Efficient Time Series SSL via Signal Descriptors

Parv Thacker*
Computer Science and Engineering
Indian Institute of Technology
Gandhinagar, India
parv.thacker@iitgn.ac.in
0009-0008-4813-6640

Ayush Shrivastava*
Computer Science and Engineering
Indian Institute of Technology
Gandhinagar, India
shrivastavaayush@iitgn.ac.in
0009-0003-7241-0447

Nipun Batra
Computer Science and Engineering
Indian Institute of Technology
Gandhinagar, India
nipun.batra@iitgn.ac.in
0000-0002-0736-7169

***Abstract*—Self-supervised learning (SSL) learns representations from unlabeled data by constructing a supervisory signal from the data itself. Most existing methods build this signal through augmentation, applying image-inspired transformations to the input and training the model to be invariant to them, often via a contrastive objective. In the time series domain, these include jittering, flipping, scaling, and temporal permutation. However, these transformations can produce signals that are no longer physically coherent in the time domain, encouraging invariances that discard information essential for downstream tasks. A second paradigm, masked modeling, reconstructs withheld segments of the signal, but this assumes missing values can be recovered from local context, an assumption that often breaks down in non-stationary domains such as biomedical sensing. We argue that time series SSL should instead capture the signal's intrinsic structure. To this end, we train an encoder to predict statistical, temporal, and spectral descriptors extracted from the raw signal as a direct supervisory target. This yields an interpretable objective that requires neither augmentation nor contrastive learning. We evaluate the approach across two sensing datasets, four backbone architectures, multiple label fractions, and a range of hyperparameters. Feature-based supervision consistently ranks among the top methods while requiring the least compute, and it is competitive with, or even outperforms, conventional SSL under frozen fine-tuning. These results suggest that time series descriptors are an effective yet underexplored source of self-supervision.**

***Index Terms*—Self-Supervised Learning, Time Series Descriptors, Representation Learning**

I. Introduction

Time series data are prevalent across many domains, including healthcare [1], remote sensing [2], industrial and household monitoring [3], finance [4], among others [5]. Recent advances in deep learning and sensing techniques have enabled large-scale signal collection across various systems [6].

Despite the abundant availability of raw data, the volume of signals frequently exceeds what can be practically labeled by human annotators [7]. Hence, supervised learning approaches are often unable to utilize the majority of available time series data, motivating the development of self-supervised learning (SSL) methods capable of learning meaningful representations directly from unlabeled signals [8], [9]. Augmentation-based SSL methods, such Barlow Twins [10], and BYOL [11], have shown strong performance on computer vision tasks [12] and have been utilized extensively for time series tasks [13]. These methods generate transformed views of the input samples and train the representations such that representations of related views remain similar, whereas those of unrelated views remain separated [8].

In computer vision, approaches such as flips, masks, crops, color augmentations, and geometric transformations are used to create consistent yet diverse views [14]. On the other hand, in time series, we use jittering, scaling, masking, signal reversals, etc., for augmentations [15], [16].

While these augmentations provide variance for SSL tasks, they may not always be semantically accurate to the real-world signals [17]. Jittering, for instance, forces invariance to noise, effectively training the encoder to suppress high-frequency variations that may encode meaningful events. In applications such as physiological, industrial, or environmental monitoring, even small signal fluctuations can correspond to meaningful events, anomalies, or system states [15].

Time series data, unlike images, are often causal in nature and hence encode temporal progression and frequency structure, as well as physical constraints or amplitude-dependent information. In many applications, this information (ordering, continuity, etc.) is crucial to the meaning of the signal or the task itself. For example, reversing the signals, scaling, or aggressively permuting a signal might destroy the necessary signals or may create physically impossible samples. Hence, the derived augmentation strategies may distort the meaning of the signals or miss crucial information during the representation learning process. Therefore, there is a growing need for self-supervised learning techniques that capture the semantics of time series data.

Handcrafted features, on the other hand, have played a crucial role in classical time series analysis. Frameworks such as TSFEL [18] and TSFRESH [19] provide a rich collection of temporal, spectral, and statistical features that encode the meaningful properties of sequential signals. While modern DL techniques replace handcrafted features, these features still encode valuable low-level information.

In this paper, we investigate a TSFEL-based SSL framework for time series analysis. We examine whether handcrafted time series descriptors can improve not only performance but also robustness to optimization choices and compute efficiency across various tasks. We perform a large-scale evaluation

* Both authors contributed equally to this research.
Corresponding Author : Parv Thacker (email: parv.thacker@iitgn.ac.in)

across multiple datasets, architectures, and SSL baselines, and compare them via a normalized rank-based evaluation framework.

