Dynamic Temperature Scaling in Contrastive Self-Supervised Learning for Sensor-Based Human Activity Recognition

Bulat Khaertdinov, *Graduate Student Member, IEEE*, Stylianos Asteriadis, and Esam Ghaleb

***Abstract*—The use of deep neural networks in sensor-based Human Activity Recognition has led to considerably improved recognition rates in comparison to more traditional techniques. Nonetheless, these improvements usually rely on collecting and annotating massive amounts of sensor data, a time-consuming and expensive task. In this paper, inspired by the impressive performance of Contrastive Learning approaches in Self-Supervised Learning settings, we introduce a novel method based on the SimCLR framework and a Transformer-like model. The proposed algorithm addresses the problem of negative pairs in SimCLR by using dynamic temperature scaling within a contrastive loss function. While the original SimCLR framework scales similarities between features of the augmented views by a constant temperature parameter, our method dynamically computes temperature values for scaling. Dynamic temperature is based on instance-level similarity values extracted by an additional model pre-trained on initial instances beforehand. The proposed approach demonstrates state-of-the-art performance on three widely used datasets in sensor-based HAR, namely MobiAct, UCI-HAR and USC-HAD. Moreover, it is more robust than the identical supervised models and models trained with constant temperature in semi-supervised and transfer learning scenarios.**

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

I. Introduction

A. Human Activity Recognition

HUMAN Activity Recognition (HAR) is a crucial problem of human-computer interaction and ubiquitous computing addressed in smart manufacturing [1], health applications [2], [3] and ambient intelligence [4], [5]. HAR algorithms can be based on different types of data describing human movements. Sensor-based HAR algorithms exploit time-series signals coming from various devices, such as accelerometers and gyroscopes, using smartphones or Inertial Measurement Units (IMU) placed on different body parts. Prior to training a deep learning model on sensor signals, they should be labeled. Obviously, it is almost impossible for a person to precisely label time-series data coming from accelerometer and gyroscope sensors without using corresponding videos. Moreover, it is even more challenging, expensive and time-consuming than creating labels for videos, since there should be a specific tool where a data annotator can match time-series data and videos. In this paper, motivated by the issues of data labeling, we propose a self-supervised learning framework which uses unlabeled data to pre-train a deep feature extraction model.

Manuscript received 25 February 2022; revised 10 May 2022; accepted 19 May 2022. Date of publication 8 June 2022; date of current version 5 December 2022. This work was supported in part by the European Union's Horizon2020 Project: PeRsOnalized Integrated CARE Solution for Elderly facing several short or long term conditions and enabling a better quality of LIFE (Procare4Life) under Grant 875221. This article was recommended for publication by Associate Editor S. Yu upon evaluation of the reviewers' comments. *(Corresponding author: Bulat Khaertdinov.)*

The authors are with the Department of Data Science and Knowledge Engineering, Maastricht University, 6211 LK Maastricht, The Netherlands (e-mail: b.khaertdinov@maastrichtuniversity.nl).

Digital Object Identifier 10.1109/TBIOM.2022.3180591

B. Self-Supervised Learning

The self-supervised learning (SSL) paradigm can be used when vast amounts of labeled data could not be acquired. SSL approaches are exploited to train deep feature encoders using unannotated data samples. Normally, a model is trained in two stages within a self-supervised learning framework. During the first step, namely a pre-text task, feature encoders are pre-trained on data without annotations using a complementary task which enables a model to learn meaningful representations. Later, during the second stage, or fine-tuning, a simple model is trained to recognize class labels on top of the representations extracted by the pre-trained feature encoder using a limited amount of annotated data. Modern SSL frameworks allow models to learn robust feature representations without labeled data and show satisfactory performance even comparing to models trained in a supervised manner.

There are various families of SSL approaches which have been applied to sensor-based HAR, including reconstruction-based methods [6], [7] and transformation networks [8]. However, in recent years, contrastive learning demonstrates state-of-the-art performance in various SSL applications, including HAR [9], [10]. In contrastive learning frameworks, feature encoders are pre-trained by solving an instance classification task. Specifically, their objective is to assign higher similarity scores to inputs, or views, representing the same instance (positive samples) in a batch while giving lower scores to inputs crafted from different instances (negative samples). The positive views can be obtained by using different modalities and parts of a data instance or by applying random augmentations to it. Two inputs are considered a negative pair if they were extracted from different instances. This implies that two inputs may even belong to the same class in practice, but, since the framework is self-supervised, this information

is not available. That is one of the drawbacks when relying on negative samples in contrastive learning approaches, since negative samples might represent the same class, also known as false negative samples [11]. Moreover, this problem might become more crucial in those applications, including sensor-based HAR, where the number of classes is relatively low, increasing the chances for false negative pairs.

Some recent frameworks used in computer vision applications, such as SimSiam [12] and BYOL [13], deal with the problem of negative pairs by using positive pairs only. Other approaches [14], [15] craft different data representations and exploit them as different modalities in order to mine additional positive pairs based on similarities of representations in latent space.

C. Main Contributions

In this paper, we address the issue of false negative pairs in contrastive learning frameworks for sensor-based HAR, by introducing a dynamic temperature scaling framework shown in Figure 2. While, in SimCLR [16] (or CSSHAR for Human Activity Recognition [10]), similarity scores of augmented views, or view-level similarities, are scaled using a constant temperature value, we suggest a novel approach that uses dynamic temperature for scaling and aims to address the problem of false negative inputs. Dynamic temperature is computed based on similarities between initial instances, or instance-level similarities. These similarities are calculated for the embeddings generated by an additional SSL model, namely a convolutional autoencoder, which is pre-trained on initial instances. Subsequently, the instance-level similarities are used in training the main transformer model within the SimCLR framework. Specifically, this approach allows to dynamically scale the view-level similarities based on their instance-level similarities. In particular, we aim to decrease the impact of the most similar negative pairs on the final loss function assuming they might belong to the same activity class.

