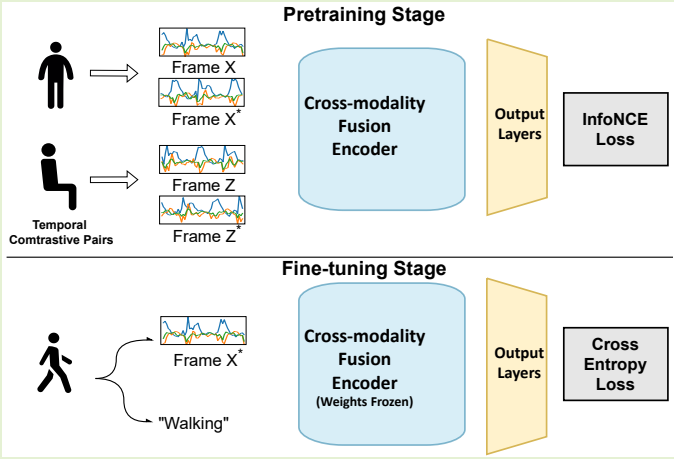
Temporal Contrastive Learning for Sensor-Based Human Activity Recognition: A Self-Supervised Approach

Xiaobing Chen, Xiangwei Zhou, Mingxuan Sun, and Hao Wang

***Abstract*— Deep learning techniques can make use of a large amount of time-series data from wearable devices and greatly benefit the development of sensor-based human activity recognition (HAR). However, representation learning in a supervised manner requires massive labeled sensory data that are time-consuming to obtain and hindered by privacy concerns. To address these issues, we utilize the plentiful unlabeled sensory data and propose a novel self-supervised learning framework, namely Temporal Contrastive Learning in Human Activity Recognition (TCLHAR), which learns the meaningful feature representations for time-series data without labels. Our TCLHAR framework utilizes the temporal co-occurrence relationship among time windows as the supervisory signals to construct positive pairs in the encoder pretraining stage. The encoder is designed for cross-modality fusion, which leverages the local interactions of each sensor modality and the global fusion of features from different sensors. The proposed framework is extensively evaluated on public HAR datasets in supervised, self-supervised, and semi-supervised settings. Our method outperforms several self-supervised learning benchmark models, achieving comparable results with fully labeled data training. When labeled data is scarce, our method can boost the F1 score by up to 65% over traditional supervised training, which demonstrates the effectiveness of our feature representations.**



***Index Terms*— Human activity recognition, self-supervised learning, contrastive learning, representation learning.**

I. Introduction

HUMAN activity recognition (HAR) is a general task to identify the performed activities of a person based on sensory data, such as videos, visual images, and measurements from inertial measurement units. With the massive implementation of multi-modality sensors in portable and wearable devices, collecting sensory data is becoming easier, fueling the growth of applications such as health monitoring, assistive technology, and sports analytics. Integrating multiple modalities of sensory data is thus a promising way to accomplish the task. Generally, sensor-based HAR methods adopt the time-series data from various sensors and combine signal processing and machine learning techniques to classify the inputs into different activities in a pre-defined infinite activity set.

With the rapid development of deep learning techniques, many supervised deep learning models have been successfully applied to HAR [1], [2], [3]. These end-to-end models outperform the traditional HAR methods by replacing the hand-crafted features with learned features. By leveraging the power of deep neural networks, these models are capable of detecting complex patterns and subtleties in the sensory data, resulting in more precise and adaptive activity recognition. However, collecting labeled data and applying a supervised deep learning model to HAR is challenging due to the demands of the giant training data. Data annotation, which requires semantic segmentation of time-series sequences, is expensive and time-consuming, given the enormous data volume. This challenge is exacerbated in real-world applications where activities can vary significantly among individuals and contexts, demanding high granularity in annotations. The heterogeneity of devices, sensor types, and device-working environments makes it non-trivial to unify the types and quantities of data collected from different sources and thus produces large-scale datasets. This heterogeneity poses interoperability challenges, leading to potential inconsistencies and biases in the data, which can, in

This paper was supported in part by the National Science Foundation under Grant No. 1943486, 2246757, 2315612, 2332011, 2403247, 2431595, and the Office of Research and Economic Development at Louisiana State University. *(Corresponding author: Xiangwei Zhou.)*

X. Chen and X. Zhou are with the Division of Electrical and Computer Engineering at Louisiana State University, Baton Rouge, LA, 70803 (e-mail: {xchen87, xwzhou}@lsu.edu).

M. Sun is with the Division of Computer Science and Engineering at Louisiana State University, Baton Rouge, LA, 70803 (e-mail: msun11@lsu.edu).

H.Wang is with the Department of Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, NJ, 07030 (e-mail: hwang9@stevens.edu).

turn, affect the overall performance of the model. Furthermore, data collection inevitably raises privacy issues since HAR data samples are mainly generated from personal devices, such as smartphones and watches, and owned by consumers. The sensitive nature of the data, reflecting individuals' behaviors, locations, and routines, calls for robust ethical guidelines and privacy-preserving mechanisms to ensure that personal information is handled with utmost care and confidentiality.

Unlike labeled data, which are hard to collect, unlabeled data are readily available. Self-supervised learning is a promising technique to address the aforementioned challenges by utilizing unlabeled data. The prevalence of unlabeled data, particularly in domains where manual annotation is complex or privacy-sensitive, makes self-supervised learning a powerful alternative to traditional supervised learning. Instead of constructing an end-to-end model, self-supervised learning decomposes the model into an encoder and a classifier head. The encoder is pretrained with unlabeled data in a pretext task to generate meaningful feature representations. Then, the encoder is frozen, and a trainable classifier head is added to the top of the encoder. The whole model is fine-tuned in a supervised manner with a small amount of labeled data. This two-stage process enables the model to first capture generalizable features from the large-scale unlabeled data and then specialize its learning to the specific task using limited labeled data. Self-supervised learning has been widely applied in computer vision (CV) and natural language processing (NLP) tasks [4], [5], [6], [7], greatly reducing the performance gap with supervised methods. Its success in these domains demonstrates its potential to build robust models in scenarios with limited labeled resources.

Inspired by the image enhancement in CV, a few recent studies apply augmentation techniques to the time-series data and use the transformed data as the supervisory signals [8], [9], [10], [11], [12], [13]. For instance, multi-task SSL [8] applies a set of signal transformations to time-series inputs and feeds the original signals with the transformed signals into the model. A group of binary classification tasks are performed to train the encoder in the pretraining stage. This demonstrates the effectiveness of self-supervised learning for HAR. These studies have paved the way for understanding how to leverage the richness of unlabeled data in a domain characterized by temporal dynamics, multisensory inputs, and complex human behaviors. However, unlike static images or discrete text data, time-series data in HAR have intrinsic temporal dependencies and contextual nuances that demand specialized treatment, which lacks explorations from the above work. Moreover, in the field of CV, it has been empirically shown that semantic information of original data is invariant to some augmentation methods [7], [14]. However, it is unclear whether augmentations on time-series data would preserve or undermine the semantic information.

Additional work attempts to exploit the temporal relationship among the sequence of timestamp signals [15], [9]. However, these studies exclusively concentrate on timestamp-level prediction, predicting future timestamps based on the preceding ones. While being valuable in capturing short-term dependencies and immediate temporal patterns, this approach may miss the broader context of activities that are defined by more extended and intricate temporal dynamics.

To address the limitations of traditional methods with timestamp-level predictions, we propose TCLHAR, a novel self-supervised learning framework that takes advantage of the temporal adjacency among time windows to generate better feature representations for time-series data. The design of TCLHAR is motivated by the need to capture temporal dependencies in time-series data and leverage contrastive learning for self-supervised representation learning. Human activities are sequential, and using temporal co-occurrence relationships between time windows helps capture these dynamics more effectively. Contrastive learning, via the InfoNCE loss, maximizes the similarity between positive pairs (adjacent time windows) while minimizing it for negative pairs, ensuring that learned representations are robust and generalizable.