Across over 1400 experiments spanning two wearable sensing benchmarks (HHAR activity recognition [20] and PPG-Dalia heart rate estimation [21]), four backbone architectures (ResNet-18 [22], MLP, MLP-Mixer [23], and PatchTST [24]), multiple label fractions, frozen and unfrozen finetuning regimes, and a range of hyperparameter settings, we evaluate feature-based self-supervised learning against six conventional SSL baselines. Methods are compared via a rank-based evaluation framework that aggregates performance across all configurations and avoids best-case reporting. We find that feature-based supervision consistently ranks among the strongest-performing approaches while incurring the lowest computational cost and remains stable across finetuning regimes and hyperparameter choices. These findings suggest that utilizing domain-informed time series features in self-supervised objectives offers a robust and efficient alternative to augmentation-based representation learning methods

II. Related Work

Using handcrafted signal properties as self-supervised targets has been explored in various modalities. In vision, tasks involved predicting objectives such as image colorization [25] or spatial patch arrangements [26]. MaskFeat [27] showed that predicting HOG features, a handcrafted descriptor, outperforms pixel-level reconstruction as a masked pretraining objective. data2vec [28] generalized this idea across speech, vision, and language modalities, using contextualized representations rather than raw inputs as prediction targets. Feature-based pretraining has been largely missing from the SSL literature for time series, and TSFEL [18] and TSFRESH [19] have demonstrated that statistical, temporal, and spectral descriptors capture task-relevant signal descriptions.

A major challenge in SSL research is that results are highly sensitive to the evaluation protocol. Prior work has shown that rankings between SSL methods change substantially depending on whether the encoder is finetuned, the amount of labeled data available, and the choice of hyperparameters [29], cautioning against conclusions drawn from a single evaluation setting. For time series, Haresamudram et al. [30] conducted a large-scale empirical study evaluating wearable HAR in various sensor positions, activity types, and label availability. The authors showed that method behavior changes significantly between settings, motivating more systematic comparison protocols. Motivated by the issues, rather than reporting performance under a single favorable setting, we aggregate ranks across many experimental configurations to measure performance consistency.

We address these gaps by introducing feature-targeted supervision as a simple, augmentation-free self-supervised alternative for time series representation learning, and we evaluate it across a wide range of backbones, fine-tuning regimes, hyperparameters, and data fractions using a rank-based comparison framework that aggregates performance across all configurations to avoid best-case reporting.

III. Methodology

A. Overview of the Proposed Framework

We propose a self-supervised learning (SSL) framework for time series representation learning based on predicting handcrafted multi-domain time series descriptors. Rather than using supervisory signals generated by contrastive objectives, signal augmentations, etc., the proposed approach uses time series features as targets during pretraining.

The proposed framework consists of three stages. First, the fixed-length time series windows are processed using the TSFEL feature extraction library to generate a set of statistical, temporal, and spectral descriptors. These features are then used as self-supervised learning targets. Second, an encoder is trained to predict the extracted handcrafted feature representations directly from the raw input signal.

B. Time Series Feature-Based Self-Supervised Learning

Following the predictive SSL paradigm established by MaskFeat and data2vec (Section 2), the supervisory signal is derived directly from the input, requiring no human annotation. The core idea of the proposed framework is to use interpretable multi-domain time series descriptors as supervisory signals for representation learning. Rather than relying on techniques such as masking, contrastive objectives, or reconstruction losses, we train a neural network to predict a set of time series characteristics directly from the input.

Given an input time series segment $\mathbf{X} \in \mathbb{R}^{T \times C}$, where T is the window length and C is the number of channels, an engineered time series feature extractor obtains a target feature vector $\mathbf{f}(\mathbf{X}) \in \mathbb{R}^d$. The feature extractor is applied independently to each channel of $\mathbf{X}$, and the resulting per-channel descriptors are concatenated to form the target vector, where $d = n_f \times C$ and n_f is the number of selected features per channel. The extracted descriptors comprise descriptors from all three: statistical, temporal, and spectral characteristics of the signal. Statistical descriptors summarize the distributional properties of a dataset, such as the mean, variance, skewness, etc. Temporal descriptors capture signal behavior through features such as zero-crossing rate, autocorrelation, and temporal entropy. Spectral descriptors characterize frequency-domain behavior using features such as the spectral centroid, dominant frequency, and spectral entropy.

To evaluate how the number of features used affects the models, multiple feature subsets were evaluated, consisting of 15, 30, 45, and 90 signal descriptors, each selected from an equal number of descriptors across domains. These were chosen to span from compact representations to richer descriptor sets, while remaining divisible by three to ensure equal domain coverage across statistical, temporal, and spectral descriptors.