The contributions of our paper are summarized as follows:

- We address the problem of sensor-based HAR in SSL settings by pre-training the transformer encoder within the SimCLR framework using a novel negative dynamic temperature-scaled cross entropy (NDT-Xent) loss function. NDT-Xent aims to deal with the problem of false negatives in contrastive learning approaches exploiting negative samples.
- Dynamic temperature scaling addresses the issue of false negative pairs in contrastive SSL approaches. Dynamic temperature is computed using similarities of initial data instances produced by the additional autoencoder-like model, pre-trained in advance. Dynamic temperature is calculated differently for positive and negative pairs. Specifically, the negative dynamic temperature is computed for each negative pair according to the similarity scores of initial instances. Moreover, dynamic temperature for positive pairs is calculated for the whole batch and penalizes the model for generating high view-level similarities for negative samples.
- The models, pre-trained with dynamic temperature, obtain state-of-the-art results on three datasets, namely, MobiAct [17], UCI-HAR [18] and USC-HAD [19] in various scenarios comparing to other SSL methods. What is more, the proposed approach significantly outperforms CSSHAR [10] and supervised models in semi-supervised learning scenario and shows great potential for transfer learning.

II. Related Work

In this section, we list major self-supervised learning approaches with an emphasis on contrastive learning. Furthermore, we elaborate on deep learning topologies that have been used for both supervised and self-supervised sensor-based Human Activity Recognition.

A. Contrastive Self-Supervised Learning

Self-supervised learning methods aim to learn robust feature representations using unannotated data. That is done by formulating a so-called pre-text task, or pre-training, when a model attempts to learn feature representations using an auxiliary objective that enables the model to observe and understand the structure and content of unlabeled data. While earlier SSL methods rely on transformation-based [20], [21], reconstruction [22], [23] and generative [24] approaches, the most recent state-of-the-art SSL frameworks are based on the contrastive learning paradigm.

In contrastive learning, models are trained to group similar inputs, positive samples, together and push different examples, negative samples, away from each other. Whilst in supervised learning similar inputs are defined by class labels, in SSL settings, they normally represent different views of the same instance. Different views might be obtained from different modalities [25], different parts of input data [26] or using augmentations [13], [16]. While various modalities might not be available for some application areas, augmentation-based approaches can be easily adapted to different types of input data.

There is an ongoing discussion on whether negative samples should be used in contrastive learning. While the SimCLR framework [16] which uses negative pairs is quite simple and demonstrates impressive performance in different applications, it requires large batches and careful consideration of augmentations [12]. Moreover, the fact that negative pairs might be sampled from the same class during pre-training also negatively affects model performance, especially when the number of classes is low. In order to address this issue, some works suggested using positive pairs only, although they had to introduce some additional constraints to avoid model collapse [12], [13]. Another idea proposed in [11] is to correct contrastive loss by decomposing negative sample distribution based on the assumption that negative samples might not be truly negative.

B. Sensor-Based Human Activity Recognition

1) Supervised Methods: In recent works, various deep learning architectures have been applied in order to address

the sensor-based HAR problem. They include but are not limited to Convolutional Neural Networks (CNNs) [27], Recurrent Neural Networks (RNNs) [28], [29] and their combinations [30]. Furthermore, more advanced models exploiting the latest advances in deep learning were applied to the HAR problem. In [31] and [32], attention mechanisms were added on top of the LSTM networks which were trained in an end-to-end manner using classical cross-entropy loss and triplet loss, respectively. Mahmud *et al.* [33] combined Transformer-like architecture with sensor and temporal attention blocks to extract features from raw sensor data.

2) Self-Supervised Learning: Although various modern supervised models have been applied to the problem, few SSL and, especially, contrastive learning methods, have been used for sensor-based HAR, where data annotation is extremely challenging. In [6], authors compared different types of autoencoders in terms of their ability to learn feature representations. Another type of reconstruction models was later applied to HAR in [7]. These models aimed to reconstruct inputs where signals at certain timesteps had been masked out beforehand and reconstruction loss values were calculated only for these timesteps. Another type of SSL approaches previously used for the problem is transformation networks which aim to recognise a set of transformations applied to each time-window in a batch using a multi-head output layer [8].

Finally, as in other applications, contrastive SSL algorithms, namely contrastive predictive coding [9] and adapted SimCLR [10], demonstrate state-of-the-art performance for the sensor-based HAR task. However, both approaches do not consider that negative pairs can affect performance, although the number of activity labels in the datasets employed is relatively low. This work aims to address this issue by introducing a dynamic temperature that leverages knowledge about similarities between initial instances to adjust values of view-level similarities.

III. Methodology

In this section, we formally define the problem of sensor-based HAR and describe the typical pre-text task in contrastive learning frameworks, such as Contrastive Self-supervised learning approach to Sensor-based HAR (CSSHAR) [10]. Moreover, we propose a novel framework introducing a concept of dynamic temperature. Dynamic temperature leverages similarities between initial instances in order to dynamically scale the view-level similarities within a contrastive learning framework.

A. Problem Formulation

1) Sensor-Based HAR in SSL Settings: Sensor-based HAR can be defined as a multivariate time-series classification problem, i.e., each sequence of signals $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_T}] \in \mathbb{R}^{T \times S}$, where T is the length of the sequence and S is the number of input channels, is associated with label $y \in Y$, where Y is a set of activities in a dataset. A t-th input of the sequence consists of S channels and can be written as $\boldsymbol{x_t} = [x_t^1, x_t^2, \ldots, x_t^S] \in \mathbb{R}^S$.

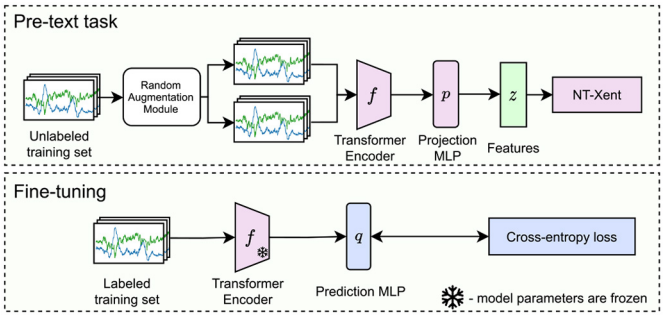


Fig. 1. The SimCLR framework applied to sensor-based HAR – CSSHAR [10]. First, a transformer encoder is pre-trained using NT-Xent loss. Then, it is frozen and generates features which are used to train output layers.

In self-supervised learning approaches, models are normally trained in two stages, namely pre-text or pre-training and fine-tuning tasks. During the pre-text task, the encoder is trained to produce feature representing signals in a latent space and can be represented as a function $f : \mathbb{R}^{T \times S} \rightarrow \mathbb{R}^D$, where D is the dimensionality of the feature space. Next, during fine-tuning, these features are passed to a model $h : \mathbb{R}^D \rightarrow \mathbb{R}^Y$ producing values corresponding to softmax probabilities for each activity in a dataset.