The main contributions of this paper include:

1) We propose a novel self-supervised learning framework, namely Temporal Contrastive Learning in Human Activity Recognition (TCLHAR). To our knowledge, it is the first framework in HAR utilizing the temporal co-occurrence relationship among adjacent time windows to construct meaningful positive pairs. By extending the focus beyond individual timestamps, this framework aims to encapsulate a more nuanced understanding of activities as dynamic processes evolve over time.
2) We introduce a cross-modality fusion encoder based on a convolutional neural network (CNN) that combines cross-channel interactions of each sensor modality with a global fusion of features from different sensors. The encoder is composed of individual subnets and a merging subnet. The individual subnets independently extract features from each sensor, and the merging subnet is utilized for feature fusion. This architecture allows the model to analyze each sensor modality's distinct characteristics and leverage the correlations between different modalities to synthesize a more robust feature set, enhancing the overall performance.
3) We conduct extensive experiments on three public HAR datasets, including MobiAct [16], UCI-HAR [17], and USC-HAD [18], to compare our framework with several self-supervised learning benchmarks models on HAR. We also conduct ablation studies on the impact of multimodal fusion and the encoder pretraining task.

The paper is organized as follows. Section II provides an overview of existing representation learning and self-supervised learning studies for HAR. In Section III, we introduce our proposed self-supervised representation learning framework, i.e., TCLHAR. We present in Section IV our experiments, evaluating our framework on publicly available datasets. Finally, we conclude the paper and point out future research directions in Section V.

II. Related Work

In this paper, we address the meaningful representation learning for HAR, utilizing the temporal contrastive learning technique. In what follows, we introduce some important existing work in this field.

A. Representation Learning

Feature extraction and feature engineering play a critical role in HAR. Existing feature extraction techniques mainly involve hand-crafted feature design and representation learning. Hand-crafted feature design requires expert knowledge and uses heuristic features, such as mean and standard deviation in the time domain and frequency features extracted via Fourier transformation [3], [19]. These traditional methods provide a controllable and understandable way to create features, but they often rely heavily on domain expertise and can be laborious to develop. Inherent limitations of this technique include the dependency on data modality and data structure, requirements of task-specific knowledge and feature extraction, and relatively poor performance.

In contrast, representation learning takes advantage of deep learning networks to automatically learn the representations of raw data in a supervised or unsupervised manner. Supervised learning models for HAR have been widely developed and achieve promising performance [1], [3], [20], [21]. They excel in capturing complex patterns and relations in data, enabling more robust and accurate activity recognition. Existing unsupervised learning models use autoencoder networks to extract features, such as restricted Boltzmann machines (RBM) [22] and CNN [23]. Utilizing unlabeled data offers a cost-effective and scalable solution, opening new avenues for leveraging vast amounts of unannotated sensory data. These unsupervised learning models use unlabeled data to pretrain the encoder. The learned features from the frozen encoder become inputs of the classifier in the downstream task, such as activity recognition.

Recently, self-supervised learning, a branch of the unsupervised learning technique, has raised a lot of interest and shown great success in multiple fields such as CV [24], [25], [26] and NLP [27], [28], [29]. Self-supervised learning is a technique extracting meaningful features from unlabeled raw data by pretraining the encoder in a certain pretext task that utilizes the inherent data relationship as the supervisory signals. The design of the pretext task is of the utmost importance, which decides the learned knowledge of the raw data and profoundly affects the performance of downstream tasks. Some examples of pretext tasks include predicting rotation of images [24], predicting relative positions of image patches [25], and reconstructing masked patches [26] in CV, as well as predicting next words [27], [28] and reconstructing masked words [29] in NLP.

B. Self-supervised Learning in HAR

Recently, a few studies have introduced self-supervised learning to HAR and demonstrated its promising performance [23], [10], [30], [12]. In [23], the impacts of different feature representation techniques, including statistical features, distribution-based representation, and several autoencoder-based unsupervised representations, on the classification performance have been compared. It shows that autoencoder-based methods achieve comparable performance with the supervised learning model. In [8], a multi-task self-supervised model has been proposed to learn sensor representations and predict human activities. Multiple independent classification heads are used to predict which transformation is applied to the inputs. Similarly, a recent study named CSSHAR [10] applies different transformations to the raw data and constructs the positive pair with two samples transformed from the same data and the negative pair with samples from different data. A transformer encoder is trained to make the positive samples close and negative ones apart in the latent space.

To address the issue of false negative pairs, CluterCLHAR [12] is proposed, which clusters the representations of samples and only constructs negative pairs from different clusters. Similarly, a dynamic temperature scaling method is proposed in [13] to alleviate the problem of false negative pairs. These methodologies showcase novel strategies to refine the training process, enhancing the robustness and precision of the learned models.

Although the aforementioned studies show the feasibility of rendering decent performance by applying self-supervised learning to time-series data, they omit the temporal relationship among time-series data since the temporal dynamics of human activities can offer rich information for learning. In [15], the mask reconstruction is adopted as the pretext task similar to the BERT model [29], which randomly masks out several timestamp readings of the raw data and pretrains the encoder to reconstruct the masked-out positions. Instead of predicting the timestamps in the original data space, the contrastive predictive coding (CPC) framework is introduced to HAR in [9]. The CPC first inputs a sequence of timestamp signals in a time window, encodes the raw data at each timestamp to the latent space, and then uses the recurrent neural network (RNN) to generate a context vector that is utilized to predict the future timestamps in the latent space.

Unlike CPC using timestamp-level prediction in one time window, our self-supervised learning framework named TCLHAR improves the performance of temporal contrastive learning in HAR from two perspectives. *First*, instead of inputting one time window, our framework accepts multiple time windows as inputs and encodes them to conduct time-window level contrastive learning. This is because different activities are more distinct with respect to their timestamp sequences rather than just certain timestamps. This novel approach allows the model to capture intricate temporal patterns and variations, leading to a more nuanced understanding of human activities. *Second*, regarding the selection of time windows, we perform temporally adjacent sampling to select positive pairs of time windows and construct negative pairs like SimCLR [5]. This design helps to learn meaningful representations and increases the number of positive and negative pairs beneficial to the diversity of the inputs. By increasing the diversity and complexity of the input samples, this technique enriches the learning process and amplifies the ability to distinguish between different activities.

III. Method

This section introduces TCLHAR, our proposed self-supervised learning framework that fully utilizes unlabeled time-series data for HAR. Unlike commonly-used end-to-end learning schemes that utilize labeled data to train the

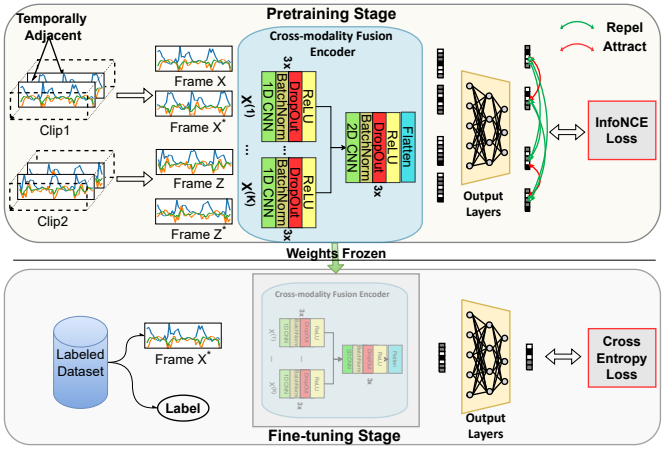


Fig. 1: The overview of our TCLHAR framework. The cross-modality fusion encoder is trained in the pretraining stage and applied in the fine-tuning stage.

encoder and classifier simultaneously, TCLHAR decomposes the learning process into two stages, the *pretraining* and *fine-tuning* stages. The goal of the pretraining stage is to train an encoder that generates meaningful feature representations of inputs in a pre-defined pretext task with unlabeled data. In the fine-tuning stage, we transfer and freeze the weights of the encoder from the pretraining stage. Then we train a classifier, such as a multi-layer perceptron (MLP), following the encoder with labeled data to perform activity recognition. Rather than training the whole model in a fully supervised manner, TCLHAR only requires a smaller amount of labeled data because the encoder is well trained, and high-quality representations are obtained in the pretraining stage.

Figure 1 presents the overview of the TCLHAR framework. There are four major components in the pretraining stage: a contrastive loss function, temporally adjacent sampling, a cross-modality fusion encoder, and neural network based output layers, which will be introduced in Sections III-A, III-B III-C. The procedure of the fine-tuning is briefly introduced in Section III-D.