During self-supervised pretraining, the encoder network, $E(\cdot)$, maps the input segment to a latent representation $\mathbf{z} \in \mathbb{R}^D$,

TABLE I: Summary of datasets used in this study.

Dataset	Task	Channels	Target
HHAR [20]	Classification	6	Activity Class
PPG-Dalia [31]	Regression	7	Heart Rate

$$\mathbf{z} = E(\mathbf{X}), \quad \mathbf{z} \in \mathbb{R}^D \tag{1}$$

A lightweight prediction head, $P(\cdot)$, then projects the latent representation into the time series feature space,

$$\hat{\mathbf{f}}(\mathbf{X}) = P(\mathbf{z}), \quad \hat{\mathbf{f}} \in \mathbb{R}^d \tag{2}$$

The network is trained to minimize the difference between the feature vector $\hat{\mathbf{f}}(\mathbf{X})$ and the target feature vector $\mathbf{f}(\mathbf{X})$. Feature descriptors are standardized at the dataset level before training. The self-supervised objective is defined as the mean squared error (MSE) between predicted and target feature values,

$$L_{\text{feat}} = \frac{1}{d}\sum_{i=1}^{d}(\hat{f}_i - f_i)^2 \tag{3}$$

where $d = n_f \times C$ denotes the dimensionality of the target feature vector, n_f is the number of selected features per channel, and C is the number of input channels.

The exact feature configurations for each subset are provided in the project repository for reproducibility.

C. Backbone Architectures

To evaluate the generality of the proposed SSL architecture, the experiments were conducted across multiple backbones, including multilayer perceptron (MLP), convolution, and transformer-based architectures. The convolution architecture consists of a 1D ResNet-18 encoder [22]. To evaluate the transformer-based architecture, we conducted experiments using PatchTST [24], which represents time series signals as sequences of temporal patches. Two MLP-based architectures were additionally considered. The first is a fully connected encoder that provides a lightweight baseline. The second is an MLP-Mixer [23], which combines token-mixing and channel-mixing operations to learn temporal features.

For all backbones, the same time series feature prediction pretraining was employed, alongside the conventional SSL baselines during pretraining.

D. Experimental Setup

Experiments were conducted on two wearable sensing benchmarks, one for classification and one for regression. Together, these cover both discrete and continuous prediction targets, two distinct sensor modalities (inertial and photoplethysmography), and multiple subjects and device positions, providing a representative evaluation scope within the wearable sensing domain. Human activity recognition experiments were performed using the Heterogeneity Human Activity Recognition (HHAR) dataset [20], containing inertial sensor recordings collected during activities such as walking, sitting, standing, stair climbing, and cycling. The downstream task is multi-class activity classification.

TABLE II: Experimental grid structure: permutation factors and configurations per backbone–method–dataset pair.

Group	Factor	Values	Configs
LR Robustness	Finetuning regime	Frozen, Unfrozen	12
	Pretraining LR	10^{-3}, 10^{-2}	
	Finetuning LR	10^{-5}, $3*10^{-4}$, $3*10^{-3}$	
Label Scarcity	Label fraction	5%, 100%	4
	Finetuning regime	Frozen, Unfrozen	
Head Augmentation	Prediction head	Linear, MLP	4
	Additional LR	10^{-3}, 10^{-4}	
Per backbone–method–dataset pair			20
Total Experiments (4 backbones × 9 methods × 2 datasets × 20 configs)			1440

Regression experiments were conducted using the PPG-Dalia dataset [21], a wearable physiological sensing benchmark that includes photoplethysmography (PPG), accelerometer, and additional sensor modalities. The downstream task is continuous heart-rate estimation. For this work, only PPG and accelerometer channels were retained, while other sensor modalities were excluded to focus on signals commonly available in consumer wearable devices.

Signals were segmented into fixed-length windows according to the predefined evaluation protocols of each benchmark [31] and standardized per channel using training-set statistics. Predefined train, validation, and test splits were used for the experiments. To evaluate the effects of label availability, the experiments were repeated with multiple training fractions generated by random sampling. For downstream adaptation, a common pipeline was implemented across all backbones and methods, with a prediction head and two fine-tuning regimes: one with the encoder frozen and only the prediction head trained, and another with all network parameters fine-tuned.

The proposed framework was compared with several representative self-supervised learning baselines spanning contrastive, redundancy-reduction, reconstruction, and masked-modeling paradigms. Specifically, experiments included BYOL [11], and Barlow Twins [10] as contrastive and redundancy-reduction approaches, TS2Vec [32] as a time series representation learning baseline, SimMTM [33] as a masked time series modeling method, and MPM [24] as a PatchTST-based masked prediction approach.