2) Pre-Text Task in Contrastive Learning: The purpose of a pre-text task, or pre-training, in SSL approaches is to train a deep feature encoder on unlabeled data. In the pre-text task of contrastive learning approaches, such as SimCLR, models are trained to group semantically similar inputs together in a latent space and push different examples apart from each other. Unlike in supervised settings, SSL models do not have access to annotations of instances. Thus, they are trained to match different representations of the same instance. This is done by assigning high similarity scores to different views corresponding to the same data instance, also called positive samples, and generate low scores for the views coming from different instances, i.e., negative samples.

In the SimCLR framework [16], different views of the same instance are obtained by applying two different augmentations t and t' to each data instance $\mathbf{X}_l$ in a batch. Views $t(\mathbf{X}_l)$ and $t'(\mathbf{X}_l)$ are called a positive pair. These augmented views are then passed through an encoder f and a projection head p to obtain feature representations $z_i = p(f(t(\mathbf{X}_l)))$ and $z_j = p(f(t'(\mathbf{X}_l)))$. Here, z_i and z_j is a positive pair of view-level features. Meanwhile, z_i and z_j form negative pairs with features z_k obtained from different instances in the batch. Finally, a matrix with pairwise cosine similarities of the features computed for the whole batch is used to calculate loss. In particular, normalized temperature-scaled cross entropy (NT-Xent) loss aims to contrast positive pair similarity in comparison to similarities of negative pairs present in a batch. The loss function for the representations of positive views z_i and z_j is defined as follows:

$$l(i,j) = -log \frac{exp\left(\frac{s_c(z_i,z_j)}{\tau}\right)}{exp\left(\frac{s_c(z_i,z_j)}{\tau}\right) + \sum_{k=1}^{2n} \mathbb{I}_{[k \neq i,j]} exp\left(\frac{s_c(z_i,z_k)}{\tau}\right)} \quad (1)$$

where $s_c(z_i, z_j)$ is a cosine view-level similarity between z_i and z_j, τ is a temperature parameter, and n is a batch

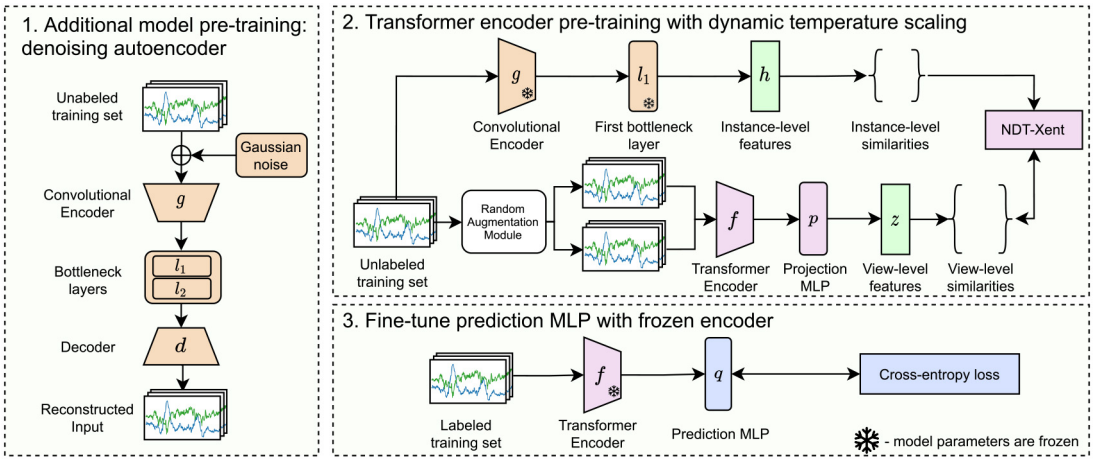


Fig. 2. The proposed framework. First step: a convolutional encoder is pre-trained within a denoising autoencoder framework. Second step: the convolutional encoder is frozen and used to produce instance-level similarities for dynamic temperature scaling, while the transformer encoder within the SimCLR-like framework computes view-level similarities. Both view and instance-level similarities are used to calculate the NDT-Xent loss. Third step: the main transformer encoder is frozen and exploited to fine-tune the prediction MLP model.

size [16]. A diagram for the SimCLR approach adapted for the sensor-based HAR problem, known as CSSHAR [10], is demonstrated in Figure 1.

Note that, since data is not annotated, the representations z_i and z_k in the denominator of equation (1) can come from instances of the same class, forming a false negative pair. Hence, in this case, the learning algorithm will push the instances corresponding to the same class away from each other in a manifold space [11]. This problem becomes even more evident in those tasks, where the number of classes is low, since it is likely that instances from the same class will be considered as negative samples in a batch. In equation (1), temperature τ is a static hyperparameter that should be selected carefully as it affects model performance significantly [16]. The selected temperature value controls the softness of similarities distribution. Specifically, high temperature values produce soft final similarity scores, whereas low values make them sharper.

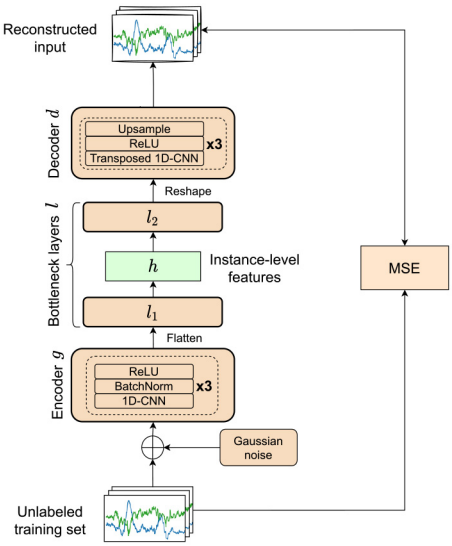


Fig. 3. Convolutional denoising autoencoder contains three blocks of convolutional layers. It passes the output of convolutional blocks to the bottleneck layers generating features in the latent space. These features are later reconstructed into the initial signal using the second bottleneck layer and the decoder using the MSE loss function.