For the rest of this paper, vectors are represented by bold lower-case letters (e.g., $\mathbf{x}$ and $\mathbf{y}$), while bold upper-case letters, such as $\mathbf{X}$ and $\mathbf{Y}$, denote matrices and arrays. For a vector $\mathbf{x}$, the i-th element is represented by x_i. We use calligraphic letters to denote sets such as $\mathcal{X}$ and $\mathcal{Y}$, while $|\mathcal{X}|$ denotes the cardinality of $\mathcal{X}$.

A. Contrastive Loss and Mutual Information

We select InfoNCE loss [31] as our objective function in the pretraining stage, which has shown success in other self-supervised learning frameworks [5], [6]. We choose the InfoNCE loss function also because of its strong theoretical foundations in maximizing mutual information between positive pairs of samples while ensuring that negative pairs remain dissimilar. In the context of time-series data for HAR, this is particularly effective because temporally adjacent frames often contain shared activity information, and the goal is to learn a representation that captures this co-occurrence.

Let $\mathbf{x} \in \mathbb{R}^{T \times S}$ represent the input sensor data, where T denotes the number of time steps and S is the dimensionality of the sensor data. Then, N positive pairs are the observations following the joint distribution $\left(\{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^{N}\right) \sim p\left(\mathbf{x}, \mathbf{x}'\right)$, such as different channels of an image [32] and different augmentations of a signal [10] in practice, while negative pairs are sampled from the marginal distributions, i.e., $\left(\{\mathbf{x}_i, \mathbf{x}'_j\}_{i \neq j}\right) \sim p(\mathbf{x})p(\mathbf{x}')$.

The corresponding representation of the input $\mathbf{x}_i$ through the encoder $f(\cdot)$ is $\mathbf{h}_i = f(\mathbf{x}_i)$, and the outputs of output layers $g(\cdot)$ are $\mathbf{z}_i = g(\mathbf{h}_i)$. The InfoNCE loss of the positive pair $(\mathbf{z}_n, \mathbf{z}'_n)$ is given by

$$\mathcal{L}_n(\mathbf{z}_n, \mathbf{z}'_n) = -\log \frac{\exp\left(\frac{sim(\mathbf{z}_n, \mathbf{z}'_n)}{\tau}\right)}{\sum_{j=1}^{2N} \mathbf{1}_{[j \neq n]} \exp\left(\frac{sim(\mathbf{z}_n, \mathbf{z}'_j)}{\tau}\right)}, \quad (1)$$

where $\mathbf{1}_{[.]}$ is an indicator function to compute the loss of the negative pairs, τ is the temperature parameter controlling the degree of discrepancy among similarity scores, and $sim(\mathbf{u}, \mathbf{v})$ is the similarity score evaluated by the cosine distance and defined as

$$sim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{||\mathbf{u}||_2 ||\mathbf{v}||_2}. \quad (2)$$

Therefore, the total loss of N input pairs is given by

$$\mathcal{L}_{\text{NCE}} = \frac{1}{2N} \sum_{n=1}^{N} (\mathcal{L}_n(\mathbf{z}_n, \mathbf{z}'_n) + \mathcal{L}_n(\mathbf{z}'_n, \mathbf{z}_n)). \quad (3)$$

If we define

$$h_\theta(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{sim(g(f(\mathbf{x})), g(f(\mathbf{x}')))}{\tau}\right), \quad (4)$$

the loss can be simplified as

$$\mathcal{L}_n(\mathbf{x}_n, \mathbf{x}'_n) = -\log \frac{h_\theta(\mathbf{x}_n, \mathbf{x}'_n)}{\sum_{j=1}^{2N} \mathbf{1}_{[j \neq n]} h_\theta(\mathbf{x}_n, \mathbf{x}'_j)}. \quad (5)$$

By minimizing the InfoNCE loss, we obtain the optimal value of $h_\theta(\mathbf{x}, \mathbf{x}')$, which is proportional to the density ratio between the joint distribution $p(\mathbf{x}, \mathbf{x}')$ and the product of marginal distributions $p(\mathbf{x})p(\mathbf{x}')$, where θ represents the trainable weights of encoder and output layers [33], [31]. The relationship is given by:

$$h_\theta(\mathbf{x}, \mathbf{x}') \propto \frac{p(\mathbf{x}, \mathbf{x}')}{p(\mathbf{x})p(\mathbf{x}')}, \quad (6)$$

which connects $\mathcal{L}_{\text{NCE}}$ with mutual information of input variables $I(\mathbf{x}; \mathbf{x}')$. Specifically,

$$I(\mathbf{x}; \mathbf{x}') \geq \log(2N) - \mathcal{L}_{\text{NCE}}. \quad (7)$$

The proof of the inequality can be found in [31]. This indicates that minimizing the InfoNCE loss gives a tighter lower bound on the mutual information $I(\mathbf{x}; \mathbf{x}')$. Moreover, it is also beneficial to increase the number of negative pairs and thus to increase the amount of shared information between two random variables, which will be empirically demonstrated in our ablation experiments.

B. InfoMin Principle and Temporally Adjacent Sampling

In contrastive learning, we create positive pairs (e.g., two related signals) and negative pairs and learn an encoder such that features of positive pairs are similar to each other and features of negative pairs are distant from each other. Usually, two signals with a spatial, temporal, or transformed relationship can form positive pairs.

Given the relationship between the InfoNCE loss and the mutual information of inputs $I(\mathbf{x};\mathbf{x}')$ shown in (7), the amount of information shared by $\mathbf{x}$ and $\mathbf{x}'$ decides the lower bound of the InfoNCE loss. Therefore, how to construct the positive pairs profoundly affects the effectiveness of representations and the performance in the downstream task. Furthermore, the optimal positive pair should satisfy the InfoMin principle introduced in [32]. Specifically, optimal positive pairs are supposed to share minimal information necessary for the downstream task. Formally, a proposition is given as follows.

Proposition 1 (InfoMin Principle): Suppose the representations and outputs in the contrastive learning are lossless in terms of the mutual information, i.e., $I(\mathbf{x};\mathbf{x}') = I(\mathbf{h};\mathbf{h}') = I(\mathbf{z};\mathbf{z}')$. *Given a downstream task* $\mathcal{T}$ *with label* $\mathbf{y}$*, the optimal positive pairs of the original data* $\tilde{\mathbf{x}}$ *for task* $\mathcal{T}$ *are* $(\mathbf{x}^*,\mathbf{x}'^*) = arg\min_{\mathbf{x},\mathbf{x}'} I(\mathbf{x};\mathbf{x}')$*, subject to* $I(\mathbf{x};\mathbf{y}) = I(\mathbf{x}';\mathbf{y}) = I(\tilde{\mathbf{x}};\mathbf{y})$.

This proposition succeeds in analyzing the effectiveness of augmentations on images, where strong and task-relevant augmentations reduce mutual information and improve the performance of the downstream task. However, there is limited literature on the augmentations on time series [34], [35], [36], and whether those augmentations undermine the task-relevant mutual information is unknown, i.e., $I(\mathbf{x};\mathbf{y}) = I(\mathbf{x}';\mathbf{y}) = I(\tilde{\mathbf{x}};\mathbf{y})$ may not stand.

Therefore, unlike existing studies, such as [10], that apply different augmentations to the same input to construct positive and negative pairs, we adopt temporally adjacent sampling and propose a temporal contrastive learning (TCL) algorithm for the pretraining stage, as shown in Algorithm 1.

Specifically, assume there are K different sensor modalities, such as accelerometers and gyroscopes. The k-th sensor has S_k channels. $S = \sum_{k=1}^{K} S_k$ denotes the total number of channels. The input signal vector at timestamp t, consisting of signal samples from S channels, is $\tilde{x}_t = \left[\tilde{x}_t^{(1)}, \tilde{x}_t^{(2)}, ..., \tilde{x}_t^{(S)}\right]^T$, where $(\cdot)^T$ denotes the transpose operator. Then, we have the unlabeled sensory readings $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_U\}$ for the self-supervised training.