E. Evaluation Protocol

Each self-supervised learning method was evaluated across multiple combinations of datasets, backbone architectures, training fractions, encoder adaptation strategies, and optimization settings.

Experiments were conducted using ResNet18 [22], MLP, MLP-Mixer [23], and PatchTST [24] backbones. The full experimental grid was structured across three groups as detailed in Table II, resulting in a total of over 1400 experiments covering both tasks.

TABLE III: Summary results on the HHAR activity recognition task aggregated across all experimental units. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 9 (worst). Avg Rel Score (↑) is the average of relative F1 scores across experimental configurations from 1 (best) to 0 (worst). Best results per condition are marked in **bold**.. Best results per condition are marked in **bold**. *The proposed approach, TSFEL90 achieves both the best average rank (3.44) and highest Avg Rel F1-score (0.73), with Barlow being close second. Rest of the baselines perform notably worse.*

Method	Avg Rank ↓	SD	Avg Rel F1 ↑	SD
barlow	*3.48*	4.00	**0.73**	0.38
byol	5.16	3.45	0.55	0.32
mpm	7.15	3.06	0.21	0.28
simmtm	6.43	3.15	0.39	0.31
ts2vec	7.22	2.45	0.31	0.29
tsfel15	4.75	3.03	0.62	0.32
tsfel30	4.15	2.86	0.66	0.30
tsfel45	4.07	2.95	0.65	0.31
tsfel90	**3.44**	3.22	**0.73**	0.33

TABLE IV: Summary results on the PPG-Dalia heart-rate estimation task aggregated across all experimental units. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 9 (worst). Avg Rel Score (↑) is the average of relative MAE across experimental configurations from 1 (best, lowest MAE) to 0 (worst, highest MAE). Best results per condition are marked in **bold**. *The proposed approach, TSFEL90 achieves the best average rank (4.25) and the highest Avg Rel Score (0.69), with extracted time series features consistently matching or outperforming all self-supervised methods.*

Method	Avg Rank ↓	SD	Avg Rel Score ↑	SD
barlow	4.90	3.76	0.56	0.37
byol	5.19	3.74	0.54	0.37
mpm	5.66	3.83	0.48	0.37
simmtm	6.31	3.57	0.40	0.36
ts2vec	5.41	3.53	0.55	0.34
tsfel15	4.64	3.43	0.60	0.35
tsfel30	4.39	3.36	0.66	0.32
tsfel45	4.68	3.33	0.63	0.33
tsfel90	**4.25**	3.32	**0.69**	0.31

Performance was evaluated using task-specific metrics. For classification tasks, we report the macro F1-score, while for regression tasks, we report the mean absolute error (MAE). To ensure fair comparison, methods were ranked only within the experimental settings.

An experimental setting was defined by a unique combination of dataset, backbone architecture, training fraction, encoder adaptation strategy (frozen or unfrozen), along with the hyperparameters used (learning rates, batch sizes, etc.). Within each experimental setting, methods were ordered according to the primary evaluation metric, with Rank 1 assigned to the best-performing method, Rank 2 to the second-best method, and so on. For classification tasks, rankings were determined using the macro F1-score, whereas for regression tasks, rankings were determined using the MAE.

The reported average rank corresponds to the mean rank achieved by a method across all experimental settings. The lower average ranks indicate stronger and more consistent performance across the evaluation conditions. We also report rank standard deviation to capture performance stability.

The codebase and experimental framework are available https://github.com/parvthacker/SSL-for-Time-Series.

IV. Results

A. Regression Task

Table IV summarizes aggregate performance on the PPG-Dalia Heart Rate estimation experiments. Across the evaluated configurations, time series feature-based representations achieved equal or better performance than traditional SSL baselines in the majority of cases. In particular, TSFEL-90 and TSFEL-30 achieved the best ranks among the evaluated frameworks, and increasing the number of time series features slightly degraded performance.

The rank distribution shown in Fig. 1a shows these trends. Although several baseline SSL techniques achieve better performance in specific configurations, the time series feature-based self-supervised learning method consistently performs well across various configurations. The distributions suggest that feature-based supervision techniques maintain their performance across the evaluated optimization settings and pretraining configurations, with no increase in tuning effort.

While no method universally dominated all configurations, the time series feature-based self-supervised learning method demonstrated strong aggregate performance while requiring the least computational effort among the evaluated methods.

B. Classification Task

Table III summarizes aggregate performance on the HHAR activity recognition benchmark. Across the evaluated HHAR activity recognition experiments, time series feature-based SSL consistently ranked among the strongest performing approaches. Feature-based supervision occupied four of the five highest-ranked positions in the aggregate ranking, alongside Barlow Twins, while consistently achieving a competitive Avg Rel F1 score compared to Barlow Twins, by margins of 0.0 to 0.04. TSFEL variants achieved a consistent average rel F1 score of 0.62-0.73.

