation 21 still holds. □

The bound in Equation 21 is tight when $s_{\text{pos}} = 1$ and exactly K negatives have similarity $1 - \delta$ while all others contribute negligibly, so the denominator equals $e^{1/\tau} + Ke^{(1-\delta)/\tau}$.

Extending to the full objective, recall that $\mathcal{L}_{\text{ID}}$ (Eq. 3) sums/averages $\ell_{d,d'}$ over all ordered domain pairs (d, d') with $d \neq d'$. For a fixed anchor x, if invariance induces $K_{d,d'}$ near-positive negatives with parameter $\delta_{d,d'}$ for pair (d, d'), then applying Equation 21 to each pair and summing gives

$$\sum_{d \neq d'} \ell_{d,d'}(\mathrm{x}) \;\geq\; \sum_{d \neq d'} \log\Big(1 + K_{d,d'}\, e^{-\delta_{d,d'}/\tau}\Big), \tag{22}$$

and by linearity of expectation,

$$\mathcal{L}_{\text{ID}} \;\geq\; \mathbb{E}_{\mathrm{x}}\left[\sum_{d \neq d'} \log\Big(1 + K_{d,d'}\, e^{-\delta_{d,d'}/\tau}\Big)\right] \tag{23}$$

Asymptotically, for fixed τ and bounded $\delta_{d,d'}$, each affected pair contributes $\Theta(\log(1 + K_{d,d'})) = \Omega(\log K_{d,d'})$. If a nonzero fraction p of a batch of size B yields near-positives per affected pair ($K_{d,d'} = \Theta(B)$), then $\ell_{d,d'}(\mathrm{x}) = \Omega(\log B)$ and consequently $\mathcal{L}_{\text{ID}} = \Omega\Big(\sum_{d \neq d'} \log B\Big)$ over those pairs. Thus any unintended invariance that produces even a single near-positive per ordered pair enforces a nontrivial lower bound; if collisions scale with batch size, the bound grows at least logarithmically in B.

Unintended invariances therefore inflate the NT-Xent denominator through near-positive negatives, yielding the lower bound in Equation 21. We complete the proof by showing the objective cannot be minimized without bound when such invariances hold.

B Algorithm

In this section, we present the pseudocode for our method during pre-training and inference. Algorithm 1 describes the training procedure, which takes a sample, x, and model components as inputs, and outputs the trained encoder and latent space mappers.

Algorithm 1 Pre-training algorithm for the proposed method

Input: x, $\mathcal{F}_{\mathrm{x}}$, $\mathcal{W}_{\mathrm{x}}$, and the required models, i.e., $f_{\mathrm{x}}(.)$.

Output: $f_{\mathrm{x}}(.)$, $\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}$, $\Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}$ ▷ The output of the pre-training is the single encoder with mappers

$\boldsymbol{h}^{(\mathrm{t})} = f_{\mathrm{x}}(\mathrm{x})$, $\boldsymbol{h}^{(\mathcal{F})} = f_{\mathcal{F}}(\mathcal{F}_{\mathrm{x}})$, $\mathbf{h}^{(\mathcal{W})} = f_{\mathcal{W}}(\mathcal{W}_{\mathrm{x}})$ ▷ Obtain representations for each input

$\boldsymbol{z}^{(t)} = g_{\mathrm{x}}(\boldsymbol{h}^{(t)})$, $\boldsymbol{z}^{(\mathcal{F})} = g_{\mathcal{F}}(\boldsymbol{h}^{(\mathcal{F})})$, $\boldsymbol{z}^{(\mathcal{W})} = g_{\mathcal{W}}(\boldsymbol{h}^{(\mathcal{W})})$ ▷ Obtain embeddings for each representation

$\mathcal{L}_{\mathrm{ID}} = \sum_{\substack{d,d'\in\{t,\mathcal{F},\mathcal{W}\}\\ d\neq d'}} \frac{1}{2N}\sum_{k=1}^{N}\Big[\ell\big(\boldsymbol{z}_{k-1}^{(d)}, \boldsymbol{z}_{k}^{(d')}\big) + \ell\big(\boldsymbol{z}_{k}^{(d)}, \boldsymbol{z}_{k-1}^{(d')}\big)\Big]$ ▷ Instance discrimination loss

$\boldsymbol{z}_{\mathrm{est}}^{(\mathcal{F})} = \Phi_{\boldsymbol{z}}^{t\to\mathcal{F}}(\boldsymbol{z}^{(t)})$, $\boldsymbol{z}_{\mathrm{est}}^{(\mathcal{W})} = \Phi_{\boldsymbol{z}}^{t\to\mathcal{W}}(\boldsymbol{z}^{(t)})$ ▷ Obtain the estimated embeddings for both transformations

$\mathcal{L}_{\mathrm{map}} = \frac{1}{N}\left\|\Phi_{\boldsymbol{z}}^{t\to\mathcal{F}}\left(\boldsymbol{z}^{(t)}\right) - \boldsymbol{z}^{(\mathcal{F})}\right\|_1 + \frac{1}{N}\left\|\Phi_{\boldsymbol{z}}^{t\to\mathcal{W}}\left(\boldsymbol{z}^{(t)}\right) - \boldsymbol{z}^{(\mathcal{W})}\right\|_1$ ▷ Mapping loss for embeddings

---

Freeze $\{f_{\mathrm{x}},\ f_{\mathcal{F}},\ f_{\mathcal{W}}\}$; **Omit**, $\{g_{\mathrm{x}},\ g_{\mathcal{W}},\ g_{\mathcal{W}},\ \Phi_{\boldsymbol{z}}^{t\to\mathcal{F}}, \Phi_{\boldsymbol{z}}^{t\to\mathcal{M}}\}$; **Train** $\{\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}, \Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}\}$

$\boldsymbol{h}^{(\mathrm{t})} = f_{\mathrm{x}}(\mathrm{x})$, $\boldsymbol{h}^{(\mathcal{F})} = f_{\mathcal{F}}(\mathcal{F}_{\mathrm{x}})$, $\mathbf{h}^{(\mathcal{W})} = f_{\mathcal{W}}(\mathcal{W}_{\mathrm{x}})$ ▷ Obtain representations using trained models

$\boldsymbol{h}_{\mathrm{est}}^{(\mathcal{F})} = \Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}(\boldsymbol{h}^{(t)})$, $\boldsymbol{h}_{\mathrm{est}}^{(\mathcal{W})} = \Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}(\boldsymbol{h}^{(t)})$ ▷ Obtain the estimated representations for transformations

$\mathcal{L}_{\mathrm{map}} = \frac{1}{N}\left\|\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}\left(\boldsymbol{h}^{(t)}\right) - \boldsymbol{h}^{(\mathcal{F})}\right\|_1 + \frac{1}{N}\left\|\Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}\left(\boldsymbol{h}^{(t)}\right) - \boldsymbol{h}^{(\mathcal{W})}\right\|_1$ ▷ Mapping loss for representations

Return: $f_{\mathrm{x}}(.)$, $\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}$, $\Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}$

Algorithm 2 outlines the inference process, using only the main encoder and lightweight mappers. After applying the mappers, we concatenate the resulting representations for linear probing. During linear probing, both the main encoder and the mappers are kept frozen.

Algorithm 2 The proposed method for inference

Input: x, $f_{\mathrm{x}}(.)$, $\Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}$, and $\Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}$.