In Algorithm 1, we first segment the raw data $\mathbf{D}_i$ into a series of T-length frames with 50% overlapping, where T is the number of timestamps. As a result, a frame can be denoted by a $T \times S$ matrix $\tilde{\mathbf{x}}_i = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_T]^T$. We apply temporally adjacent sampling to those frames to construct positive and negative pairs. Specifically, we bundle M temporally consecutive frames into one clip $\{\tilde{\mathbf{x}}_i\}_{i=1}^{M}$ with shape $M \times T \times S$ and suppose we have N clips. Since M frames in one clip are temporally adjacent, M is the parameter deciding the range of similarity. We randomly sample two frames $\mathbf{x}_n$ and $\mathbf{x}'_n$ from the n-th clip as a positive pair. To construct the negative sample for frame $\mathbf{x}_n$, a frame is randomly selected from the remaining $(M-1)$ clips. Then, with the constructed positive and negative pairs, the InfoNCE loss is calculated by (1) and back propagated to update the model parameters.

Algorithm 1: Temporal Contrastive Learning (TCL)

Input: unlabeled signals $\mathcal{D} = \{\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_U\}$; encoder $f(\cdot)$; output layers $g(\cdot)$; batch size N; frame length T; clip size M

Output: encoder network $f(\cdot)$

define $\mathcal{L}_n(\mathbf{z}_n, \mathbf{z}'_n) = -\log \frac{\exp\left(\frac{sim(\mathbf{z}_n,\mathbf{z}'_n)}{\tau}\right)}{\sum_{j=1}^{2N} \mathbf{1}_{[j\neq n]} \exp\left(\frac{sim(\mathbf{z}_n,\mathbf{z}'_j)}{\tau}\right)}$,
where $sim(\mathbf{u},\mathbf{v}) = \frac{\mathbf{u}^T\mathbf{v}}{||\mathbf{u}||_2||\mathbf{v}||_2}$;

for $\mathbf{D}_i \in \mathcal{D}$ **do**
  Segment $\mathbf{D}_i$ along channels into T-length frames $\{\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_M, ...\}$;
  Bundle every consecutive M frames into one clip $\{\tilde{\mathbf{x}}_i\}_{i=1}^{M}$;
end

for $n \in \{1, 2, ..., N\}$ **do**
  Sample adjacent frames $\mathbf{x}_n$ and $\mathbf{x}'_n$ from clip $\mathbf{C}_n$, $(\mathbf{x}_n, \mathbf{x}'_n) \sim p(\mathbf{x}, \mathbf{x}')$;
  $\mathbf{z}_n = g(f(\mathbf{x}_n))$;
  $\mathbf{z}'_n = g(f(\mathbf{x}'_n))$;
  $\mathcal{L}_{\text{NCE}} \leftarrow \frac{1}{2N}\sum_{n=1}^{N}(\mathcal{L}_n(\mathbf{z}_n, \mathbf{z}'_n) + \mathcal{L}_n(\mathbf{z}'_n, \mathbf{z}_n))$;
  Minimize $\mathcal{L}_{\text{NCE}}$ to update $f(\cdot)$ and $g(\cdot)$;
end

Since $\mathbf{x}_n$ and $\mathbf{x}'_n$ are sampled from the original data and no augmentation is applied, $I(\mathbf{x};\mathbf{y}) = I(\mathbf{x}';\mathbf{y}) = I(\tilde{\mathbf{x}};\mathbf{y})$ is guaranteed. Moreover, the downstream task, namely human activity recognition, requires the model to correctly recognize the activities with respect to frame data with various starting timestamps. In that sense, our positive pairs throw out the trivial temporal information and reduce $I(\mathbf{x};\mathbf{x}')$, while preserving all the information needed in the downstream task. As a result, Algorithm 1 takes advantage of the temporal co-occurrence relationship among neighboring frames of time-series data and makes a pair of temporally adjacent positive frames close in the embedding space and a pair of negative ones far away.

C. Cross-Modality Fusion Encoder and Output Layers

Encoders are to convert inputs to meaningful feature representations and shared in two stages. To achieve better cross-modality fusion for HAR applications that usually involve multi-modality sensors, our encoder consists of individual subnets for input denoted by $\mathbf{x}_n^{(k)}$ of each sensor modality with shape $T \times S_k$, and a single merging subnet for outputs of K individual subnets. Individual subnets are used to extract local features along each sensory channel, while the merging subnet is to merge the channel-wise features and extract the cross-modality features.

Specifically, an individual subnet consists of three convolution layers with 1D filters of kernel size 3 and stride 1. Each convolution layer is followed by a batch normalization layer, a ReLU activation layer, and a dropout layer. We apply "SAME" padding in the convolution layers to keep the first dimension of the input unchanged. Through each individual subnet, the

output is of shape $T \times C$, where C is the number of kernels of the last convolution layer.

The outputs of individual subnets are concatenated, so the input of the merging subnet is of shape $K \times T \times C$. To extract the cross-modality features, we adopt three 2D convolution layers with $(K, 3)$ filters followed by batch normalization and dropout layers. The output of the merging subnet is flattened before it is passed to the output layers.

As a result, the encoder $f(\cdot)$ maps the initial time-series signal $\mathbf{x}_n$ and $\mathbf{x}'_n$ into embeddings $\mathbf{h}_n = f(\mathbf{x}_n)$ and $\mathbf{h}'_n = f(\mathbf{x}'_n)$, where $\mathbf{h}_n, \mathbf{h}'_n \in \mathbb{R}^d$.

The pretraining and fine-tuning stages both have output layers. However, output layers behave differently in two stages. Output layers in the pretraining stage are used to reduce the dimensions of the features for efficiently computing the contrastive loss, while the output layers in the fine-tuning stage are trained to predict the activity classes. An MLP with one hidden layer is used in both stages to obtain $\mathbf{z}_n = g(\mathbf{h}_n) = W^{(2)} ReLU(W^{(1)} \mathbf{h}_n)$, where $ReLU$ is the activation function.

D. Fine-tuning

The fine-tuning stage is application-specific and requires a relatively small amount of labeled data for supervised learning. Suppose we have a set of labeled data for HAR $\mathcal{X} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_N\}$ with labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, where $(\tilde{\mathbf{x}}_i, \mathbf{y}_i)$ is a tuple of frame data and the corresponding label and $\mathbf{y}_i \in \mathbb{R}^E$ is the one-hot vector denoting the ground truth in E activities.

After being pretrained with unlabeled data, the encoder is frozen and reused in the fine-tuning stage. In the fine-tuning stage, a new MLP $g'(\cdot)$ is added to the encoder and trained with available labeled data in a supervised manner. Given an input $\mathbf{X}_i$, the prediction can be given as

$$\hat{\mathbf{y}}_i = \sigma(g'(\mathbf{h}_i)), \tag{8}$$

where $\mathbf{h}_i = f(\tilde{\mathbf{x}}_i)$ is the feature representation of $\tilde{\mathbf{x}}_i$ and $\sigma(\cdot)$ is the softmax function [37].

Finally, the weights of the MLP are updated by the cross entropy loss function as

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{B} \sum_{j=1}^{E} \mathbf{1}_{[y_i^{(j)}=1]} \log(\hat{y_i}^{(j)}), \tag{9}$$

where B is the training batch size.

E. Computational Cost and Model Deployment

For the HAR task, the model will be deployed in real-world wearable devices that typically have limited computational resources and batteries. Thus, in this section, we discuss the computational cost and the model deployment of our TCLHAR framework.

Training the TCLHAR model involves two stages: pretraining with unlabeled data and fine-tuning with labeled data. The pretraining stage is computationally intensive but can be performed in the central server with powerful computational resources. Fine-tuning requires significantly fewer computational resources and can be adapted to run on standard hardware. Our framework processes data in real time with minimal latency, making it suitable for real-time human activity recognition.

Our TCLHAR framework includes a cross-modality fusion encoder consisting of individual subnets for each sensor modality and a merging subnet. The encoder utilizes three 1D convolution layers in each individual subnet and three 2D convolution layers in the merging subnet, which are computationally efficient compared to more complex architectures. The model size is relatively small, ensuring that it can be stored and executed on devices with limited storage capacity. Additionally, the modular design of TCLHAR allows for easy adaptation to different sensor modalities and configurations, ensuring scalability and customization for various deployment environments.