B. Framework Overview

The proposed framework shown in Figure 2 aims to address the problem of false negative samples in the contrastive learning approaches by introducing dynamic temperature scaling and a novel negative dynamic temperature-scaled cross entropy (NDT-Xent) loss function. Dynamic temperature is used to scale view-level similarities and computed based on similarities of initial data instances, or instance-level similarities. The view level similarities, as in SimCLR, are computed using the main encoder f and the projection MLP p, while the instance-level similarities are calculated for features extracted by an additional encoder g. In this paper, we employ a transformer encoder (Figure 4) as the main encoder and denoising convolutional autoencoder (Figure 3) as the additional model.

The training routine for the suggested method consists of three stages. First, the convolutional autoencoder is pre-trained on initial data instances. Secondly, we pre-train the main transformer encoder with the proposed dynamic temperature scaling technique and NDT-Xent loss. In this stage, the convolutional encoder is not being updated and used to produce instance-level similarities for scaling. The transformer encoder is trained on augmented views of data instances and generates view-level features and similarities between them. The view-level similarities are scaled by instance-level similarities within the NDT-Xent loss function. Finally, the last stage is fine-tuning, when the prediction MLP model is trained based on features generated by the frozen transformer encoder.

A step-by-step description for the first two stages is listed in Algorithm 1. The presented algorithm is described for the case when two transformed views are generated for each input instance. In the following subsections, we provide specific

Algorithm 1 Model Pre-Training Using NDT-Xent

Data: unlabelled dataset $\{\boldsymbol{X}_l\}_{l=1}^{N}$, where N is the number of samples in it.
Input: transformer encoder f, convolutional encoder g, bottleneck layers $l = [l_1, l_2]$, decoder d, projection head p
# stage 1: autoencoder pre-training
for *each epoch* **do**
  sample a random batch $\{\boldsymbol{X}_l\}_{l=1}^{n}$ of size n;
  apply random noise to each sample in a batch to obtain $\{\boldsymbol{X'}_l\}_{l=1}^{n}$;
  pass $\{\boldsymbol{X'}_l\}_{l=1}^{n}$ through encoder g, bottleneck layers l, and decoder d to obtain reconstructed input $\{\boldsymbol{X^{rec}}_l\}_{l=1}^{n}$;
  compute MSE for $\{\boldsymbol{X}_l\}_{l=1}^{n}$ and $\{\boldsymbol{X^{rec}}_l\}_{l=1}^{n}$;
  backpropagate gradients and update learnable parameters;
end
stage 2: transformer encoder pre-training
freeze the convolutional encoder g and bottleneck l_1;
for *each epoch* **do**
  sample a random batch $\{\boldsymbol{X}_l\}_{l=1}^{n}$ of size n;
  pass $\{\boldsymbol{X}_l\}_{i=l}^{n}$ through the frozen convolutional encoder g and bottleneck l_1 and compute instance-level similarities $s_a(h_i, h_k)$ on top of the produced features;
  pass $\{\boldsymbol{X}_l\}_{l=1}^{n}$ through the random augmentation module to obtain transformed views $\{\boldsymbol{X'}_l\}_{l=1}^{2n}$;
  pass $\{\boldsymbol{X'}_l\}_{l=1}^{2n}$ through transformer encoder f and projection head p to obtain view-level similarities $s_a(z_i, z_k)$;
  use $s_a(h_i, h_k)$ and $s_a(z_i, z_k)$ to compute NDT-Xent loss according to equations 4 and 5;
  backpropagate gradients and update learnable parameters of transformer encoder f;
end

details for each stage of training and introduce the NDT-Xent loss function. The details of the topology along with the selected values of the hyperparameters are specified in Section IV.

C. Convolutional Autoencoder Pre-Training

The convolutional autoencoder shown in Figure 3 is trained on initial data instances. It consists of the three-layer convolutional encoder g, two bottleneck layers $l = [l_1, l_2]$ and reflective decoder d. To train the autoencoder, we apply Gaussian noise to each instance $\mathbf{X}_l$ in a batch, pass them through the encoding part of the network, obtain instance-level features $h_l = l_1(g(\mathbf{X}_l))$ in a latent space and then reconstruct the input using the decoding part of the autoencoder. Finally, the reconstruction loss, namely mean squared error (MSE), is computed for the initial and reconstructed inputs. The convolutional encoder and first bottleneck layer are later used with frozen parameters to generate instance-level features.

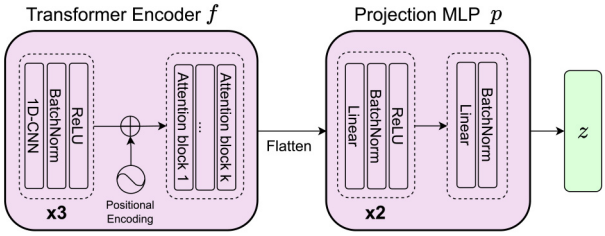


Fig. 4. Transformer encoder architecture. First, three convolutional blocks are applied to input signal. Later, positional encoding and transformer-like self-attention are used to obtain feature representations. Finally, the extracted features are passed through the projection MLP.

D. Transformer Encoder Pre-Training Using NDT-Xent

The second step is a pre-text task for the transformer encoder f. In this paper, inspired by recent success of transformer-like architectures in both supervised and SSL approaches to sensor-based HAR, the main model consists of three CNN layers followed by positional encoding and transformer self-attention layers [34], as shown in Figure 4.

The pre-text task for the transformer encoder is separated into two branches (Figure 2, step 2). First, the pre-trained convolutional autoencoder, namely the encoder and the first bottleneck layer, is used with frozen parameters in order to generate instance-level features and compute the instance-level similarity matrix (Section III-C). Another branch has a typical SimCLR-like flow, also known as Contrastive Self-supervised learning approach to Sensor-based HAR (CSSHAR) [10]. First, two random transformations t and t' are applied to each input $\mathbf{X}_l$. Then, the augmented views are passed through the transformer encoder f and projection head p extracting view-level features z_i and z_j.

Once both instance and view-level similarities are calculated for the whole batch, they are used to compute NDT-Xent loss. First, we introduce dynamic negative temperature $\tau_{i,k}^{neg}$ for a negative pair formed by views i and k as follows:

$$\tau_{i,k}^{neg} = \tau * \sqrt{s_a(h_i, h_k)}, \tag{2}$$

where $s_a(\cdot)$ - is angular similarity, τ is a default temperature hyperparameter, and $h_i = h_j = l_1(g(\mathbf{X}_l))$ is a representation of instance $\mathbf{X}_l$. Furthermore, in equation (2), h_k is a representation of an instance other than $\mathbf{X}_l$, obtained through the encoder part of the convolutional autoencoder. Note that, $\tau_{i,k}^{neg} = \tau_{j,k}^{neg}$, because both views $z_i = p(f(t(\mathbf{X}_l)))$ and $z_j = p(f(t'(\mathbf{X}_l)))$ forming positive pair are obtained from the same instance $\mathbf{X}_l$. This scaling allows to decrease the impact of the similar views, assuming that they can represent the same activity, into the negative similarity sum in the denominator of the NT-Xent loss function (Equation (5)).