Unlike in Regression experiments, the separation between the methods is less pronounced. Traditional SSL approaches, such as Barlow Twins, achieved aggregate performance comparable to that of the time series feature-based self-supervised learning method, but showed higher standard deviation. Time series feature-based self-supervised learning methods displayed greater consistency in their performance across the

TABLE V: Effect of frozen and unfrozen fine-tuning on HHAR activity recognition performance, aggregated across all experimental configurations. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 9 (worst). Avg Rel Score (↑) is the average of relative MAE across experimental configurations from 1 (best, lowest MAE) to 0 (worst, highest MAE). *While unfrozen fine tuning substantially improves performance for weaker self-supervised methods (e.g. MPM, TS2Vec), TSFEL features remain competitive under both conditions, suggesting their advantage is due to genuinely stronger representations, not just finetuning.*

	Frozen		Unfrozen	
Method	**Rank ↓**	**Avg Rel F1 ↑**	**Rank ↓**	**Avg Rel F1 ↑**
barlow	3.28	0.79	3.68	0.67
byol	5.12	0.61	5.20	0.49
mpm	7.88	0.12	6.42	0.30
simmtm	7.10	0.34	5.75	0.45
ts2vec	7.11	0.36	7.32	0.26
tsfel15	4.86	0.64	4.64	0.60
tsfel30	3.85	0.71	4.45	0.62
tsfel45	3.29	0.72	4.85	0.58
tsfel90	3.53	0.71	3.35	0.75

TABLE VI: Effect of frozen and unfrozen fine-tuning on PPG-Dalia heart rate estimation performance, aggregated across all experimental configurations. Average Rank (↓) is the mean rank across experimental configurations, where each method is ranked from 1 (best) to 9 (worst). Avg Rel Score (↑) is the average of relative MAE (inverted) across experimental configurations from 1 (best, lowest MAE) to 0 (worst, highest MAE). *Under frozen evaluation, TSFEL30 achieves the best rank, with most conventional self-supervised methods performing notably worse. Unfrozen fine-tuning improves the ranks of TS2Vec, and performs slightly better. This suggests TSFEL features capture important time-series structure.*

	Frozen		Unfrozen	
Method	**Rank ↓**	**Avg Rel Score ↑**	**Rank ↓**	**Avg Rel Score ↑**
barlow	3.60	0.73	6.19	0.38
byol	4.46	0.65	5.91	0.43
mpm	7.03	0.32	4.28	0.65
simmtm	7.04	0.27	5.58	0.54
ts2vec	6.63	0.43	4.24	0.66
tsfel15	4.46	0.63	4.82	0.57
tsfel30	3.51	0.74	5.27	0.57
tsfel45	4.94	0.63	4.41	0.64
tsfel90	4.24	0.71	4.26	0.67

evaluated backbones and training fractions. This observation is supported by Fig. 1b, where the feature-based supervision variants remain concentrated in lower-ranking regions, while the competing method displayed broader performance distribution.

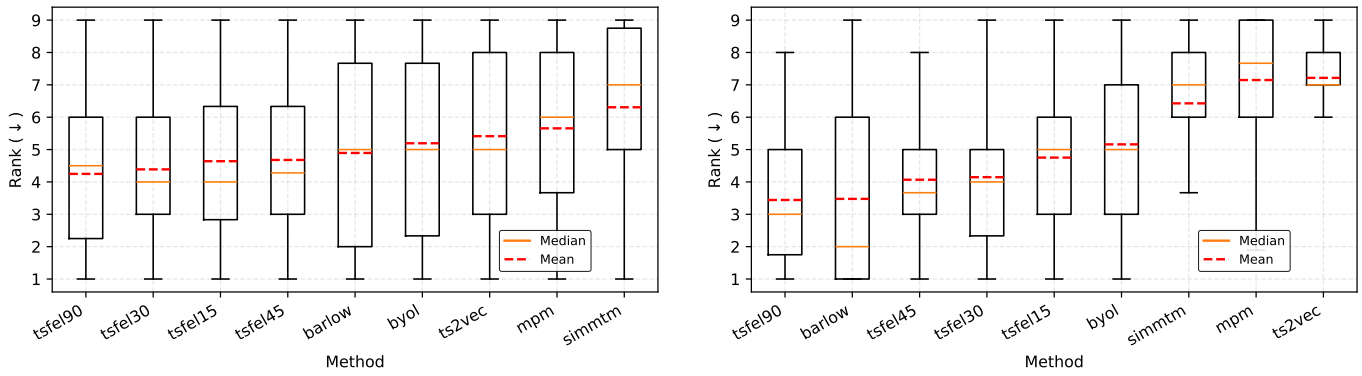
This behavior suggests that time series feature-based self-supervised learning is less dependent on hyperparameter choices and continues to perform consistently across the evaluated configurations.