Output: $\boldsymbol{h}$ ▷ The output of the pre-training is the representations

$\boldsymbol{h}^{(\mathrm{t})} = f_{\mathrm{x}}(\mathrm{x})$ ▷ Obtain representations for inputs

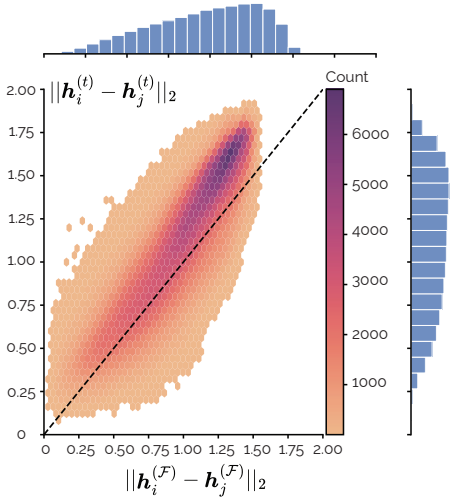
$\boldsymbol{h}_{\mathrm{est}}^{(\mathcal{F})} = \Phi_{\boldsymbol{h}}^{t\to\mathcal{F}}(\boldsymbol{h}^{(t)})$, $\boldsymbol{h}_{\mathrm{est}}^{(\mathcal{W})} = \Phi_{\boldsymbol{h}}^{t\to\mathcal{W}}(\boldsymbol{h}^{(t)})$ ▷ Obtain the estimated representations in other domains

$\boldsymbol{h} = \left[\boldsymbol{h}^{(t)}\ ;\ \boldsymbol{h}_{\mathrm{est}}^{(\mathcal{F})}\ ;\ \boldsymbol{h}_{\mathrm{est}}^{(\mathcal{W})}\right]$ ▷ Concatenate features from all domains

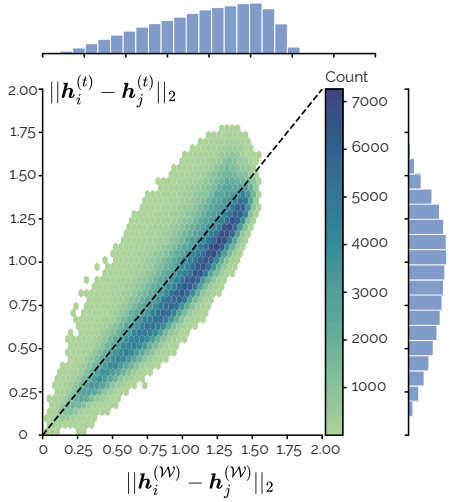
Return: $\boldsymbol{h}$

C Pairwise Distances

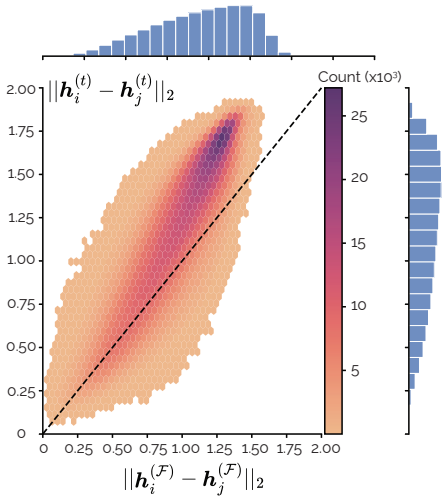
In addition to angular comparisons, we report the ℓ_2 distances between sample pairs in the time, Fourier, and wavelet domains. Figures 5, 6 and 7 visualize these results. If pairwise distances were preserved across latent spaces, all points would lie along the $y = x$ line—i.e., the distance between samples i and j in the time-domain latent space $\boldsymbol{h}^{(t)}$ would match that in the transformed domain $\boldsymbol{h}^{(d)}$, where $d \in \{\mathcal{F}, \mathcal{W}\}$.



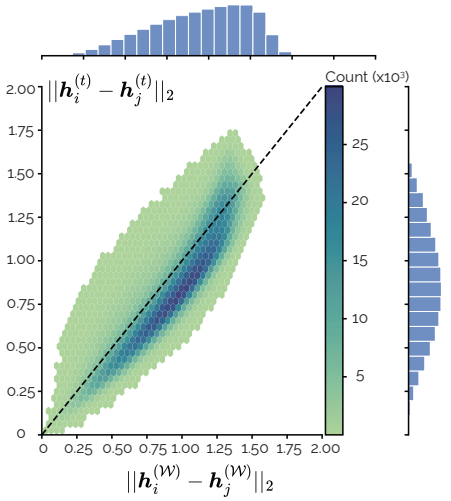
(a) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on IEEE SPC12



(b) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on IEEE SPC12



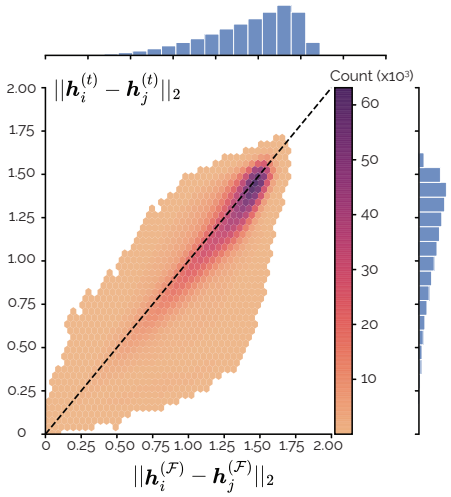
(c) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on IEEE SPC22



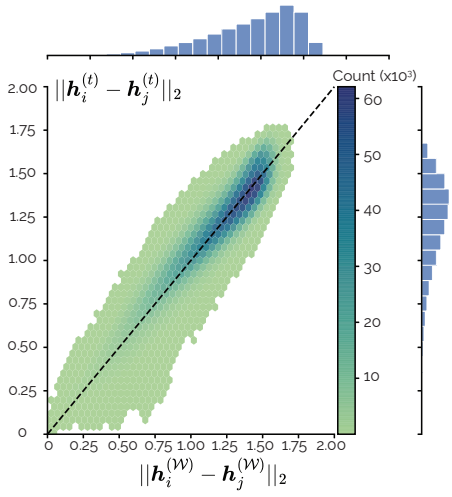
(d) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on IEEE SPC22

Figure 5: Pairwise ℓ_2 distance comparisons across domains and datasets for *heart rate* estimation.

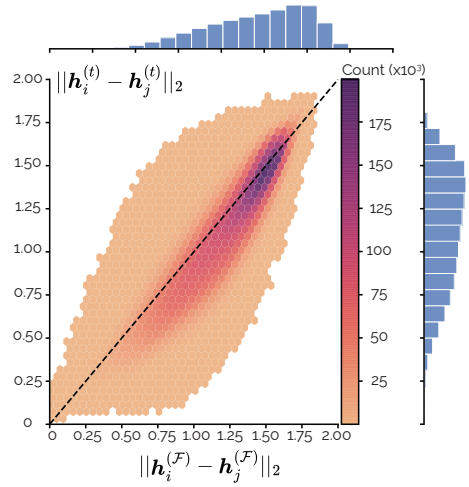
We conducted this investigation across tasks involving both single- and multi-channel temporal data. It is worth noting that while some datasets exhibit closer alignment between latent spaces, we observed consistent and non-negligible deviations across all tasks, indicating that latent space geometries differ across applications.



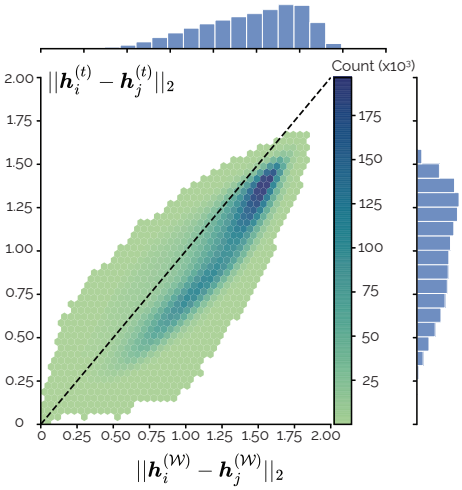
(a) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on HHAR



(b) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on HHAR

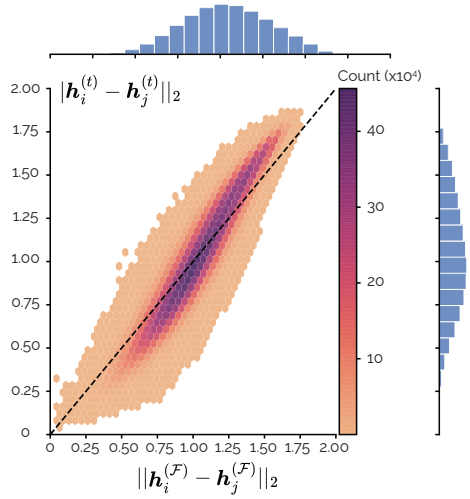


(c) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on USC

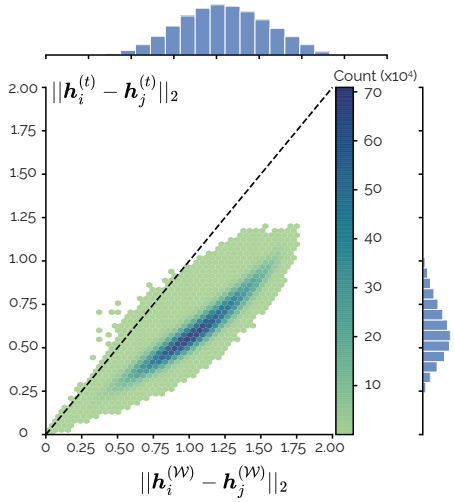


(d) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on USC

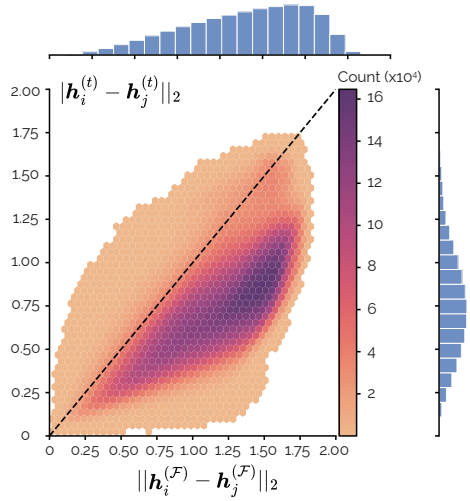
Figure 6: Pairwise ℓ_2 distance comparisons across domains and datasets for *activity* recognition.