TABLE I: Experimental dataset comparison. Acc. and Gyro. denote accelerometer and gyroscope, respectively.

Dataset	# of Subjects	# of Activities	Sampling Rate	Sensor Placement	Sensor Modality
MobiAct	61	11	200	pocket	Acc. & Gyro.
UCI-HAR	30	6	50	waist	
USC-HAD	14	12	100	right hip	

IV. Experiments

Following the common evaluation protocol in the self-supervised learning [10], [23], we mainly evaluate the effectiveness and superiority of our TCLHAR framework by freezing the parameters of the encoder trained in the pretraining stage and training a simple classifier on top of it. We conduct experiments in supervised, self-supervised, and semi-supervised settings.

A. Datasets

We conduct several experiments on the public HAR datasets, where each measurement data is from an accelerometer and a gyroscope, i.e., $K = 2$ and $S = 6$. There are several reasons to adopt these two sensor modalities. First, these sensors are widely installed in various portable devices, such as smartphones and wearable devices. The sensory data are easily and cheaply accessible, even in real-time scenarios. Moreover, these sensors are representative and proved to accomplish the HAR task well in existing studies [10]. It is worth mentioning that our TCLHAR framework is independent of the sensor type and can be easily generalized to other time-series data.

Datasets in the experiments include MobiAct [16], UCI-HAR [17], and USC-HAD [18]. They encompass a range of sensor placements, sensor types, noise environments, subjects, activity types, and experimental protocols with smartphones and wearable devices. The comparison of the datasets is given in Table I.

MobiAct. The MobiAct dataset [16] comprises data collected from the inertial sensors of a Samsung Galaxy S3 smartphone, including an accelerometer, a gyroscope, and an orientation sensor, sampled at 200 Hz. The random orientation and placement in a trouser pocket simulate daily usage, introducing natural variability and noise. The dataset contains 4

types of falls and 12 different daily activities from 66 subjects of varying ages, weights, and heights. We use 11 activities of daily life performed by 61 subjects in our experiments, including standing, jogging, walking, jumping, stairs up, stairs down, stand to sit, sitting on chair, sit to stand, car step in and out.

UCI-HAR. The UCI-HAR dataset [17] consists of sensory data from 30 subjects, collected using a waist-mounted Samsung Galaxy S2 smartphone equipped with an accelerometer and a gyroscope, sampled at 50 Hz. The sensor signals are preprocessed by applying noise filters to reduce environmental noise. Six different activities are performed: standing, sitting, laying down, walking, stairs up and down.

USC-HAD. The USC-HAD dataset [18] comprises measurements from an inertial measurement unit (IMU) that includes a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer, sampled at 100 Hz. The IMU device is worn at the front right hip and transmits data via a wired connection, minimizing wireless transmission noise and providing high-fidelity motion data. The raw data are generated from 14 subjects performing 12 different activities, including walking forward, walking left and right, stairs up, stairs down, running, jumping, sitting, standing, sleeping, elevator up and down.

For a rigorous comparison, we employ similar data preprocessing techniques as in [10]. Specifically, we first downsample the raw time-series data to 50 Hz, which is the lowest sampling rate among those for the experimental datasets. The downsampled data are segmented into 50% overlapping frames with a 1-second length. That renders the frame matrix X of shape 50×6. All frames are partitioned into training, validation, and testing splits based on the subjects, resulting in the data of a subject appearing in one split only. For all datasets, we allocate 20% of the subjects' data to the testing split, 20% of the remaining subjects' data to the validation split, and use the rest for the training split.

B. Experiment Setup

To evaluate the performance and effectiveness of our TCLHAR framework, we conduct several experiments, including downstream activity recognition, supervised learning, and semi-supervised learning, to compare ours with several self-supervised learning models. In the downstream activity recognition, encoders are trained with unlabeled data in the pretraining stage and frozen in the fine-tuning stage, while encoders in the supervised learning experiments are directly trained with labeled data. Semi-supervised learning experiments are designed to evaluate the performance of models on small-size labeled data. Several benchmark methods compared include Multi-task SSL [8], CAE [23], Masked Reconstruction (MR) [15], CPC [9], CSSHAR [10], ClusterCLHAR [12], and Dynamic Temperature [13].

We adopt the mean F1 score as the metric that is less affected negatively by the imbalanced class distribution. Specifically, the mean F1 score is given by

$$F1 = \frac{2}{|c|} \sum_{c} \frac{p_c \times r_c}{p_c + r_c}, \tag{10}$$

where $|c|$ denotes the number of classes and p_c and r_c represent the precision and recall of the class c, respectively. We present the mean F1 score in percentage format throughout the paper, i.e., in the range of $0\% - 100\%$. A higher F1 score represents better learning performance.

C. Implementation Details

We use ADAM [38] as our optimizer with the initial learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.009$, and $\epsilon = 10^{-7}$. The encoder is shared in both stages, which consists of two individual subnets and a merging subnet. Each individual subnet adopts three 1D convolution layers with the number of kernels 32, 64, and 128, respectively. The merging subnet uses three 2D convolution layers with 128, 128, and 256 kernels, respectively.

Temporal contrastive learning in pretraining. For the representation learning, we choose $M = 10$ frames as one clip, which renders the best performance, and set the batch size to 128. The training epoch is set to 6,000 and the unlabeled data for 80% of the subjects in each dataset are used in the training. The temperature τ in the InfoNCE loss is set to 1. The output layers include three fully connected layers with 256, 128, and 128 neurons, respectively.

Downstream activity recognition. Minimizing the cross entropy loss is the objective function in the downstream task. The number of epochs is set to 200. The training batch size is 256 for MobiAct, 256 for UCI-HAR, and 128 for USC-HAD, respectively. An MLP head with three fully-connected layers is added to the top of the pretrained encoder. We also apply the early stopping technique and save the model with the least validation loss. We report the mean F1 score for the downstream activity recognition. The results shown are averaged from five runs of experiments with different random seeds.

D. Results

1) Performance in Activity Recognition: Although the self-supervised learning framework always includes the pretraining stage, learning the feature representation, and fine-tuning stage adaptive to the downstream task, there is no uniform and established evaluation criterion for the representation learning. Therefore, we compare our TCLHAR framework with the aforementioned self-supervised learning methods in terms of mean F1 score in the downstream task, i.e., HAR. Specifically, we first pretrain the encoder in a contrastive learning manner on a dataset and then train a classifier on top of the frozen encoder with the labeled data from the same dataset.

Furthermore, we compare with supervised baselines, including Challa et al. [39], DeepConvLSTM [1], Supervised CSSHAR, the supervised version of the CSSHAR method, and Supervised TCLHAR using the same architecture as ours in the self-supervised learning but trained in a supervised manner. Including both supervised and self-supervised benchmarks provides a comprehensive view of the proposed method's performance, capturing insights into how it stands relative to both traditional supervised models and self-supervised strategies. Table II lists the mean F1 scores of experimental models.

TABLE II: Mean F1 score (%) comparison in activity recognition. SL and SSL denote supervised learning and self-supervised learning models, respectively.

Method	Type	MobiAct	UCI-HAR	USC-HAD
DeepConvLSTM [1]	SL	82.40	82.83	44.83
Supervised CSSHAR [10]	SL	83.92	95.26	60.56
Challa et al. [39]	SL	-	97.14	-
Supervised TCLHAR	SL	**89.35**	**97.48**	**69.61**
Multi-task SSL [8]	SSL	75.41	80.20	45.37
CAE [23]	SSL	79.58	80.26	48.82
MR [15]	SSL	76.81	81.89	49.31
CPC [9]	SSL	80.97	81.65	52.01
CSSHAR [10]	SSL	81.13	91.14	57.76
ClusterCLHAR [12]	SSL	-	92.63	58.85
Dynamic Temperature [13]	SSL	82.02	93.00	56.47
TCLHAR (ours)	SSL	**86.88**	**95.30**	**64.78**

In terms of performance in a supervised learning manner, our Supervised TCLHAR outperforms Supervised CSSHAR and DeepConvLSTM by significant margins. Ours achieves 5.43%, 2.22%, and 9.05% improvements in comparison with Supervised CSSHAR on MobiAct, UCI-HAR, and USC-HAD, respectively. In comparison with DeepConvLSTM, our framework gains 6.95%, 14.65%, and 24.78%, respectively. Even compared with the state-of-the-art Challa et al. [39], ours achieves better performance. This indicates that designing the individual convolution subnets for each sensor modality can achieve better performance in comparison with the single convolution subnet of DeepConvLSTM, especially for a cross-modality application like HAR. The architecture, specifically engineered for handling diverse sensor data, leads to a more nuanced and accurate understanding of human activities. We will conduct more analyses of the impact of individual convolution subnets in the ablation study.