All instance-level positive similarities are equal to 1, since positive pairs are sampled from the same example. That is why we propose to use positive dynamic temperature τ^{pos} which is calculated for the whole batch as demonstrated in Equation (3). In this way, positive dynamic temperature penalizes the model

for producing high negative similarities.

$$\tau^{pos} = \tau * \sqrt{\frac{1}{4n^2 - 2n} \sum_{i=1}^{2n} \sum_{k=1}^{2n} \mathbb{I}_{[e_i \neq e_k]} s_a(z_i, z_k)}. \quad (3)$$

In both Equations (2) and (3), the default temperature parameter is multiplied with the square root of terms containing instance-level similarities. We propose applying square root to narrow the range of temperature values used for scaling.

Finally, we plug the proposed negative and positive temperature scaling into equation (1), forming the negative dynamic temperature-scaled cross entropy (NDT-Xent) loss as follows:

$$l_{dt}(i,j) = -log \frac{exp\left(\frac{s_a(z_i,z_j)}{\tau^{pos}}\right)}{exp\left(\frac{s_a(z_i,z_j)}{\tau^{pos}}\right) + \sum_{k=1}^{2n} \mathbb{I}_{[k \neq i,j]} exp\left(\frac{s_a(z_i,z_k)}{\tau_{i,k}^{neg}}\right)} \quad (4)$$

Note that negative view-level similarities $s_a(z_i, z_k)$ are scaled by instance-level similarities in $\tau_{i,k}^{neg}$, while positive view-level similarities are scaled by τ^{pos} computed for the whole batch. Thus, the error for the whole batch is:

$$L_{dt} = \frac{1}{2n} \sum_{k=1}^{2n} (l_{dt}(2k-1, 2k) + l_{dt}(2k, 2k-1)), \quad (5)$$

where $2k-1$ and $2k$ are indexes of positive pairs.

It is important to mention that the original NT-Xent loss function exploits cosine similarity to compute similarity scores between features. However, the cosine similarity ranges between -1 and 1, and in order to adapt it to the proposed approach we represent similarity scores between 0 and 1 by using angular similarity. More formally, given two feature vectors z_i and z_j, the angular similarity is defined as follows:

$$s_a(z_i, z_j) = 1 - \frac{arccos(s_c(z_i, z_j))}{\pi}. \quad (6)$$

E. Fine-Tuning

Finally, the fine-tuning stage remains the same as for most sensor-based HAR approaches. It is illustrated in the third part of Figure 2. Specifically, for this stage, the main encoder trained within the pre-training, or pre-text, stage (Section III-D) remains frozen. Those input data instances that have labels are passed through the frozen transformer encoder to generate features. Subsequently, prediction MLP q is trained on top of these features to recognize activities using cross-entropy loss.

IV. Experimental Setup

This section introduces datasets used to evaluate the proposed models and implementation details for the experiments.

A. Datasets

In this paper, three datasets for sensor-based Human Activity Recognition are used to evaluate the proposed models, namely MobiAct [17], UCI-HAR [18] and USC-HAD [19]. Before feeding multivariate time-series data into models, it should be pre-processed. The pre-processing routine applied to all three datasets is adapted from previous studies on SSL for sensor-based HAR [8], [9], [10]. First, three channels of gyroscope and accelerometer data (6 channels overall) are extracted for each dataset. Then, all signals are downsampled to 30Hz, segmented into sequences of 1 second with 50% overlapping and normalized to zero mean and unit variance channel-wise. The specific details about each dataset are described in the following paragraphs.

MobiAct: This dataset has been collected using a smartphone (Samsung Galaxy S3) equipped with a gyroscope and accelerometer placed in pockets of 61 human users. As in previous works, we used the second version of the dataset covering 11 unique activities of daily living. As in previous studies, 20% of the users were randomly held out for the test set, 20% of the remaining subjects – for the validation set and the remaining ones were used for training. The following activities have been collected: standing, walking, jogging, jumping, going upstairs, going downstairs, stand-to-sit, siting, sit-to-stand, entering a car, exiting a car.

UCI-HAR: This dataset, also known as the UCI-smartphones dataset, has been collected using a Samsung Galaxy S2 mobile device attached to subjects' waists. The smartphones captured 3-channel accelerometer and gyroscope data from 30 subjects aged 19-48 years. For the dataset, we ignored transition activities and used 6 main activities of daily living. As for the MobiAct dataset, 20% of subjects have been used for tests, and 20% of the remaining subjects were used for validation. The activities presented in the dataset are: sitting, laying, standing, walking, going downstairs, and going upstairs.

USC-HAD: The USC-HAD dataset has been recorded using a MotionNode IMU device placed on one hip of each user. Out of 14 participants, subjects 11 and 12 were used for validation, while data from subjects 13 and 14 were used as the test set. The participants have performed 12 activities, including walking forward, left, right, upstairs, downstairs, running forward, jumping, sitting, standing, sleeping, going up and down in an elevator.

B. Implementations Details

1) Convolutional Autoencoder: The convolutional encoder was trained within the autoencoder framework shown in Figure 3 for 50 epochs using the MSE loss function and Adam optimizer with a learning rate equal to 0.001. The input data is noised with the Gaussian noise ($\sigma = 0.5$). The kernel sizes of all convolutions are 3, 3 and 5 for the UCI-HAR, MobiAct and USC-HAD datasets, respectively. The number of feature maps was set to [32, 64, 128] for UCI-HAR and USC-HAD, and [64, 128, 256] for MobiAct. The sizes of feature vectors were set to 256, 512, and 1024 for UCI-HAR, USC-HAD and MobiAct datasets, respectively.