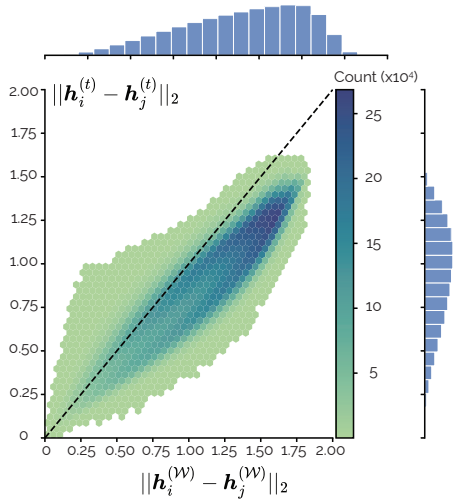
(a) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on CPSC



(b) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on CPSC



(c) Pairwise ℓ_2 distance comparison between time and Fourier domain latent spaces on Sleep



(d) Pairwise ℓ_2 distance comparison between time and wavelet domain latent spaces on Sleep

Figure 7: Pairwise ℓ_2 distance comparisons across domains and datasets for *cardiovascular disease* and *sleep* classification.

These figures reveal substantial deviations from $y = x$ line, indicating that distances between samples ($i \neq j$) in the latent space vary across domains. Importantly, these discrepancies are consistent despite all encoders being trained jointly with the same objective. These observations further support our motivation for leveraging multiple latent spaces to capture complementary structure in the data.

D Additional Experiments

D.1 Embedding mappers

We have employed embedding space mappers ($\Phi_{\boldsymbol{z}}^{t\to d}$) after projection layers to improve predictability across latent spaces and reduce estimation error. In this section, we have presented the ablation experiments regarding the performance when the embedding mappers are excluded (w/o $\Phi_{\boldsymbol{z}}^{t\to d}$) from our method. The results are given in Tables 8, 9 and 10.

Table 8: Further ablation on proposed method in *PPG* datasets for HR estimation

Method	IEEE SPC12			IEEE SPC22			DaLiA_{PPG}		
	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑
Ours	**8.84**±0.50	**14.37**±0.95	**82.67**±1.30	**14.06**±1.09	**21.48**±2.01	**54.88**±1.89	9.13±0.20	16.92±0.56	63.72±0.06
w/o $\Phi_{\boldsymbol{z}}^{t\to d}$	9.87 (+1.03)	15.85 (+1.48)	79.78 (-2.89)	15.17 (+1.11)	24.75 (+2.27)	53.19 (-1.69)	9.83 (+0.70)	17.68 (+0.76)	61.15 (-2.57)
Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	9.77 (+0.93)	16.27 (+1.90)	78.77 (-3.90)	15.61 (+1.55)	24.10 (+2.58)	54.10 (-0.78)	9.23 (+0.10)	16.98 (+0.06)	63.67 (-0.05)
Non. Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	10.11 (+1.27)	16.06 (+1.69)	78.97 (-3.70)	15.62 (+1.56)	24.65 (+3.17)	53.03 (-1.85)	8.92 (-0.21)	16.52 (-0.30)	64.29 (+0.57)

Table 9: Further ablation on proposed method in *IMU* datasets for Activity and Step

Method	HHAR			USC			Clemson		
	Acc ↑	W-F1 ↑	F1 ↑	Acc ↑	W-F1 ↑	F1 ↑	MAPE ↓	MAE ↓	RMSE ↓
Ours	70.67±0.06	67.74±0.29	68.79±0.25	52.21±1.09	48.64±1.52	48.22±1.11	**5.16**±0.44	**2.50**±0.23	**4.65**±0.50
w/o $\Phi_{\boldsymbol{z}}^{t\to d}$	67.17 (-3.50)	67.03 (-0.71)	67.01 (-1.78)	52.09 (-0.12)	49.31 (+0.67)	**49.25** (+1.03)	5.31 (+0.15)	2.57 (+0.07)	4.76 (+0.11)
Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	68.13 (-2.54)	68.23 (+0.49)	67.08 (-1.71)	51.13 (-1.08)	47.55 (-1.09)	47.54 (-0.68)	5.79 (+0.63)	2.55 (+0.05)	4.67 (+0.02)
Non. Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	**71.13** (+0.46)	**69.19** (+1.45)	**70.25** (+1.46)	**53.20** (+0.99)	48.27 (-0.37)	49.01 (+0.79)	6.40 (+1.24)	3.11 (+0.61)	5.46 (+0.81)

Table 10: Further ablation on proposed method in *ECG* and *EEG* datasets for CVD and Sleep

Method	Chapman			CPSC			Sleep		
	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	AUC ↑	Acc ↑	F1 ↑	Kappa (κ) ↑
Ours	87.21 ±0.80	96.50 ±0.21	85.30 ±0.98	52.10±0.90	87.01 ±1.10	51.26 ±1.18	77.30 ±1.04	68.05 ±0.86	69.16 ±1.32
w/o $\Phi_{\boldsymbol{z}}^{t\to d}$	86.43 (-0.78)	96.31 (-0.19)	84.40 (-0.90)	52.03 (-0.07)	87.39 (+0.38)	51.10 (-0.16)	77.98 (+0.68)	68.04 (-0.01)	70.23 (+1.11)
Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	86.35 (-0.86)	96.06 (-0.44)	84.26 (-1.04)	50.58 (-1.52)	86.55 (-0.46)	48.78 (-1.48)	77.83 (+0.53)	68.61 (+0.56)	70.02 (+0.86)
Non. Lin. $\Phi_{\boldsymbol{h}}^{t\to d}$	**90.91** (+3.70)	**97.96** (+1.46)	**89.76** (+4.46)	**56.30** (+4.20)	**88.09** (+1.08)	**53.44** (+2.18)	**79.93** (+2.63)	**70.27** (+2.22)	**73.10** (+3.94)

As can be seen in these tables, removing the embedding space mappers ($\Phi_{\boldsymbol{z}}^{t\to d}$) consistently leads to decreased performance across all datasets. In a few cases where the performance appears slightly better, the improvement remains within the range of standard deviation and is not statistically significant. These results support the role of embedding-level mapping in enhancing performance.

D.2 Latent space mappers

When using latent space mappers, we employed lightweight nonlinear convolutional networks (see Section E.3 for architectural details). Here, we report results from replacing the mapper with either a simple linear layer (Lin. $\Phi_{\boldsymbol{z}}^{t\to d}$) or a nonlinear two-layer multilayer perceptron (MLP $\Phi_{\boldsymbol{z}}^{t\to d}$). We present the results in Tables 8, 9 and 10.

As shown in the tables, replacing the convolutional mapper with a linear layer results in a noticeable performance drop, suggesting that the relationships between latent spaces are too complex for simple linear mappings. Although a non-linear MLP improves performance on some datasets, its two-layer structure significantly increases the parameter count. To ensure a fair comparison with prior work while avoiding added computational overhead, we use our lightweight convolutional architecture.

D.3 Using all encoders instead of mappers

We evaluate a variant of our method that directly uses the original representations from all encoders, rather than mapping the time-domain representations into other domains. This setup yields a 4–5% performance improvement over our default approach. However, it significantly increases the number of parameters at inference time, making it less efficient. These results suggest a promising direction for future work: leveraging principled transformations can substantially boost performance, though careful trade-offs with inference cost should also be considered.

D.4 Cross domain results

Cross-domain evaluations are commonly used to assess the generalization of self-supervised learning methods across datasets [47, 48]. Following this practice, we pretrain each model on a dataset and fine tune it on another from a different domain, following the setup in [48]. Both supervised models in our experiments (FCN and ResNet) are initialized randomly and trained. We present the result in Tables 11 and 12. Unlike linear probing, where only the linear classifier is trained with a limited data, fine-tuning updates the entire encoder. To ensure consistency, we use the same limited data size as in linear probing, avoiding settings that resemble supervised training with abundant labeled data.