For self-supervised models, our TCLHAR achieves the best mean F1 score and outperforms all self-supervised learning benchmark methods on all three datasets. Improvements from our framework are significant for MobiAct, UCI-HAR, and USC-HAD by 5.75%, 4.16%, and 7.02%, respectively. The leading performance on all three datasets indicates the effectiveness of our temporal contrastive learning in capturing the co-occurrence relationship among neighboring frames and affirms the versatility and generalizability of TCLHAR in handling various sensor configurations and environmental conditions, ensuring reliable human activity recognition across different real-world scenarios.

Comparing Supervised TCLHAR with TCLHAR and Supervised CSSHAR with CSSHAR, we can see that self-supervised models always perform worse than the supervised learning models, which can also be seen in existing studies [8], [10], [9]. It is reasonable since the trainable parameters in the self-supervised model are fewer. This limitation reflects the trade-off between reducing dependency on labeled data and potential performance reductions. It is worth noting that our TCLHAR achieves comparable or better performance with respect to the supervised models. Especially, our self-supervised model outperforms DeepConvLSTM by a great margin, which demonstrates the advantage of self-supervised learning.

2) Semi-supervised Learning: In practice, data collection from a great number of users who produce and own private data is troublesome due to privacy concerns and information safety. Furthermore, data segmentation and annotations are also tedious and expensive. Therefore, HAR systems should adapt to small-size labeled data in practice.

Experiment settings. We conduct semi-supervised learning experiments to show the effectiveness of our TCLHAR framework when very limited labeled data are accessible. Specifically, for each dataset, we use all the data without labels to pretrain the encoder in a contrastive learning manner. In the downstream task, the weights of the encoder are frozen, and a trainable classifier head is added. Randomly sampled data with a number of $k \in \{1, 2, 5, 10, 25, 50, 100, 200, 250, 300, 400\}$ per class are used for the training and $\lceil k/4 \rceil$ for the validation. For example, the amount of training data for six activities from UCI-HAR is 60 in total when $k = 10$. However, the testing split still consists of the data for 20% of the subjects from the entire dataset despite the changes in the training and validation splits, which is consistent with practical scenarios. The proportion of training data is in the range of $0.02\%-5.7\%$, $0.36\%-14.50\%$, and $0.17\%-6.7\%$ for MobiAct, UCI-HAR, and USC-HAD datasets, respectively.

We compare TCLHAR with Supervised TCLHAR and Random TCLHAR, where Random TCLHAR adopts the same architecture as TCLHAR, but its encoder is randomly initialized and frozen. We report the averaged results for each k in 10 trials. The curves of the averaged mean F1 score versus k for MobiAct, UCI-HAR, and USC-HAD are shown in Figure 2.

We can see that on all datasets, our self-supervised learning model, TCLHAR, achieves the best performance when k is extremely small. For example, our TCLHAR has about 30% and 65% F1 score improvements over Random TCLHAR and Supervised TCLHAR, respectively, when the training samples per class are fewer than 100 on UCI-HAR. When $k \leq 50$ on MobiAct, our TCLHAR achieves about 15% performance improvement in comparison with Supervised TCLHAR and Random TCLHAR. In the extreme case of $k = 1$, our TCLHAR framework can produce an F1 score larger than 55%, much better than the supervised learning model with 200 times more training data on UCI-HAR. This demonstrates the great benefits obtained from our temporal contrastive learning and the superiority of self-supervised learning in scenarios with scarce labeled data.

As the volume of training samples increases to a certain level, Supervised TCLHAR eventually outperforms TCLHAR, although the required amount of data varies with the dataset. For example, the supervised learning model outperforms TCLHAR on USC-HAD with a small margin when $k = 50$. However, when $k = 50$ on UCI-HAR, TCLHAR achieves about 70% performance improvement over Supervised TCLHAR. They achieve comparable performance until k reaches 400. It shows that our self-supervised learning model can render consistent and predictable performance on different datasets. In contrast, supervised learning models are highly volatile on different datasets when the same amount of data is

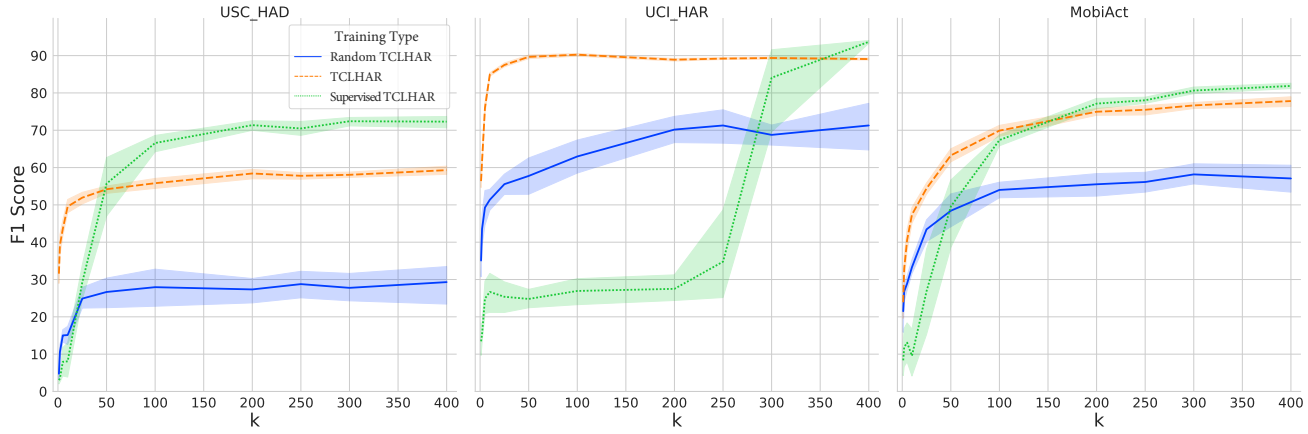


Fig. 2: Performance comparison when limited labeled data are available. Random TCLHAR and Supervised TCLHAR are trained with the labeled data in a supervised manner. TCLHAR is pretrained with all the unlabeled data and then fine-tuned with the labeled data.

available. The performance of the supervised learning models is no more than 5% over their self-supervised counterparts on both MobiAct and UCI-HAR when $k = 400$.

Unlike TCLHAR, which consistently performs better than Random TCLHAR, Supervised TCLHAR is highly volatile. For example, the F1 score of Supervised TCLHAR on UCI-HAR is stuck around 25% when the number of available samples per class $k \leq 250$, which is worse than Random TCLHAR. This is because the supervised learning model is quickly overfitted in one or two epochs, leaving it in the local minimum. In contrast, the F1 score has suddenly rocketed up to more than 90% when more training data are available. It also shows the importance of the labeled training data size to the performance of supervised learning models. In contrast, self-supervised learning models depend less on the size of training samples.

In addition, our framework consistently shows more stability in terms of performance fluctuation among different trials compared with the supervised and randomly initialized models. For example, Supervised TCLHAR and Random TCLHAR achieve about 10% and 8% less in F1 score on the UCI-HAR dataset when $k \leq 100$, while the variance of TCLHAR is less than 1%. This reduced variance indicates the superior generalization capability of the framework, reflecting its stability to different training sets and its potential for reliable deployment in various real-world scenarios.

3) Visualization of Representations: Besides evaluating the quality of representations by comparing the downstream performance, we can visualize the high-dimensional representations utilizing t-SNE [40]. Visualization techniques like t-SNE offer an intuitive way to inspect and interpret complex high-dimensional data, providing insights that numerical evaluations may overlook. t-SNE is a non-linear visualization technique that maps high-dimensional data into low-dimensional space and minimizes the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. This creates a visual map where similar data points are clustered together, making identifying patterns and relationships within the dataset easier.