C. Ablation Study

1) Effect of Finetuning Strategy: To better understand the behavior of time series representations under different downstream configurations, we evaluated all models under frozen and unfrozen finetuning regimes. Across both benchmarks, feature-guided representation learning methods show a strong performance in frozen finetuning. On classification tasks, for frozen finetuning, as shown in Table V, TSFEL45 and Barlow achieved the lowest average rank, 3.29 and 3.28, respectively, whereas the lowest achieved by other conventional SSL methods was 5.12. With finetuning, on the other hand, TSFEL90 dominated with a rank of 3.35, versus 3.68 of Barlow. Notably, the average performance of time-series feature-based self-supervised learning methods changes very little across regimes, indicating that the representations learned are highly informative from the start. In conventional SSL methods, on the other hand, we see a large jump when the encoders are finetuned, particularly with methods like SimMTM, where the average rank drops by more than 1, and avg rel F1 increases by 0.11. These results suggest that, on the evaluated benchmarks, most conventional SSL methods benefit more from encoder fine-tuning, whereas representations learned through handcrafted time series feature supervision appear to retain a larger fraction of their downstream utility prior to adaptation.

A similar trend is observed in Table VI, showing the behavior on the regression task. Under the frozen regimen, time series approaches show the strongest aggregate ranks, with TSFEL30 achieving an average rank of 3.51. After unfrozen finetuning, the performance gap narrowed, in line with observations during the classification task; MPM and SimMTM became competitive. *The strong performance under frozen evaluations and competitive performance in unfrozen cases suggest that such time series descriptors capture information that remains useful for downstream tasks on the evaluated benchmarks.*

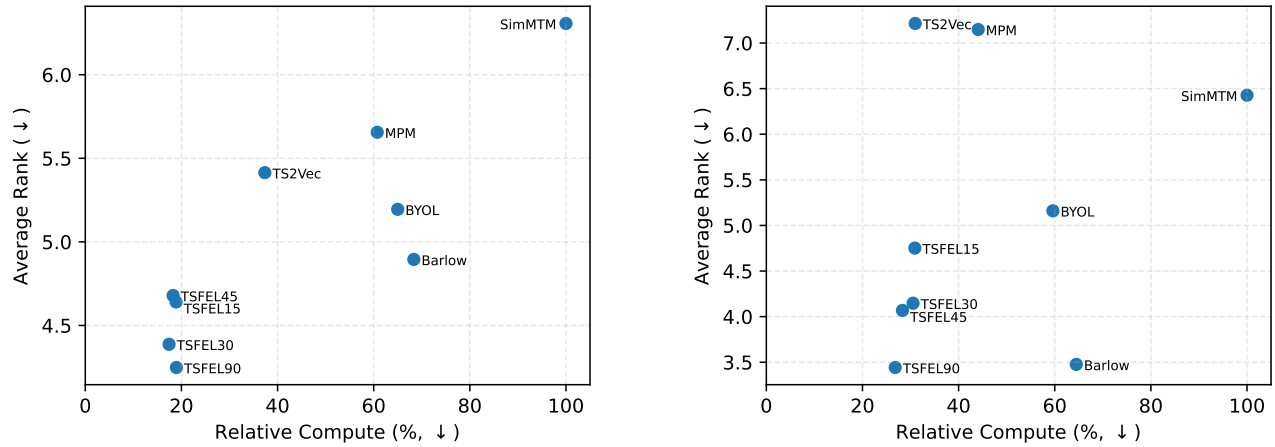
2) Effect of Backbone Architecture: We examined per-backbone-normalized ranks to assess how feature-based supervision generalizes across models. OnxPPG-Dalia, TSFEL variants ranked first across three of the four backbones versus the closest competitor, with average ranks of 2.93 vs. 4.56, 3.62 vs. 4.04, 4.32 vs. 3.83, and 3.56 vs. 5.11 across MLP, MLP-Mixer, ResNet-18, and PatchTST, respectively, indicating that the gains are generally architecture-agnostic. On HHAR, the results are more varied. Barlow significantly outperformed TSFEL on MLP-Mixer (1.7 vs. 3.4) and ResNet (1.6 vs. 2.7), suggesting that contrastive objectives are more effective for the task evaluated with MLP-Mixer. Overall, feature-based supervision is broadly robust on the evaluated datasets, but its relative advantage is strongest on regression tasks, and the advantage reduces when a baseline is aligned with the backbone.