Table 11: Performance comparison of methods for *Activity* and *Step* in cross domain settings

Method	HHAR			USC			Clemson		
	Acc ↑	W-F1 ↑	F1 ↑	Acc ↑	W-F1 ↑	F1 ↑	MAPE ↓	MAE ↓	RMSE ↓
Supervised									
FCN	74.21±1.56	72.88±2.06	71.58±1.81	48.87±0.74	46.02±0.95	45.33±0.82	5.02±0.26	2.86±0.15	4.05±0.13
ResNet	69.85±2.32	68.61±2.81	67.29±2.52	52.17±1.22	49.38±0.84	48.01±1.22	6.55±2.37	3.78±1.44	5.04±1.43
Self-Supervised									
SimCLR (in)	40.55±0.62	39.21±0.64	39.41±0.66	29.16±0.69	29.02±0.67	28.99±0.79	8.70±0.22	4.36±0.13	6.30±0.24
SimCLR (cross)	41.25±0.62	39.75±0.64	40.05±0.66	29.75±0.69	29.55±0.67	29.45±0.79	8.55±0.22	4.28±0.13	6.18±0.24
BYOL (in)	49.64±2.48	48.63±2.75	48.02±2.59	28.40±1.23	28.23±1.42	28.23±0.96	9.35±0.19	4.72±0.12	6.79±0.24
BYOL (cross)	50.10±2.48	49.05±2.75	48.60±2.59	28.95±1.23	28.70±1.42	28.65±0.96	9.20±0.19	4.62±0.12	6.68±0.24
VICReg (in)	38.05±3.01	37.12±2.66	37.38±3.02	23.75±1.00	23.16±1.03	22.92±1.21	10.87±0.61	5.47±0.35	7.78±0.14
VICReg (cross)	38.55±3.01	37.55±2.66	37.85±3.02	24.30±1.00	23.70±1.03	23.40±1.21	10.70±0.61	5.35±0.35	7.65±0.14
Barlow Twins (in)	38.97±0.65	37.75±1.00	38.21±1.12	27.24±0.19	26.84±0.20	26.25±0.77	9.89±0.35	4.95±0.15	7.03±0.21
Barlow Twins (cross)	39.60±0.65	38.25±1.00	38.85±1.12	27.85±0.19	27.30±0.20	26.80±0.77	9.72±0.35	4.85±0.15	6.92±0.21
CLIP (in)	43.78±0.89	42.53±0.90	43.07±0.98	25.55±0.63	25.78±1.25	25.17±0.75	8.52±0.46	4.26±0.23	6.73±0.63
CLIP (cross)	44.25±0.89	42.95±0.90	43.55±0.98	26.10±0.63	26.25±1.25	25.65±0.75	8.35±0.46	4.15±0.23	6.60±0.63
TS-TCC (in)	68.56±1.19	66.90±1.22	68.10±1.30	33.61±0.72	33.11±1.09	33.91±0.79	5.61±0.15	2.70±0.06	4.69±0.38
TS-TCC (cross)	69.10±1.19	67.35±1.22	68.60±1.30	34.15±0.72	33.60±1.09	34.30±0.79	5.50±0.15	2.62±0.06	4.60±0.38
SimMTM (in)	44.78±0.62	42.48±0.37	43.60±0.62	22.34±0.28	25.68±0.41	29.72±1.78	8.77±0.18	4.61±0.32	6.90±0.18
SimMTM (cross)	45.52±0.54	43.01±0.41	44.10±0.68	22.90±0.28	26.10±0.41	30.05±1.78	8.60±0.18	4.50±0.32	6.78±0.18
TF-C (in)	31.13±0.42	30.57±0.40	31.00±0.31	30.78±0.39	28.16±0.23	30.82±1.41	12.47±0.72	6.31±0.37	7.93±0.30
TF-C (cross)	31.70±0.42	31.05±0.40	31.55±0.31	31.30±0.39	28.75±0.23	31.25±1.41	12.25±0.72	6.15±0.37	7.75±0.30
TS2Vec (in)	67.13±0.11	65.56±0.21	64.13±0.21	35.40±0.96	32.17±1.26	35.47±1.42	5.92±0.93	3.01±0.28	5.02±0.42
TS2Vec (cross)	67.70±0.11	66.05±0.21	64.70±0.21	36.00±0.96	32.75±1.26	36.05±1.42	5.85±0.93	2.95±0.28	4.95±0.42
Ours (in)	70.67±0.06	67.74±0.29	68.79±0.25	52.21±1.09	48.64±1.52	48.22±1.11	5.16±0.44	2.50±0.13	4.65±0.45
Ours (cross)	**71.20**±0.06	**68.20**±0.29	**69.30**±0.25	**53.10**±1.09	**49.20**±1.52	**48.80**±1.11	**5.05**±0.44	**2.40**±0.13	**4.55**±0.45

For experiments regarding the activity recognition and step counting tasks (Table 11), we have first used electrocardiogram signals to pretrain models and then fine tune on the specific dataset. Each method includes two rows: the first shows in-domain results (also reported in the main manuscript), and the second shows cross-domain performance. Results are organized top-to-bottom per method to facilitate easier comparison.

For the cardiovascular disease classification (CVD) task (Table 12), we pretrain models on EEG signals and fine-tune them on ECG datasets. For sleep stage classification, we pretrain on both ECG datasets and fine-tune on a small subset of the EEG dataset. The results are summarized in Table 12.

Table 12: Performance comparison of methods for *CVD* and *Sleep* in cross domain settings

Method	Chapman			CPSC			Sleep		
	Acc ↑	AUC ↑	F1 ↑	Acc ↑	AUC ↑	F1 ↑	Acc ↑	W-F1 ↑	Kappa ↑
Supervised									
FCN	84.63±2.13	95.40±0.57	82.41±2.40	63.64±1.12	91.30±0.02	60.43±1.04	71.98±0.86	63.33±0.84	62.01±1.30
ResNet	93.16±0.41	98.59±0.05	92.02±0.42	75.21±1.73	95.02±0.03	71.70±1.90	76.94±0.97	67.52±1.95	69.14±0.61
SimCLR (in)	75.28±0.57	93.55±0.25	74.04±0.50	50.10±0.41	87.20±0.07	50.10±0.24	72.45±2.32	58.93±1.59	59.47±3.20
SimCLR (cross)	76.58±0.55	93.92±0.22	75.39±0.48	51.03±0.40	87.48±0.06	50.25±0.22	73.56±2.25	59.84±1.55	60.41±3.10
BYOL (in)	77.08±0.40	93.74±0.18	75.80±0.35	52.90±0.30	87.05±0.22	50.89±0.38	70.77±0.27	58.23±0.55	55.90±1.20
BYOL (cross)	77.10±0.38	94.81±0.16	77.18±0.42	52.95±0.28	87.32±0.20	51.35±0.36	71.82±0.25	59.29±0.52	56.80±1.15
VICReg (in)	70.10±1.90	89.35±0.93	67.84±1.79	46.21±1.29	84.70±0.50	42.51±0.96	68.72±1.03	57.24±1.04	57.13±1.42
VICReg (cross)	71.31±1.83	89.78±0.90	68.75±1.70	47.38±1.26	85.09±0.47	43.25±0.91	69.75±1.00	58.10±1.01	58.06±1.38
Barlow Twins (in)	72.43±1.45	91.17±0.60	70.42±1.53	48.67±0.51	85.78±0.19	44.57±0.53	70.10±0.62	57.72±0.81	57.88±0.82
Barlow Twins (cross)	73.21±1.41	91.61±0.57	71.38±1.50	49.32±0.48	86.14±0.18	45.33±0.50	71.13±0.60	58.52±0.78	58.71±0.79
CLIP (in)	82.98±0.96	95.15±0.42	81.00±1.03	50.01±0.89	86.40±0.32	47.99±0.89	73.16±0.81	62.06±0.91	63.75±1.23
CLIP (cross)	83.69±0.94	95.39±0.39	81.85±1.01	51.13±0.85	86.61±0.31	48.88±0.86	74.25±0.78	63.10±0.89	64.80±1.20
TS-TCC (in)	73.50±0.55	90.65±0.07	71.10±0.57	51.59±1.22	86.32±0.16	50.27±1.32	62.80±1.13	52.43±1.05	48.98±1.68
TS-TCC (cross)	74.66±0.53	91.00±0.06	72.10±0.55	52.35±1.18	86.51±0.15	51.17±1.29	63.78±1.10	53.42±1.03	49.97±1.65
SimMTM (in)	84.29±1.29	95.87±0.18	83.31±1.25	51.70±0.23	87.08±0.21	50.62±0.55	74.69±1.84	63.53±1.21	65.31±2.76
SimMTM (cross)	85.21±1.25	96.03±0.17	84.20±1.20	52.34±0.22	87.33±0.20	51.52±0.53	75.88±1.78	64.60±1.18	66.31±2.71
TF-C (in)	85.84±0.39	96.10±0.10	84.71±0.40	47.86±0.69	86.27±0.05	45.42±0.66	64.50±1.80	56.77±2.21	52.61±2.41
TF-C (cross)	86.67±0.37	96.28±0.09	85.43±0.38	49.31±0.67	86.55±0.04	46.21±0.64	65.65±1.75	57.86±2.17	53.72±2.35
TS2Vec (in)	78.87±1.03	90.23±0.24	81.32±0.47	48.73±0.85	85.49±0.37	46.57±1.10	65.71±1.06	55.32±1.77	56.81±1.90
TS2Vec (cross)	80.03±1.00	90.48±0.23	82.13±0.45	49.61±0.82	85.76±0.35	47.28±1.07	66.94±1.04	56.34±1.74	57.84±1.87
Ours (in)	87.21±0.80	96.50±0.21	85.30±0.98	52.10±0.90	87.11±0.40	51.26±1.18	77.30±1.04	68.05±0.86	69.16±1.32
Ours (cross)	**88.03**±0.78	**96.78**±0.20	**86.01**±0.95	**53.11**±0.87	**88.03**±0.48	**52.02**±1.14	**78.21**±1.01	**69.01**±0.83	**70.15**±1.28