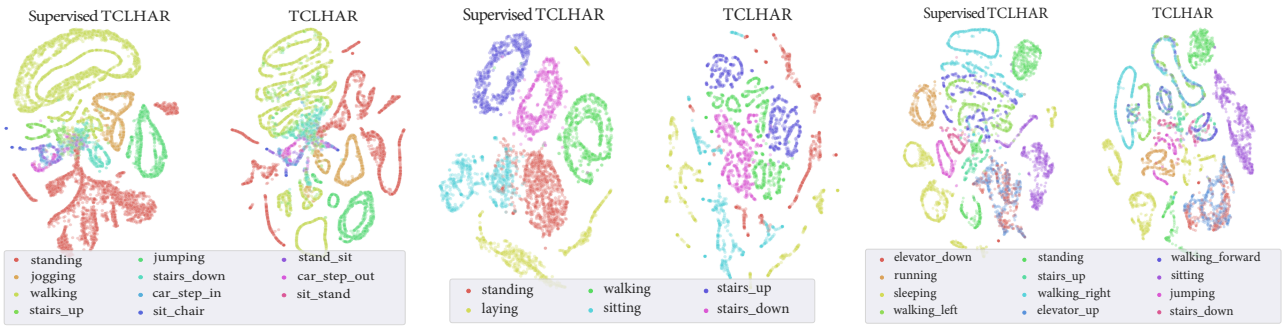
The dimension of our representation is $25,600$ in the experiments. This high dimensionality presents challenges in terms of computational efficiency and interpretation, necessitating dimensionality reduction. Therefore, we first apply principal component analysis (PCA) to reduce the dimensions to 256 and then use t-SNE to visualize them in the 2D space. Representations from TCLHAR and Supervised TCLHAR are visualized in Figure 3.

Representations with the same label are well clustered and separable from other classes for both supervised and self-supervised models. These visualizations highlight the effectiveness of our framework in capturing meaningful and distinct feature representations. The clear separation of activity classes in the t-SNE plots demonstrates the stability and generalizability of TCLHAR. The two models perform similarly to well separate among classes and to struggle. For instance, for both models, decision regions for walking and standing are clearly separated, while representations from stairs up and stairs down are greatly overlapped.

E. Ablation Study

1) Multimodal Fusion Analysis: HAR typically involves data from multiple sensor modalities, making multimodal fusion essential for effective feature extraction. Multimodal fusion combines information from diverse sensor sources, thereby enhancing the representational power of the model and leading to a more comprehensive understanding of the underlying human activities. In this ablation study, we specifically evaluate the effectiveness of the cross-modality fusion component in our TCLHAR framework.

Specifically, four settings are considered: Acc, Gyro, Acc+Gyro (Single), and Acc+Gyro (Individual). In the first two settings, data only from one sensor modality are used and the encoder consists of one individual subnet and a merging subnet. Acc+Gyro (Individual) adopts the encoder of our TCLHAR. In Acc+Gyro (Single), we replace the two individual subnets with a single convolution subnet consisting of the same number of convolution layers with twice the



(a) MobiAct (b) UCI-HAR (c) USC-HAD

Fig. 3: t-SNE visualization of representations. We extract high-dimensional representations from TCLHAR and Supervised TCLHAR and first apply PCA to reduce the number of dimensions to 256 and then utilize t-SNE to visualize them in 2D space.

number of kernels to keep the model size comparable. All encoders are followed by the same classifier head and trained on the UCI-HAR dataset in a supervised manner. Figure 4 shows the performance using different sensor modalities.

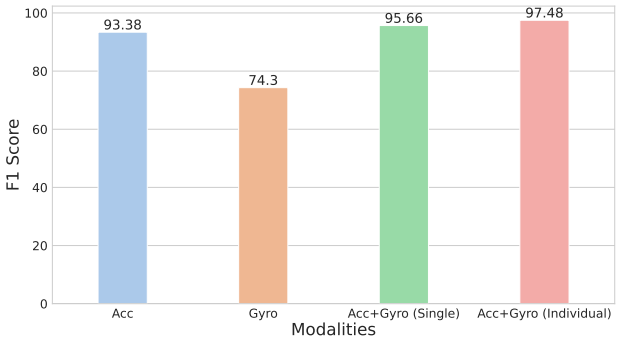


Fig. 4: Mean F1 score (%) using different sensor modalities in the supervised learning.

From the results, we can see that the performance on data of the accelerometer is greatly better than that using only the gyroscope, achieving 19.08% improvement. This significant difference underscores the accelerometer's unique capability to capture specific aspects of human movement central to the HAR, setting a foundational understanding of the relative importance of different sensors.

With more sensor modalities used, the performance benefits from the multiple sensory feature fusion, improving from 93.38% to 95.66%. This enhancement indicates the complementary nature of the information captured by different sensors, suggesting that a holistic fusion of multimodal data leads to a richer and more robust representation of human activities.

Furthermore, utilizing individual convolution subnets to each sensory modality achieves better performance over a single convolution subnet by 1.82%, demonstrating that combining inter-modality and intra-modality feature extraction is a promising way to achieve better multi-modality fusion.

2) Hyperparameter Evaluation: The hyperparameter settings of the framework significantly affect the final performance and are critical to finding the optimal model for the activity recognition task. We evaluate the impact of two important hyperparameters in the pretraining stage: the training batch size and the number of pretraining epochs. We train the encoder with a different set of hyperparameters and utilize it for downstream activity recognition.

The pretraining batch size directly influences the number of positive and negative pairs generated during the pretraining stage, which in turn impacts the representation of the temporal co-occurrence relationship among samples. As larger batch sizes produce more pairs, they increase the diversity of temporal co-occurrence patterns available for learning. Furthermore, the InfoNCE loss is also sensitive to the training batch size since the number of negative pairs, i.e., $2(N-1)$, depends on the size N. To show the effect of pretraining batch size on downstream task performance, we evaluate the pretraining batch size $N \in \{64, 128, 256, 512\}$ and results are shown in Figure 5.

From Figure 5, we can see that the best performance is obtained when the batch size is 128 for all three datasets. The downstream performance could be improved by up to 20%. The F1 scores with respect to different batch sizes form a reverse U shape. Task-relevant information is increasingly learned as the batch size increases from 64 to 128. However, noises increase to undermine the quality of representations when more negative samples are involved in contrastive learning, which can be seen from the huge performance drop when $N = 512$. Interestingly, this observation counters the empirical results in CV, where a larger batch size consistently benefits the representation learning [5], highlighting a key distinction between the learning dynamics of images and time-series data. This divergence points to underlying differences in data structure, feature space, and learning process between these two domains, emphasizing the need for domain-specific considerations in hyperparameter selection and model design.

The number of pretraining epochs also plays a critical role in capturing the temporal co-occurrence relationships within the data. More pretraining epochs allow the model to observe a greater variety of temporally adjacent positive and negative pairs, enhancing its ability to learn meaningful representations of the underlying temporal dynamics of human activities. We evaluate the number of pretraining epochs from 999 to

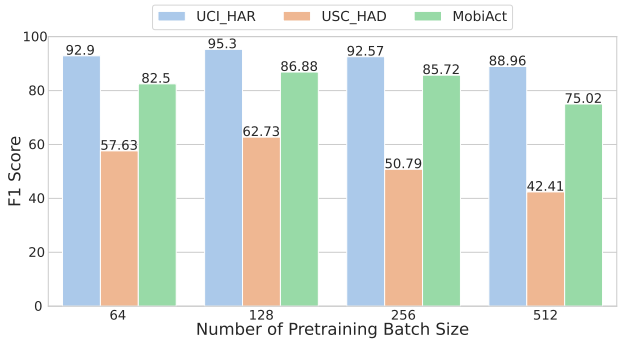


Fig. 5: Downstream task performance with encoders pretrained with batch size $N \in \{64, 128, 256, 512\}$. The optimal pretraining batch size is 128 in all three HAR datasets.

5,999, where the optimal batch size $N = 128$ is used. The performance is shown in Figure 6.