2) Random Augmentation Module: The augmentation module used in this work is adapted from the CSSHAR approach [10]. Specifically, four simple random time-series

augmentations, namely jittering, scaling, rotation and permutation, were used in composition to transform the initial instances. Each augmentation, apart from the jittering augmentation used for all instances, is applied with a pre-defined probability $p = 0.5$. As suggested in the CSSHAR paper, we used the following sets of augmentations for each dataset: MobiAct – {jittering, scaling, rotation}, UCI-HAR – {jittering, scaling, permutation}, USC-HAD – {jittering, scaling}.

3) Transformer Encoder: The model was trained for 200 epochs and optimized using LARS (on top of ADAM) with an initial learning rate 0.0001. The proposed transformer encoder (demonstrated in Figure 4) consists of 3 convolutional blocks with the same hyperparameters as in the convolutional autoencoder. Each convolutional block contains a one-dimensional CNN layer, batch normalization and the ReLU activation function. Besides, 6, 8 and 10 self-attention blocks with 8 heads were added on top of the convolutional blocks for MobiAct, UCI-HAR and USC-HAD datasets, respectively.

4) Fine-Tuning Details: The prediction MLP was trained on top of the features extracted by the pre-trained transformer encoder for 50 epochs using cross-entropy loss. The model parameters were optimized using Adam with initial learning rate of 0.0001 which is decreased twice after 10 epochs of training with no improvement. The prediction MLP consists of 2 hidden layers (256 and 128 neurons) with ReLU activation and dropout ($p = 0.2$).

V. Evaluations

In this section, we describe evaluation scenarios employed to assess the performance of the proposed models and compare them to identical supervised model and other SSL approaches. All models were implemented using PyTorch framework and experiments were conducted on Nvidia Titan V GPU card. One epoch of pre-training using the CSSHAR model with the set of hyperparameters defined in Section IV takes approximately 2 minutes on the largest MobiAct dataset. In the case of the proposed method, the autoencoder pre-training takes about 30 seconds per epoch, while the main pre-training exploiting dynamic temperature – 3.5 minutes per epoch. Nevertheless, the fine-tuning time remains the same for both approaches, namely about 14 seconds per epoch, in the specified environment.

A. Learning Representations for Activity Recognition

The first evaluation scenario, namely representation learning, is designed to evaluate the quality of features extracted by the encoder pre-trained in a self-supervised manner by comparing it to other supervised and SSL topologies exploited for sensor-based HAR.

For this scenario, first, the encoder is pre-trained on an unannotated dataset. Then, the pre-trained encoder with frozen parameters is used to fine-tune a shallow prediction model on the full annotated dataset. Finally, the test set is passed through the encoder and fine-tuned prediction model. In this work, as in the major recent studies on sensor-based HAR, the macro F1-score is used as a performance metric.

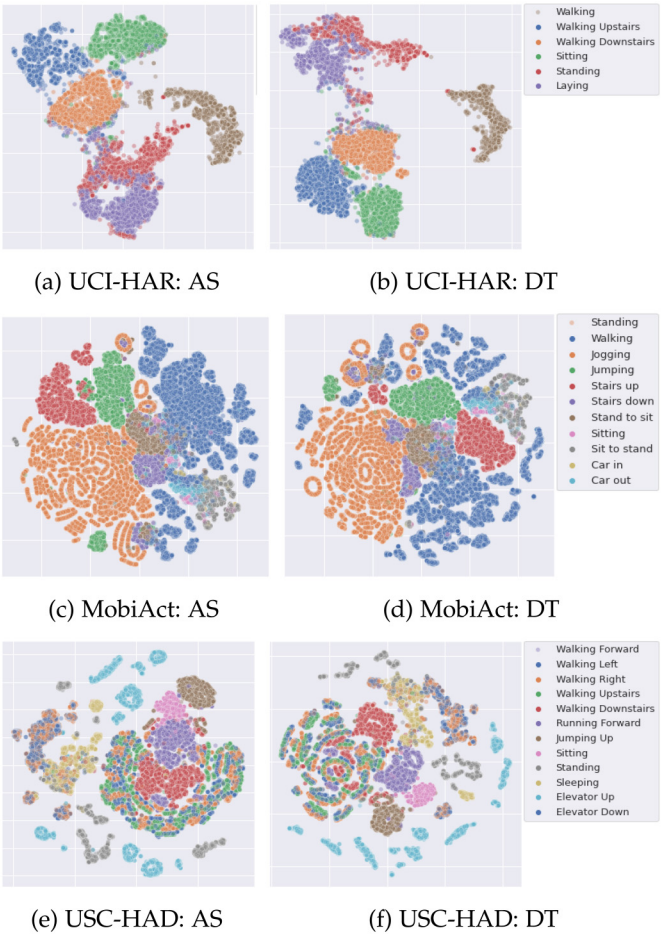
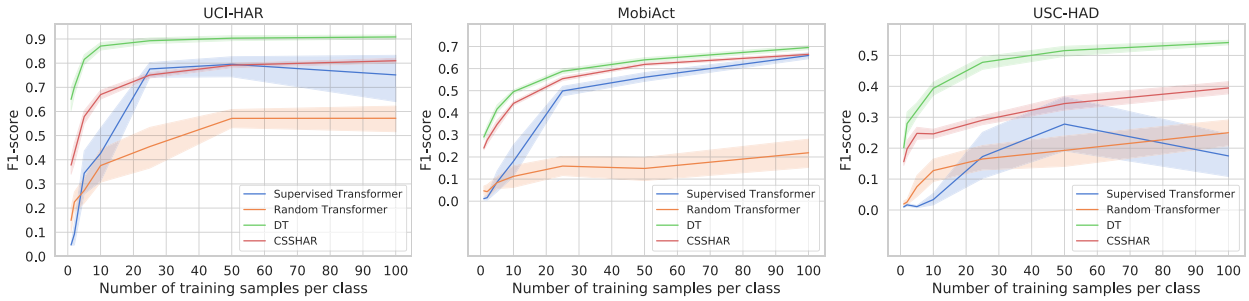


Fig. 5. Learned feature representations visualized using the t-SNE algorithm. The visualized features are produced by the transformer encoder pre-trained in a contrastive learning manner using angular similarity (a, c, e) and dynamic temperature (b, d, f).

1) Dynamic Temperature and Angular Similarity: We compare the quality of features learned with the dynamic temperature algorithm and without it. Specifically, for this experiment, the pre-text task was done for the model following the original SimCLR-like framework using angular similarity (AS), Equation (6), as a similarity function between representations. Furthermore, we have also pre-trained the proposed dynamic temperature (DT) approach described in Section III. Both models were trained and tested using the same set of hyperparameters and protocols specified in Section IV.