As shown in Tables 11 and 12, our method demonstrates strong generalization across domains, achieving the best performance in 17 out of 18 metrics across 6 datasets. This is particularly important because our approach employs specific transformations, such as Fourier and wavelet projections, which are inherently sensitive to the signal's sampling rate.

Despite variations in sampling rates between source and target datasets, our method consistently outperforms others. While similar resilience to sampling rate differences has been observed in prior works that employs frequency-domain representations [47], we hypothesize that the adaptability of our method stems from fine-tuning both the main encoder and the latent space mappers, which enables adaptation to cross-domain evaluation similar to previous methods.

D.5 Comparison with larger models

In our experiments, we aimed to ensure fair comparisons across methods by controlling for model capacity and training data. Therefore, we used the same or closely matched backbone architectures with the same amount of training data across methods, allowing us to isolate and evaluate the effect of the representation learning strategy itself rather than differences in model or data scale.

Foundation models, while powerful, involve substantially larger architectures and extensive pretraining on massive datasets. As such, a direct comparison in our experiments is not straightforward. Nonetheless, for completeness, we conducted experiments with CBraMod [81], one of the most recent of these foundation models for brain signals. Specifically, we used the publicly available pretrained weights and evaluated on the Sleep-EDF dataset. Features were extracted using the pretrained model, followed by either (i) training only a linear classifier or (ii) fine-tuning the full network on the same training split. We have reported the results in Table 13. CBraMod employs a transformer layer

Table 13: Comparison with the pretrained CBraMod on Sleep-EDF.

Method	**Linear Evaluation**	**Fine-tuning**
CBraMod (pretrained)	65.03 ± 0.63	$\mathbf{72.98 \pm 0.47}$
Ours	$\mathbf{69.19 \pm 1.32}$	71.13 ± 0.84

for sequence-to-sequence sleep staging, whereas we applied a linear classifier for consistency with other baselines. It is also worth noting that CBraMod was not originally trained on single-channel EEG, which may partly explain the lower evaluation performance. While fine-tuning increases the performance of CBraMod more compared to ours, it is important to note that it has a larger parameter count than our backbone, giving it greater capacity to adapt when all weights are updated according to the task. This makes direct comparison under fine-tuning less informative, as the performance difference can largely reflect model size rather than the quality of the learned representations.

By contrast, linear evaluation offers a fair basis for comparison, as only a lightweight classifier is trained. Our results show that, despite its smaller design, our approach remains competitive.

D.6 Comparison with heavy specialized augmentations

To further assess the performance of our method, we compared it against augmentation-heavy baselines, including TimesURL [82] and Finding Order in Chaos [5], which introduces frequency domain based mixup. Table 14 reports results. For Finding Order in Chaos (FOC), we reproduced the results on SPC12 (where dataset splits differ slightly from our setup), and report the original numbers from the paper for SPC22 and DaLiA. Our method consistently outperforms augmentation-heavy baselines, with improvements of approximately 8–9% across all datasets. These results highlight that

Table 14: Performance comparison of our method with augmentation-heavy baselines

Method	IEEE SPC12			IEEE SPC22			DaLiA		
	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑	MAE ↓	RMSE ↓	ρ ↑
Supervised									
FCN	15.13±0.50	21.63±0.48	52.09±5.43	16.57±0.91	26.20±0.60	55.98±0.78	12.45±0.12	18.35±0.24	56.98±0.78
ResNet	7.08±0.20	13.60±0.38	79.60±1.10	9.90±1.47	16.67±1.60	67.58±2.98	5.50±0.05	10.84±0.03	82.10±0.06
Self-Supervised									
SimCLR	12.42±0.05	20.96±0.30	73.62±0.52	16.41±0.22	22.62±0.39	52.16±1.12	16.88±0.19	22.64±0.22	56.37±0.21
TimesURL	13.40±0.42	19.85±0.61	75.13±1.22	17.92±0.88	24.73±1.25	50.10±1.97	14.62±0.31	21.57±0.44	60.17±0.11
FOC	11.15±0.34	17.89±0.83	77.43±0.30	**12.25±0.47**	**18.20±0.61**	**57.13±0.42**	10.57±0.55	20.37±0.73	62.83±0.22
Ours	**8.84**±0.50	**14.37**±0.95	**82.67**±1.30	14.06±1.09	21.48±2.01	54.88±1.89	**9.13**±0.20	**16.92**±0.56	**63.72**±0.06

our approach achieves stronger performance compared to methods based on heavy augmentations. While augmentations can be beneficial for certain datasets, their generalization is often limited due to the diverse characteristics of signals and the varying invariances required across tasks.

D.7 Significance analysis

In our experiments, some methods achieve strong results on specific datasets; however, our approach consistently ranks highest across tasks. For instance, TF-C performs well on ECG (ranking second to ours in Table 3), but its accuracy falls to about 30% on the activity recognition (Table 2).

We evaluate significance across 27 tasks (3 datasets × 3 metrics for HR, 3×3 for Activity/Step, 3×3 for CVD/Sleep). After ranking, we run the Friedman test and apply Nemenyi post-hoc comparisons. The critical difference diagram is given in Figure 8.

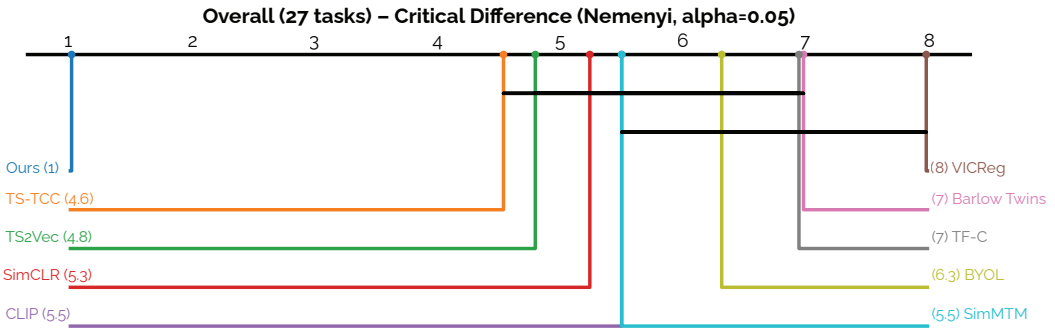


Figure 8: The critical diagram for all tasks. Numbers show average ranks (lower is better); horizontal bars connect methods not significantly different. Our method achieves the best average rank.

E Experiments

Here, we give a detailed description of datasets, architectures, metrics, and training details for our experiments. We performed our experiments on NVIDIA GeForce RTX 4090 GPUs, involving training with three random seeds for all datasets, totaling approximately 680 GPU hours including ablation. All experiments fit within 24 GB of GPU memory, without requiring excessive computational resources. We reported the mean of three runs with the standard deviation.