(a) PPG-Dalia Dataset

(b) HHAR Dataset

Fig. 1: Rank distributions for (a) PPG-Dalia and (b) HHAR datasets, where each method is ranked from 1 (best) to 10 (worst) per experimental configuration. The solid and dashed lines denote the median and mean rank, respectively. *The proposed approaches cluster toward the lower end of the rank distribution on both datasets, with lower means and tighter spreads than most SSL baselines, indicating more consistent performance across experimental configurations.*



(a) PPG-Dalia Dataset

(b) HHAR Dataset

Fig. 2: Average rank (1 = best, 10 = worst) vs. relative compute cost for each method on (a) PPG-Dalia and (b) HHAR-ACT datasets. Methods in the lower-left are preferable, achieving better rank at lower computational cost.
TSFEL based methods occupy the lower-left region on both datasets, offering competitive performance at a fraction of the compute required by SSL baselines. In contrast, SimMTM incurs a high computational cost and ranks poorly across both benchmarks. While Barlow matches TSFEL performace on HHAR, it takes considerably more compute.

D. Computational Efficiency Analysis

Fig. 2a and 2b show the relationship between average normalized rank (the average rank divided by 10) and average computational effort required to get the best performance of the method, for PPG-Dalia and HHAR, respectively. The computational effort for training is proportional to the number of network passes required for training the model.

Across the evaluated benchmarks, time-series-based approaches consistently occupied the most favorable, lower left side region of the plot, corresponding to strong aggregate performance and low computational cost. In particular, for Regression, TSFEL90 and TSFEL30 achieved strong aggregate performance while requiring the least computing. This trend was followed by TSFEL90 and TSFEL45 in the classification case. Additionally, an increase in computation does not necessarily translate into better performance. In particular, SimMTM often required higher computational effort, but rarely performed competitively with time-series-based methods.

Overall, the computational efficiency analysis shows that time series feature-based self-supervised learning provides a favorable balance between performance and efficiency on the evaluated wearable sensing benchmarks.

V. GenAI Disclosure

The authors used generative AI tools (large language models) to assist with code implementation and refactoring during

the development of the experimental framework. This assistance was limited to localized contributions such as improving or generating individual functions, classes, and code snippets, and to suggesting incremental changes to existing code. Generative AI tools were also used to a limited extent for language editing of the manuscript, such as improving clarity, grammar, and phrasing in individual sentences, without altering the technical content. All AI-assisted code was reviewed, tested, and verified by the authors to ensure it met the intended specification and produced correct results, and all AI-assisted text was checked by the authors for accuracy and consistency with the reported experiments. No generative AI tools were used to design experiments, generate or analyze data, or produce the reported results. The study design, methodology, experimental analysis, and scientific conclusions are entirely the work of the authors.

VI. Limitations

Our study has several limitations that bound its conclusions. The method is not free of inductive bias, only of augmentation: it replaces contrastive transformations with a fixed set of TSFEL descriptors, and the feature-set-size ablation shows this choice is consequential, with performance following an inverted-U trend and no principled criterion for selecting the subset. The evaluation also covers only two wearable sensing benchmarks within a single modality family of short fixed-length windows, leaving forecasting, long-context dependencies, and the non-stationary clinical signals that motivate the method untested. Finally, the analysis is empirical and does not isolate the underlying mechanism, as we neither disentangle the statistical, temporal, and spectral feature families nor probe which signal properties the encoder learns, leaving the central hypothesis supported by performance but not by direct analysis.

VII. Future Work

Several directions follow from these limitations. The most immediate is to extend evaluation beyond short-window wearable sensing to forecasting, anomaly detection, and longer non-stationary clinical recordings such as polysomnography or electrocardiography, where the assumptions behind both augmentation and masked modeling are most strained. A second is to replace the manually selected descriptor sets with a principled feature-selection or feature-weighting procedure that recovers the inverted-U behavior in a data-driven manner and removes the dependence on a fixed subset size. A third is a mechanistic analysis that ablates the statistical, temporal, and spectral targets independently to test whether the gains stem from spectral structure, temporal ordering, or distributional summaries. A fourth is to investigate whether combining feature prediction with a contrastive or reconstruction term yields complementary gains, since empirical results suggest that the two paradigms encode different aspects of the signal, with augmentation-based SSL methods occasionally outperforming feature-based supervision.