First, we visualize representations generated by the implemented models pre-trained using angular similarity and dynamic temperature in Figure 5. Specifically, a t-SNE algorithm [35] is used to project the high-dimensional representations into a two dimensional space. The high dimensional representations are acquired by flattening the outputs of the transformer encoder.

As for the quantitative assessment of the pre-text procedures, Table I demonstrates the macro F1-score values obtained on the test sets of the used datasets. As can be seen from the table, models showing the best performance on all three datasets were pre-trained using the proposed



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

Fig. 6. Average F1-scores with 95% confidence intervals for the semi-supervised learning scenario.

TABLE I
F1-SCORES FOR THE ABLATION STUDY WITHIN THE BASELINE ACTIVITY RECOGNITION SCENARIO. AS AND DT REFER TO ANGULAR SIMILARITY AND DYNAMIC TEMPERATURE

	MobiAct	UCI-HAR	USC-HAD
AS	81.19	92.4	50.12
DT	**82.02**	**93.00**	**55.78**

TABLE II
F1-SCORES FOR THE REPRESENTATION LEARNING SCENARIO ON PRE-DEFINED SUBJECT SPLITS. ST REFERS TO THE SUPERVISED TRANSFORMER

Method (Study)	MobiAct	UCI-HAR	USC-HAD
ST (**ours**)	85.18	95.26	60.56
Multi-task SSL [8]	75.41	80.20	45.37
CAE [6]	79.58	80.26	48.82
Masked Reconstruction [7]	76.81	81.89	49.31
CSSHAR [10]	81.13	91.14	**57.76**
DT (**ours**)	**82.02**	**93.00**	56.47

dynamic temperature model. The most significant improvement of more than 5% F1-score has been observed for the USC-HAD dataset.

2) Comparisons With Related Works: Table II compares the performance of the suggested approach with the recent SSL benchmarks as well as the supervised transformer (ST) model in the representation learning scenario. The supervised transformer was trained with exactly the same hyperparameters as the encoder in the dynamic temperature (DT) approach.

The proposed method outperforms all the previous SSL models on the MobiAct and UCI-HAR datasets by at least about 1% and 2%, respectively. However, for the USC-HAD dataset, CSSHAR [10] has a higher score than the best model with dynamic temperature. This might be related to the choice of hyperparameters and the use of angular similarity metric on the selected data splits, since the model with static temperature and angular similarity (presented in Table I) performed significantly worse than the model with cosine similarity (CSSHAR) with the same hyperparameters.

Another reason that might have played a significant role is a protocol with pre-defined data splits based on subjects. Thus, we have also compared the model trained with dynamic temperature (DT) from Table I with the static temperature model (CSSHAR) and the CPC model using subject-wise cross-validation. Specifically, each dataset was randomly separated into 5 folds based on subjects. Each fold was used as a test set for one of the experiments, while the remaining folds were used for training (80%) and validation (20%). For the USC-HAD dataset, as proposed in [9],instead of the fifth fold, we use the folds pre-defined in Section IV-A (subject 13 and 14 for test). The obtained average F1-scores for this experiment are presented in Table III.

TABLE III
MEAN F1-SCORES FOR THE 5-FOLD SUBJECT-WISE CROSS-VALIDATION EXPERIMENT

Method	MobiAct	UCI-HAR	USC-HAD
CPC [9]	76.24	88.22	65.15
CSSHAR [10]	81.27	89.67	59.43
DT (ours)	**81.80**	**90.21**	**67.74**

According to the table, the proposed dynamic temperature model outperforms both CSSHAR and CPC models. It is also worth mentioning that the CSSHAR model shows significantly lower performance than the dynamic temperature model when using subject-wise cross-validation, which is especially notable for the USC-HAD dataset (about 8%).

B. Semi-Supervised Learning Scenario

Semi-supervised learning is a more practical and realistic scenario. Namely, it is designed to assess models when small amounts of annotations are available. In this case, the pre-training stage remains the same as for the representation learning task. However, at the fine-tuning stage, only a limited number k of training examples per class are available for training the prediction model on top of the encoder. Finally, the model is evaluated on the whole test set, i.e., each example from the test set is passed through the encoder and the prediction model. As in previous studies [9], [10], we evaluate models using $k \in 1, 2, 5, 10, 25, 50, 100$. Moreover, the experiment is repeated 10 times for each value of k. Finally, we compute mean F1-scores as well as 95% confidence intervals for each k.

Besides evaluating the dynamic temperature approach, we also assess the performance of CSSHAR [10], identical supervised and random transformer models. The identical supervised model has the same architecture as the SSL model but it is trained in the end-to-end manner, i.e., encoder is not frozen

TABLE IV
F1-SCORES FOR THE TRANSFER LEARNING SCENARIO

	UCI-HAR	USC-HAD
Supervised Transformer	86.62	39.80
Multi-task SSL [8]	73.89	31.35
CAE [6]	84.15	**51.66**
Masked Reconstruction [7]	81.37	46.19
CSSHAR [10]	88.26	48.73
DT (ours)	**90.35**	49.90

and the whole model is fine-tuned with the annotated training set limited for the experiment. In contrast, the random model consists of a randomly initialized transformer encoder which is frozen. In other words, it can be interpreted as the CSSHAR approach with zero pre-training (pre-text) epochs.

In this experiment, we used pre-defined training, validation and test splits described in Section IV-A. The results of the semi-supervised learning experiment are summarized in Figure 6. As can be seen from the figure, the proposed model with dynamic temperature scaling demonstrates more robust performance than the other models, especially, when very limited annotated data is available. There is also a clear difference between the performance when k is set to maximum, i.e., 100. Namely, the model with dynamic temperature clearly outperforms CSSHAR and Supervised models by at least 10, 5 and 13% on UCI-HAR, MobiAct and USC-HAD, respectively.

C. Transfer Learning Scenario

The final evaluation scenario is transfer learning. This protocol is designed to assess the ability of the proposed model to transfer knowledge from one dataset to another. Specifically, we simulate a scenario when two types of data are available for training. First, a large collection of data, MobiAct in our case, is used for pre-training an encoder in a SSL manner. Later the pre-trained encoder with the frozen weights is used to fine-tune the output layers on smaller datasets, namely UCI-HAR and USC-HAD. We also compare the performance of the SSL models to the models pre-trained using annotated data. In this case, we train the identical encoder in a supervised manner on the MobiAct dataset. Later, as for the SSL approaches, the encoder weights are frozen and only the output MLP model is fine-tuned on top of the feature produced by the encoder. The results for this experiment are summarized in Table IV.