E.1 Datasets

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

E.1.1 Human Activity Recognition and Step counting

Clemson The Clemson dataset has 30 participants (15 males, 15 females), where each participant wore three Shimmer3 sensors. We used the IMU sensor readings from non-dominant wrists to predict step count where each sensor recorded accelerometer and gyroscope data at 15 Hz. We calculated the total magnitude of the accelerometer and fed it to the model as a pre-processing without any filtering. We used window lengths of 32 seconds without an overlap in the regular walking setting. We conducted 10-fold cross-validation, with each fold consisting of 3 subjects. Pre-training is performed using 9 folds, with the remaining fold held out for testing. The test fold is not used during either pre-training or linear fine-tuning.

HHAR Heterogeneity Dataset for Human Activity Recognition (HHAR) is collected by nine subjects within an age range of 25 to 30 performing six daily living activities with eight different smartphones—Although HHAR includes data from smartwatches as well, we use data from smartphones—that were kept in a tight pouch and carried by the users around their waists [37]. Subjects then perform 6 activities: 'bike', 'sit', 'stairs down', 'stairs up', 'stand', and 'walk'. Due to variant sampling frequencies of smart devices used in HHAR dataset, we downsample the readings to 50 Hz and apply 100 (two seconds) and 50 as sliding window length with step size, the windows are normalized to zero mean with unit standard deviation. We used the first four subjects (i.e., a, b, c, d) as source domains.

USC USC human activity dataset (USC-HAD) is composed of 14 subjects (7 male, 7 female, aged 21 to 49 with a mean of 30.1) executing 12 activities with a sensor on the front right hip. The data dimension is six (3-axis accelerometer, 3-axis gyroscope) and the sample rate is 100 Hz. 12 activities include walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing, sleeping, elevator up, and elevator down. We used the pre-processing technique with a smaller window size such that the input contains six channels with 100 features (it is sampled in a sliding window of 1 second and 50% overlap, resulting in 100 features for each window). The same normalization is also applied to windows before feeding to models. We used the same setup with UCIHAR while source subjects are chosen as the last four this time.

E.1.2 Heart Rate Prediction

IEEE SPC This competition provided a training dataset of 12 subjects (SPC12) and a test dataset of 10 subjects [36]. The IEEE SPC dataset overall has 22 recordings of 22 subjects, ages ranging from 18 to 58 performing three different activities [83]. Each recording has sampled data from three accelerometer signals and two PPG signals along with the sampled ECG data and the sampling frequency is 125 Hz. All these recordings were recorded from the wearable device placed on the wrist of each individual. All recordings were captured with a 2-channel pulse oximeter with green LEDs, a tri-axial accelerometer, and a chest ECG for the ground-truth HR estimation. During our experiments, we used PPG channels. We choose the first five subjects of SPC12 as source domains similar to *activity recognition* setup while the last six subjects of SPC22 are used for source domains to prevent overlapping subjects with SPC12.

DaLiA PPG dataset for motion compensation and heart rate estimation in Daily Life Activities (DaLiA) was recorded from 15 subjects (8 females, 7 males, mean age of 30.6), where each recording was approximately two hours long. PPG signals were recorded while subjects went through different

daily life activities, for instance sitting, walking, driving, cycling, working, and so on. PPG signals were recorded at a sampling rate of 64 Hz. The first five subjects are used as source domains.

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

E.1.3 Cardiovascular disease (CVD) classification

CPSC China Physiological Signal Challenge 2018 (CPSC2018), held during the 7th International Conference on Biomedical Engineering and Biotechnology in Nanjing, China. This dataset consists of 6,877 (male: 3,699; female: 3,178) and 12 lead ECG recordings lasting from 6 seconds to 60 seconds with 500 Hz. We use the original labelling [39] with one normal and eight abnormal types as follows: atrial fibrillation, first-degree atrioventricular block, left bundle branch block, right bundle branch block, premature atrial contraction, premature ventricular contraction, ST-segment depression, ST-segment elevated. We resampled recordings to 100 Hz and excluded recordings of less than 10 seconds.

Chapman Chapman University, Shaoxing People's Hospital (Chapman) ECG dataset which provides 12-lead ECG with 10 seconds of a sampling rate of 500 Hz. The recordings are downsampled to 100 Hz, resulting in each ECG frame consisting of 1000 samples. The labeling setup follows the same approach as in [40] with four classes: atrial fibrillation, GSVT, sudden bradycardia, and sinus rhythm. The ECG frames are normalized to have a mean of 0 and scaled to have a standard deviation of 1. We split the dataset to 80–20% for training and testing as suggested in [40].

We choose leads I, II, III, and V2 during our experiments for both ECG datasets. We followed a similar setup with prior works [84] and considered each dataset as a single domain different from previous tasks. The fine-tuning of the linear layer, which is added to the frozen pre-trained encoder, is performed with 80% of the same domain.

E.1.4 Sleep stage classification

We used the Sleep-EDF dataset which has five classes: wake (W), three different non-rapid eye movements (N1, N2, N3), and rapid eye movement (REM). The dataset includes whole-night PSG sleep recordings, where we used a single EEG channel (i.e., Fpz-Cz) with a sampling rate of 100 Hz. We followed the same data split as TSTCC [44], with no additional pre-processing. The only difference is that, for linear probing, we used a random 10% subset of the unseen data rather than the full set, reflecting our setup where labeled data is significantly smaller than unlabeled data.

E.2 Baselines

E.2.1 Supervised

FCN We use a 1D Fully Convolutional Network (FCN) that processes multichannel temporal inputs. The model consists of three convolutional layers with increasing filter sizes (32, 64, and 128), each followed by max pooling operations to progressively reduce the temporal resolution. A final linear layer maps the output to class logits. We chose this architecture for the supervised and self-supervised learning paradigms as it was widely used before in the literature [5].

ResNet We use the same backbone as in the self-supervised methods for the supervised baseline, integrating a linear layer and training the model from scratch using random initialization.

We perform a grid search over key hyperparameters for both networks, focusing on learning rate and batch size. The learning rate is initialized at 1e-3 and reduced by half if validation performance does not improve for 15 epochs. The batch size is fixed at 64.

E.2.2 Self-Supervised

Fundamentals

SimCLR SimCLR [1] introduces a contrastive learning framework for self-supervised visual representation learning. The method relies on maximizing agreement between differently augmented views of the same image via a contrastive loss in the latent space. We follow the previous implementations of SimCLR for time series [5, 85].

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

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

$$\ell = \lambda\left[s(z, z')\right] + \mu\left[v(z) + v(z')\right] + \nu\left[c(z) + c(z')\right], \tag{24}$$

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

Barlow Twins Barlow Twins [45, 86] presents an objective function that naturally avoids collapse for SSL by measuring the cross-correlation matrix between the outputs of two identical networks fed with augmented versions of a sample, and making it as close to the identity matrix as possible. This causes the embedding vectors of augmented versions of a sample to be similar, while minimizing the redundancy between the components of these vectors. Following the original implementation, we applied batch normalization to the extracted embeddings and set the hyperparameter λ coefficient (in Equation 25) to 0.005.

$$\mathcal{L} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2, \tag{25}$$

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

CLIP Contrastive Language–Image Pretraining (CLIP) [46] learns joint representations by aligning paired image and text embeddings through contrastive learning. To compare our method with this, we adapted a similar strategy for time series by treating the time-domain signal and its Fourier-transformed version as two modalities. Specifically, we applied a CLIP-style objective by training separate encoders for the time and frequency domains and maximizing the similarity between their paired embeddings. During inference, we only used time encoder with linear probing. This setup mirrors CLIP's approach to aligning image-text pairs, but instead aligns time-frequency representations of the same signal.

Temporal sequence

TS-TCC We follow the same architecture implementation with the losses, i.e., contextual and temporal contrasting. TS-TCC proposed applying two separate augmentations, one augmentation is weak (jitter-and-scale) and the other is strong (permutation-and-jitter). The authors also change the strength of the permutation window from dataset to dataset. In our experiments, we first used the original augmentations for each time series task, however, we observed performance decreases depending on the signal type. We, therefore, applied the specific augmentations for each time series, where we observed a general performance improvement in other SSL techniques as well.