VIII. Conclusion

We presented a feature-informed self-supervised learning framework that predicts handcrafted statistical, temporal, and spectral descriptors directly from the raw signal, requiring neither augmentation nor contrastive objectives. Across over 1400 experiments spanning two wearable benchmarks, four backbones, multiple label fractions, two fine-tuning regimes, and a range of hyperparameters, evaluated through a rank-based protocol that avoids best-case reporting, feature-targeted supervision consistently ranked among the strongest methods while requiring the least compute. Barlow comes in as a close second on the HHAR dataset, which leads us to conclude that, on wider testing, other methods may potentially compete or slightly outperform time series feature-based approaches, yet they come nowhere close to the computational efficiency edge of the proposed approach. The advantage was most pronounced on regression and under frozen evaluation, where the representations retained most of their downstream utility without adaptation, while on classification the method matched the best contrastive baselines with tighter rank distributions. These results indicate that time series descriptors are an effective and underexplored source of self-supervision on the evaluated benchmarks, complementary to augmentation and reconstruction based approaches rather than universally superior, and a promising basis for the extensions outlined above.

References

[1] L. Tomov, L. Chervenkov, D. G. Miteva, H. Batselova, and T. Velikova, "Applications of time series analysis in epidemiology: Literature review and our experience during covid-19 pandemic," *World Journal of Clinical Cases*, vol. 11, no. 29, pp. 6974–6983, 2023.

[2] Y. Fu, Z. Zhu, L. Liu, W. Zhan, T. He, H. Shen, J. Zhao, Y. Liu, H. Zhang, Z. Liu, Y. Xue, and Z. Ao, "Remote sensing time series analysis: A review of data and applications," *Journal of Remote Sensing*, vol. 4, p. 0285, 2024. [Online]. Available: https://spj.science.org/doi/abs/10.34133/remotesensing.0285

[3] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[4] C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022," *WIREs Data Mining and Knowledge Discovery*, vol. 14, no. 1, p. e1519, 2024. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1519

[5] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 6555–6565. [Online]. Available: https://doi.org/10.1145/3637528.3671451

[6] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Comput. Surv.*, vol. 57, no. 1, Oct. 2024. [Online]. Available: https://doi.org/10.1145/3691338

[7] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "Data labeling: An empirical investigation into industrial challenges and mitigation strategies," in *Product-Focused Software Process Improvement*, M. Morisio, M. Torchiano, and A. Jedlitschka, Eds. Cham: Springer International Publishing, 2020, pp. 202–216.

[8] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," 2024. [Online]. Available: https://arxiv.org/abs/2301.05712

[9] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] J. Zbontar, L. Jing, J. Ponce, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, pp. 12 310–12 320.

[11] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 21 271–21 284.

[12] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira, "Benchmarking self-supervised learning on diverse pathology datasets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 3344–3354.

[13] Y. Wang, H. Wu, J. Dong, Y. Liu, C. Wang, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2026.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[15] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ser. ICMI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 216–220. [Online]. Available: https://doi.org/10.1145/3136755.3136817

[16] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLOS ONE*, vol. 16, no. 7, pp. 1–32, 07 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0254841

[17] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, "What should not be contrastive in contrastive learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=CZ8Y3NzuVzO

[18] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "Tsfel: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711020300017

[19] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218304843

[20] H. Blunck, S. Bhattacharya, T. Prentow, M. Kjærgaard, and A. Dey, "Heterogeneity Activity Recognition," UCI Machine Learning Repository, 2015, dOI: 10.24432/C5689X. [Online]. Available: https://doi.org/10.24432/C5689X

[21] A. Reiss, I. Indlekofer, and P. Schmidt, "PPG-DaLiA," UCI Machine Learning Repository, 2019, dOI: 10.24432/C53890. [Online]. Available: https://doi.org/10.24432/C53890

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. P. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-mixer: An all-MLP architecture for vision," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=EI2KOXKdnP

[24] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=Jbdc0vTOcol

[25] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *ECCV*, 2016.

[26] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," 2016. [Online]. Available: https://boris-portal.unibe.ch/handle/20.500.12422/151635

[27] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 14 668–14 678.

[28] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 1298–1312.

[29] L. Ericsson, H. Gouk, and T. Hospedales, "Why do self-supervised models transfer? on the impact of invariance on downstream tasks," 2022. [Online]. Available: https://arxiv.org/abs/2111.11398

[30] H. Haresamudram, I. Essa, and T. Plötz, "Assessing the state of self-supervised human activity recognition using wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 3, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3550299

[31] A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, "Deep ppg: Large-scale heart rate estimation with convolutional neural networks," *Sensors*, vol. 19, no. 14, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/14/3079

[32] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8980–8987.

[33] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long, "Simmtm: A simple pre-training framework for masked time-series modeling," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.