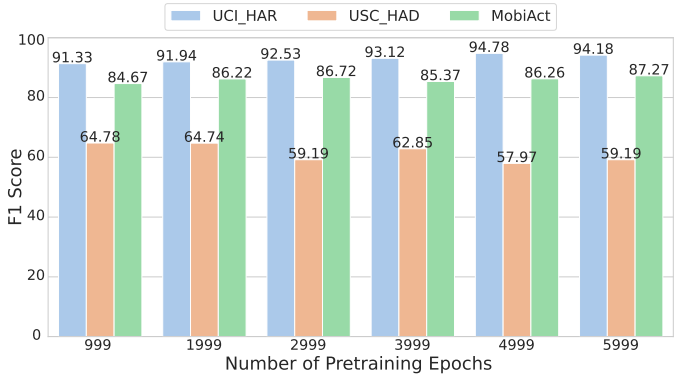


Fig. 6: Downstream task performance with encoders pretrained in different numbers of pretraining epochs.

For UCI-HAR and MobiAct datasets, increasing the number of pretraining epochs benefits the representation learning and leads to better downstream task performance. The performance gap between the highest and lowest scores is less than 3%. This relatively small variation in performance may indicate stability in the learning process, where the model's representations do not heavily depend on the exact number of pretraining epochs within this range. It could also suggest that these datasets have well-defined patterns that can be captured with moderate pretraining.

However, for the USC-HAD dataset, as the number of pretraining epochs increases, the quality of representations degenerates, and the performance decrease could be up to 6.81%. This may be due to the high similarities among some activities. For instance, the representations of elevator down and elevator up greatly overlap in the t-SNE visualization of Figure 3c. These overlapping representations signify the model's struggle to differentiate between specific activities, possibly due to the inherent similarity in their sensory patterns.

V. Conclusion and Future Work

In this paper, we present a novel self-supervised learning framework—Temporal Contrastive Learning in Human Activity Recognition (TCLHAR). This novel approach leverages temporal co-occurrence among adjacent time windows to pretrain the encoder, thereby enabling the model to grasp a more nuanced comprehension of human activities in temporal sequence. To better utilize multi-sensor measurements, we introduce a cross-modality fusion encoder that fuses features from multiple sensor modalities via distinct individual subnets and a specialized merging subnet. Extensive experiments on supervised, self-supervised, and semi-supervised settings demonstrate the effectiveness of feature representations and the superior performance of our framework to existing self-supervised models.

In the future, the impact of unlabeled data preprocessing, such as the length of time windows and the downsampling rate, on the performance can be further explored. Moreover, more metrics and direct evaluations on feature representations are needed to unearth the influence of the pretraining stage and guide the future design of self-supervised learning.

References

[1] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.

[2] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, Jun. 2017.

[3] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the International Conference on World Wide Web*, 2017, p. 351–360.

[4] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9726–9735.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning*, vol. 119, 2020, pp. 1597–1607.

[6] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6960–6970.

[7] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent a new approach to self-supervised learning," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2020, p. 21271–21284.

[8] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, Jun. 2019.

[9] H. Haresamudram, I. Essa, and T. Plötz, "Contrastive predictive coding for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, Jun. 2021.

[10] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Contrastive self-supervised learning for sensor-based human activity recognition," in *Proceedings of the IEEE International Joint Conference on Biometrics (IJCB)*, 2021, pp. 1–8.

[11] X. Ouyang, X. Shuai, J. Zhou, I. W. Shi, Z. Xie, G. Xing, and J. Huang, "Cosmo: Contrastive fusion learning with small data for multimodal human activity recognition," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, p. 324–337.

[12] J. Wang, T. Zhu, L. L. Chen, H. Ning, and Y. Wan, "Negative selection by clustering for contrastive learning in human activity recognition," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 833–10 844, 2023.

[13] B. Khaertdinov, S. Asteriadis, and E. Ghaleb, "Dynamic temperature scaling in contrastive self-supervised learning for sensor-based human activity recognition," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 4, pp. 498–507, 2022.

[14] J. Wang, T. Zhu, J. Gan, L. L. Chen, H. Ning, and Y. Wan, "Sensor data augmentation by resampling in contrastive learning for human activity recognition," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 994–23 008, 2022.

[15] H. Haresamudram, A. Beedu, V. Agrawal, P. L. Grady, I. Essa, J. Hoffman, and T. Plötz, "Masked reconstruction based self-supervision for human activity recognition," in *Proceedings of the ACM International Symposium on Wearable Computers*, 2020, p. 45–49.

[16] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," in *Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health*, 2016, pp. 100–118.

[17] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the European Symposium on Artificial Neural Networks*, 2013, pp. 24–26.

[18] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the ACM Conference on Ubiquitous Computing*, 2012, p. 1036–1043.

[19] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete consine transform and SVM," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 5041–5044.

[20] N. Kumari, A. Yadagani, B. Behera, V. B. Semwal, and S. Mohanty, "Human motion activity recognition and pattern analysis using compressed deep neural networks," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 12, no. 1, p. 2331052, 2024.

[21] P. Tokas, V. B. Semwal, and S. Jain, "Deep ensemble learning approach for lower limb movement recognition from multichannel semg signals," *Neural Computing and Applications*, vol. 36, no. 13, pp. 7373–7388, 2024.

[22] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two*, 2011, p. 1729–1734.

[23] H. Haresamudram, D. V. Anderson, and T. Plötz, "On the role of features in human activity recognition," in *Proceedings of the ACM International Symposium on Wearable Computers*, 2019, p. 78–88.

[24] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[25] 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)*, 2015, p. 1422–1430.

[26] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Conference on Neural Information Processing Systems*, 2013, pp. 3111–3119.

[28] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] H. Haresamudram, I. Essa, and T. Plötz, "Investigating enhancements to contrastive predictive coding for human activity recognition," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2023, pp. 232–241.

[31] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv e-prints*, pp. arXiv–1807, 2018.

[32] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 6827–6839, 2020.

[33] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proceedings of European Conference on Computer Vision*, 2020, p. 776–794.

[34] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv:1603.06995*, 2016.

[35] J. Gao, X. Song, Q. Wen, P. Wang, L. Sun, and H. Xu, "RobustTAD: Robust time series anomaly detection via decomposition and convolutional neural networks," *arXiv preprint arXiv:2002.09545*, 2020.

[36] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," *arXiv preprint arXiv:1702.05538*, 2017.

[37] I. Goodfellow, Y. Bengio, and C. Aaron, *Deep Learning*. MIT press, 2016.

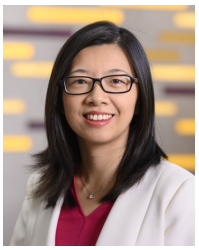
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] S. K. Challa, A. Kumar, V. B. Semwal, and N. Dua, "An optimized deep learning model for human activity recognition using inertial measurement units," *Expert Systems*, vol. 40, no. 10, p. e13457, 2023.

[40] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

Xiaobing Chen received the B.E. degree in electrical engineering and M.E. degree in control science and engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in the Division of Electrical and Computer Engineering, Louisiana State University, where he joined as a Research Assistant in 2021. His research interests include federated learning, privacy in machine learning, and optimization theory.

Xiangwei Zhou received the B.S. degree in communication engineering from Nanjing University of Science and Technology, Nanjing, China, the M.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2005, 2007, and 2011, respectively. He is currently an Associate Professor with the Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. His research interests include wireless communications and statistical signal processing, with current emphasis on wireless federated learning and Internet of Things.



Mingxuan Sun received the B.S. degree in computer science and engineering from Zhejiang University, Hangzhou, China in 2004, the M.S. degree in computer science from University of Kentucky, Lexington, KY, USA in 2006, and the Ph.D. degree in computer science from Georgia Institute of Technology, Atlanta, GA, USA in 2012. She is currently an Associate Professor with the Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA, USA. Her research interests include machine learning, information retrieval, and data mining. She is also interested in machine learning and AI applications in social informatics, security, and wireless communications. She has published research papers in leading journals and conferences including PAMI, JMLR, NeurIPS, AAAI, AISTATS, KDD, ICDM, WWW, and WSDM.

Hao Wang is an Assistant Professor in the Department Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, NJ, USA. He received both his B.E. degree in Information Security and M.E. degree in Software Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012 and 2015 respectively, and the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Toronto, Canada in 2020. His research interests include distributed ML systems, AI security and forensics, privacy-preserving data analytics, serverless computing, and high-performance computing.