TS2Vec TS2Vec [49] is a SSL method specifically designed for time series based on contrastive (instance and temporal wise) learning in a hierarchical way over augmented context views where the context is generated by applying timestamp masking and random cropping on the input time series. Following the original framework, we use a dilated CNN architecture with a depth of 10 and hidden size of 64, which has a similar number of parameters with the architectures used by other SSL methods. The batch size is set to 8 and the number of epochs to 120, following the original manuscript. Although we experimented with larger batch sizes, as SSL methods often benefit from them, we observed no performance gains.

TF-C The Time-Frequency Consistency (TF-C) method [47] introduces a self-supervised learning framework for time series data by aligning representations from both time and frequency domains using the absolute Fourier transform. TF-C employs specific augmentations in both domains, such as jittering in the time domain, and perturbations like adding or removing/decreasing frequency components in the Fourier domain, to create diverse views. During inference, TF-C utilizes both the time-domain and frequency-domain encoders, combining their outputs to form the final representation, thereby integrating information from both domains for downstream tasks. In contrast, our method avoids this by requiring only a single encoder at inference. We use the original implementation provided at github.com/mims-harvard/TFC-pretraining.

SimMTM SimMTM [48] presents a masked modeling framework tailored for temporal data. In time series, semantic information is heavily embedded in temporal variations, and random masking may disrupt critical patterns, making reconstruction unnecessarily difficult. SimMTM mitigates this by treating masked modeling as a manifold learning problem. Instead of reconstructing masked points directly from nearby unmasked values, it recovers them via weighted aggregation from multiple complementary masked sequences. We use the same ResNet backbone for SimMTM to ensure fair comparison with other SSL methods. We also experimented with the original backbone proposed in the paper but observed no performance improvement. For our evaluation, we follow the original manuscript's hyperparameter settings for the masking ratio and the number of positive (masked) series. In addition, we explore the effects of training dynamics by running SimMTM with our higher batch size (1024) compared to the paper's smaller batch sizes (128–256), and we test both short training epochs (40–50 as used in the original) and longer schedules, treating these as part of a hyperparameter search. We observed performance degradation when reducing the number of epochs, so we run all baseline methods for the same number of epochs to ensure a fair comparison. We use the official implementation provided at github.com/thuml/SimMTM.

E.3 Implementation Details

Here, we have provided the details of the architectures, and hyperparameters. Primarily, we used the 1D ResNet [51] implementation in the supervised settings. While some alternative deep learning models can perform better in a specific time series tasks such as the combination of convolutional and LSTM layers [87, 88], we focused on residually connected convolutional architectures as backbones for representation learning.

E.3.1 Architectures

Here, we present the details of architectures that are investigated for the performance of shift-invariant techniques. Some details that are not given in the tables are as follows. Batch normalization [89] is applied after each convolutional block. ReLU activation is employed following batch normalization, in line with [50]. We also applied a Dropout [90] with 0.5 after each activation and before the convolutions. Finally, a global average pooling is implemented before the linear layers.

Table 15: Model architectures used in our experiments

(a) ResNet architecture for the main encoder

# Blocks	Layer	Kernel	Output
1	Input (C,T)	-	(C, T)
1	Conv	(5, 1)	(64, T/2)
8	Conv	(5, 1)	(128, T/4)
	Conv	(5, 1)	(128, T/4)
1	Linear	-	(n_classes,)
# Parameters for *dataset* (C,F,T)			
IEEE SPC & Dalia (C=1, T=200)			≈210k
Chapman & CPSC (C=4, T=1000)			≈197k
Clemson (C=1, T=240)			≈200k
HHAR (C=6, T=51)			≈200k
USC (C=6, T=100)			≈200k
Sleep (C=1, T=3000)			≈210k

(b) Architecture for $\mathcal{W}(a, b)$

Layer	Input → Output	Description
Input	(C, F, T)	Input
AvgPool	$\rightarrow (C, F/2, T/2)$	
AvgPool2	$\rightarrow (C, F/4, T/4)$	
Conv1	$\rightarrow (N, F, T)$	Conv block
Conv2	$\rightarrow (2N, F/2, T/2)$	
Concat1	+ Pool1	Merge
Conv3	$\rightarrow (3N, F/4, T/4)$	
Concat2	+ Pool2	Merge
Conv4	$\rightarrow (4N, F/8, T/8)$	
FC1	$\rightarrow (1, 25)$	Linear (time)
FC2	$\rightarrow (128)$	Final output
# Parameters for *dataset* (C,F)		
IEEE SPC & Dalia (C=1, F=48)		≈500k
Chapman & CPSC (C=4, F=48)		≈500k
Clemson (C=1, F=48)		≈500k
HHAR (C=6, F=48)		≈550k
USC (C=6, F=48)		≈550k
Sleep (C=1, F=48)		≈520k

(c) Architecture for $\mathcal{F}(\mathrm{x})$

Branch	Layer	Output
Amplitude	Conv1D	(16, T)
	Residual	(32, T/2)
	Residual	(64, T/4)
	Linear	(64)
Phase	Conv1D	(16, T)
	Residual	(32, T/2)
	Residual	(64, T/4)
	Linear	(64)
Output	Concatenate	(128)
All datasets		≈55k

(d) Architecture for mappers $\Phi^{t \rightarrow d}$

Layer	Kernel Size	Output Size
Input (1, L)	–	(1, L)
Conv1D	(3, 1), stride=2	(64, L/2)
ReLU	–	(64, L/2)
Conv Transpose	(3, 1), stride=2	(1, L)
All datasets		≈500

Our latent space mappers are lightweight, with each containing approximately 500 parameters, resulting in a total of only 1k additional parameters during inference. This keeps the overall inference cost of our method low. Moreover, even when baseline models are scaled up to match or exceed the total parameter count of all encoders, they still fall short in performance, highlighting the effectiveness of our approach over brute-force model scaling.

E.3.2 Augmentations

We applied commonly used augmentations for each task, with details listed in Table 16. In each epoch, two random augmentations were applied per sample and instance discrimination is applied.

For the *heart rate prediction* task, we used: `permutation`, `noise`, `scale`, and `shift`.

For *activity recognition* and *step detection* from inertial signals, we used: `permutation`, `noise`, `scale`, `shift`, and `rotation`.

For *cardiovascular disease* and *sleep stage* classification, we used: `resample`, `noise`, `scale`, `negate`, and `shift`.

These augmentation sets follow prior work [5, 84, 85, 91], ensuring task-relevant diversity.

Table 16: Common time series augmentations

Domain	Augmentation	Details
Time	Noise	Add Gaussian noise sampled from normal distribution, $\mathcal{N}(0, 0.4)$
	Scale	Amplify channels by a random distortion sampled from normal distribution $\mathcal{N}(2, 1.1)$
	Negate	Multiply the value of the signal by a factor of -1
	Permute	Split signals into no more than 5 segments, then permute the segments and combine them into the original shape
	Resample	Interpolate the time-series to 3 times its original sampling rate and randomly down-sample to its initial dimensions
	Rotation	Rotate the 3-axial (x, y, and z) readings of each IMU sensor by a random degree, which follows a uniform around a random axis in the 3D space. (Only applied for *Activity Recognition*)
	Time Flip	Flip the time series in time for all channels, i.e., $\mathbf{x}_{Aug}[n] = \mathbf{x}[-n]$
	Shift	Apply circular shift with a random amount
	Random Zero Out	Randomly chose a section to zero out
	Permutation + Noise	Combination of Permutation and Noise
	Noise + Scale	Combination of Noise and Scaling
Frequency	Highpass	Apply a highpass filter in the frequency domain to reserve high-frequency components
	Lowpass	Apply a lowpass filter in the frequency domain to reserve low-frequency components
	Phase shift	Shift the phase of time-series data with a randomly generalized number
	Noise in Frequency	Add Gaussian noise, sampled from normal distribution $\mathcal{N}(0, 0.5)$, to the frequency spectrum

Although we also experimented with frequency domain augmentations given in Table 16 for each task, we observed consistent performance degradation. As a result, we excluded these frequency augmentations from all baseline comparisons.

E.3.3 Transformations

Fourier transformation For the Fourier transformation, we compute the FFT using the full length of each input signal without applying any task-specific padding or optimization. Since the input signals are real-valued, we calculate only the positive (real) frequencies, leveraging the Hermitian symmetry of the spectrum. We normalize the FFT using $\frac{1}{\sqrt{n}}$ by setting `norm='ortho'` in PyTorch's [92] `torch.fft.rfft`, ensuring the transformation is orthonormal.

Wavelet transformation For the wavelet transformation, we use the continuous Morlet wavelet with 48 logarithmically spaced scales computed via `np.geomspace(1, 128, num=48)`, without any task-specific tuning or dataset-dependent adjustments. This configuration is applied uniformly across all datasets to highlight the versatility and generality of our method. We use PyWavelets [93] implementation.

F Computational Overhead of Techniques

We analyze the computational overhead of SSL techniques designed for temporal data. We break down the main operations of TS2VEC [49], TS-TCC [44], TF-C [47], and simMTM [48], and provide empirical comparisons in Table 17.

TS2Vec TS2Vec samples overlapping subsegments for hierarchical contrastive loss. For each sample, it randomly selects a crop length $l \in [2^{u+1}, T]$, defines a crop interval $[c_l, c_r]$ and an extended context interval $[e_l, e_r]$, and applies per-sample random offsets to generate two augmented views. These views are passed through a shared encoder, producing two embeddings. TS2Vec aligns their overlapping subsegments using a temporal contrastive loss and recursively applies pooling to compute multiple loss terms at increasingly coarser resolutions. This multiscale loss is recomputed for every pair at each layer depth, leading to quadratic scaling with the number of layers. Moreover, because the crops are randomly sampled, TS2Vec forwards large sections of samples multiple times with different windows. Despite using a shared encoder, this repeated window sampling and depth-wise contrastive loss accumulation make training slow, especially on long sequences.

TS-TCC In each batch, TS-TCC generates two augmented views using weak and strong augmentations, and processes both through a shared encoder to obtain temporal feature sequences. The temporal contrasting module performs a cross-view prediction task: it uses an autoregressive model to summarize the past of one view into a context vector and predicts the future of the other view using linear projections. For each sample, this involves computing predictions for multiple future steps, comparing them against the target view, and accumulating a contrastive loss. This adds sequential dependency to training, as the model must first encode the past then perform multiple forward passes to compare predicted and true representations. While TS-TCC leverages efficient architectures like Transformers, the temporal prediction task and dual-view contrastive losses make it less parallelizable than methods that process views independently or avoid cross-timestep dependencies.

TF-C In each batch, TF-C computes time and frequency representations in parallel using separate encoders and applies multiple contrastive losses. The input is first transformed via FFT to obtain frequency-domain features. Two augmented views are generated: one in time via temporal augmentations (e.g., jittering, scaling), and one in frequency via spectral perturbations (adding/removing frequency components). Both views are processed by their respective encoders and projectors. TF-C then computes contrastive losses in three spaces: time-time, frequency-frequency, and time-frequency. The time-frequency consistency loss includes four cross-domain pairings (e.g., original time vs. perturbed frequency) and is implemented using a triplet-style margin loss. This results in eight forward passes per sample (two augmentations × two domains × two projections).

simMTM SimMTM introduces two coupled components: masked contrastive learning and masked reconstruction. In each batch, a contrastive similarity matrix is computed over all samples and their masked variants, involving a full pairwise dot-product followed by calculating KL-divergence with soft targets. The reconstruction further adds cost by aggregating weighted representations using the similarity matrix and applying a linear projection to reconstruct masked sequences. While these components operate in parallel, the simultaneous use of large similarity matrices and reconstruction targets slows down training, especially with high masking rates or long windows.

Table 17 reports the average execution time per epoch for each method, measured using an NVIDIA GeForce RTX 4090 GPU with 24GB of memory. For each run, we timed only the core training operations using a unified timing function that accounts for both CPU and GPU execution. The table presents the mean and standard deviation across the five runs. For our method, the computation of wavelet and Fourier domain transformations for a batch of samples takes ≈ 2 seconds.

Table 17: Time taken (in seconds) for each SSL technique for a single epoch

Metric	Ours	TS2Vec	TS-TCC	TF-C	simMTM
Execution Time (sec)	<u>18.67</u>±0.05	310±0.07	**6.47**±0.09	21.61±0.02	68.14±0.99

A key advantage of our method is that the frequency-domain preprocessing (FFT and Gabor transforms) is performed only once and cached, rather than recomputed at every epoch. In contrast, traditional temporal augmentations (e.g., permutation, scaling with noise) must be regenerated each epoch, introducing repeated overhead.

To quantify this difference, we report the runtime for a batch of 1024 samples (each being a univariate time series of length 200) measured on an NVIDIA RTX 4090 (Table 18). FFT and Gabor transforms take 0.0075s and 3.59s, respectively, but these costs are paid once and do not scale with the number of epochs. By comparison, temporal augmentations scale linearly with training length: permutation requires $0.021 \times$ epoch count seconds and scale+noise requires $0.010 \times$ epoch count seconds. At 500 epochs—common for SSL methods—our preprocessing yields about a $4\times$ speedup, and the advantage grows further for longer runs.

Table 18: Runtime of preprocessing steps (batch size 1024) measured on NVIDIA RTX 4090. Unlike augmentations, FFT and Gabor transforms are cached after one-time computation.

Transform	**1 epoch (s)**	**500 epochs (s)**
FFT	0.0075	0.0075
Gabor	3.59	3.59
Permutation	$0.021 \times$ epoch_count	10.5
Scale + Noise	$0.010 \times$ epoch_count	5.0

We note that our comparison focuses on the most commonly used temporal augmentations such as permutation and scaling with noise. If we additionally implement and compare with the specialized time-series augmentation strategies [5], which often require auxiliary models or complex frequency-domain modifications, the computational gap becomes even larger. In such cases, the advantage of our one-time preprocessing approach increases substantially, as these specialized augmentations add both runtime overhead and architectural complexity, whereas our method retains efficiency.

G Expanded Related Work

This section extends the related work discussed in the main paper by highlighting methods that aim to eliminate reliance on augmentations in self-supervised learning, as well as efforts to mitigate the representational biases introduced by common augmentations.

Representation invariance Representation learning methods have shown to be effective in several tasks. Several works explain this success by relating the success of learned representations to invariancy caused by data augmentations [17, 21, 94, 30]. However, such invariance could be harmful to downstream tasks [95, 96] if they rely on the characteristics of the data augmentations, e.g., location, amplitude-sensitive. One proposed solution to this limitations involves constructing separate embedding spaces, each invariant to all but one augmentation [17]. While this approach allows for disentangling the effects of different augmentations, it is constrained by the number of predefined embedding spaces and comes with increased computational cost. Another method introduces an augmentation-aware module that learns to predict the differences in augmentation parameters (e.g., cropping positions) between two randomly transformed samples [21].

However, this technique has several drawbacks. First, it requires explicit parameterization of each augmentation, which is particularly challenging for temporal data where augmentation semantics (e.g., shifts, warps) are less structured than in images. Second, its effectiveness remains dependent on the choice and quality of augmentations used, reintroducing domain-specific tuning.

In contrast, our method leverages principled, isometric transformations that preserve the geometry of the data, without enforcing explicit invariance. This allows the model to retain signal characteristics that may be important for downstream tasks and would otherwise be suppressed by augmentations [17]. For example, when shift invariance is beneficial in temporal tasks, our approach captures it naturally through domain projections such as the Fourier transform [97], without depending on handcrafted augmentations or specialized architectures [98].

Augmentation-free SSL Learning representations without relying on augmentations has been explored in domains where strong transformations risk distorting the data structure, such as graphs [99, 100] and time series [8, 74]. In graph learning, for instance, augmentation-free approaches generate alternative views by identifying nodes with similar local structures and global semantics [99]. However, these strategies are specialized for graph topologies and do not directly translate to temporal sequences. More recently, authors in [8] introduced random projections as a modality and application-agnostic alternative for self-supervised learning. While this strategy provides a generic way to avoid manual augmentations, in practice, domain-aware contrastive methods equipped with carefully designed augmentations still tend to outperform projection-based approaches when reliable inductive biases about the data are available.

Our method goes a step further by eliminating the need for augmentations across diverse time-series applications, while still outperforming specialized augmentation strategies tailored to specific tasks, thereby highlighting its algorithmic superiority.

For time-series data, a recent approach introduced a temporal-frequency co-training model for semi-supervised learning, using time and Fourier domain transformations to generate pseudo-labels [74]. These methods generally overlook the non-stationary nature of temporal signals. Relying solely on Fourier transforms limits representation to global frequency patterns and may miss short-duration events. In contrast, our method incorporates wavelet frames to capture local temporal variations, enabling finer resolution of transient signal components which are critical for representation learning.

Moreover, prior work often assumes that data should cluster similarly in time and frequency domains [74], implicitly treating the two latent spaces as geometrically aligned. Our empirical results challenge this assumption, showing that latent representations across domains differ. To address this, we introduce domain-specific latent space mappers that align and exploit the complementary structure of each space, enhancing representation learning.