According to the table, the proposed model shows the solid performance when pre-trained on a large unlabeled dataset and fine-tuned with a labeled dataset unseen during pre-training. Specifically, it has outperformed all the previous models on the UCI-HAR dataset and performed worse only than the CAE model on USC-HAD. It is also important to mention that the encoder pre-trained with dynamic temperature has significantly outperformed the supervised version of the same model. This is a clear indicator that SSL methods, and specifically the dynamic temperature algorithm, have the potential to learn robust and dataset agnostic feature representations. This is especially important for real-world applications, since it is possible to achieve satisfactory performance without collecting large datasets and by using previously collected open-source datasets to pre-train deep feature encoders.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduce a novel approach based on the contrastive self-supervised learning frameworks for sensor-based Human Activity Recognition. The suggested method addresses the problem of false negative samples in contrastive learning. The proposed framework exploits a novel loss function, namely NDT-Xent, that computes dynamic temperatures for scaling based on representations of initial instances acquired using another SSL model. The extensive evaluations held on three widely used open-source datasets have shown that the proposed method achieves state-of-the-art results in the SSL activity recognition task. Furthermore, it has demonstrated strong potential in semi-supervised and transfer learning by outperforming the contrastive learning approach with static temperature, namely CSSHAR, and the identical feature encoder trained in a supervised learning manner.

Further research can extend this study in the following directions. First, a natural progression of this work is to adapt this idea to the problem of multimodal Human Activity Recognition. In this case, the dynamic temperature needs to be computed using feature representations obtained for another modality, hence, the additional autoencoder could be discarded. Moreover, IMU data coming from different devices (e.g., accelerometer and gyroscope) can be treated as different modalities as it was previously proposed in [36], [37]. Besides this, dynamic temperature shares common characteristics with knowledge distillation methods as it passes unseen knowledge from one model to another. For future work, it might be interesting to use the proposed algorithm along with other knowledge distillation methods in a practical scenario when there are constraints on complexity of a model used in real-time applications.

REFERENCES

[1] R. Grzeszick, J. M. Lenk, F. M. Rueda, G. A. Fink, S. Feldhorst, and M. ten Hompel, "Deep neural network based human activity recognition for the order picking process," in *Proc. 4th Int. Workshop Sens. Based Activity Recognit. Interact.*, New York, NY, USA, 2017, pp. 1–6.
[2] M. Panwar *et al.*, "Rehab-net: Deep learning framework for arm movement classification using wearable sensors for stroke rehabilitation," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 11, pp. 3026–3037, Nov. 2019.
[3] G. Bhat, Y. Tuncel, S. An, H. G. Lee, and U. Y. Ogras, "An ultra-low energy human activity recognition accelerator for wearable health applications," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, 2019.
[4] P. Skocir, P. Krivic, M. Tomeljak, M. Kusek, and G. Jezic, "Activity detection in smart home environment," *Procedia Comput. Sci.*, vol. 96, pp. 672–681, Dec. 2016.
[5] H. D. Mehr and H. Polat, "Human activity recognition in smart home with deep learning approach," in *Proc. 7th Int. Istanbul Smart Grids Cities Congr. Fair (ICSG)*, 2019, pp. 149–153.
[6] H. Haresamudram, D. V. Anderson, and T. Plötz, "On the role of features in human activity recognition," in *Proc. 23rd Int. Symp. Wearable Comput.*, 2019, pp. 78–88.
[7] H. Haresamudram *et al.*, "Masked reconstruction based self-supervision for human activity recognition," in *Proc. Int. Symp. Wearable Comput.*, New York, NY, USA, 2020, pp. 45–49.
[8] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 2, pp. 1–30, 2019.
[9] H. Haresamudram, I. Essa, and T. Plötz, "Contrastive predictive coding for human activity recognition," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–26, Jun. 2021. [Online]. Available: https://doi.org/10.1145/3463506

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

[11] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2020, pp. 1–11.

[12] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15750–15758.

[13] J.-B. Grill *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[14] T. Han, W. Xie, and A. Zisserman, "Self-supervised co-training for video representation learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2020.

[15] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang, "3D human action representation learning via cross-view consistency pursuit," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 4739–4748.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.* vol. 119, 2020, pp. 1597–1607. [Online]. Available: http://proceedings.mlr.press/v119/chen20j.html

[17] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The MobiAct dataset: Recognition of activities of daily living using smartphones," in *Proc. Int. Conf. Inf. Commun. Technol. Ageing Well e-Health Vol. 1 ICT4AWE (ICT4AGEINGWELL)*, 2016, pp. 143–151.

[18] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Int. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2013, pp. 437–442. [Online]. Available: http://hdl.handle.net/2117/20897

[19] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proc. ACM Conf. Ubiquitous Comput.*, New York, NY, USA, 2012, pp. 1036–1043.

[20] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1422–1430.

[21] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=S1v4N2l0-

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, New York, NY, USA, 2008, pp. 1096–1103.

[23] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 645–654.

[24] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–18. [Online]. Available: https://openreview.net/forum?id=BJtNZAFgg

[25] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer Int., 2020, pp. 776–794. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58621-8_45#citeas

[26] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[27] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 3995–4001.

[28] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, New York, USA, 2016, pp. 1533–1540.

[29] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual Bidir-LSTM for human activity recognition using wearable sensors," *Math. Problems Eng.*, vol. 2018, Dec. 2018, Art. no. 7316954. [Online]. Available: https://www.hindawi.com/journals/mpe/2018/7316954/

[30] F. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016. [Online]. Available: http://dx.doi.org/10.3390/s16010115

[31] M. Zeng *et al.*, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proc. ACM Int. Symp. Wearable Comput.*, New York, NY, USA, 2018, pp. 56–63.

[32] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Deep triplet networks with attention for sensor-based human activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Kassel, Germany, Mar. 2021, pp. 1–10.

[33] S. Mahmud *et al.*, "Human activity recognition from wearable sensor data using self-attention," in *Proc. 24th Eur. Conf. Artif. Intell. (ECAI)*, Santiago de Compostela, Spain, 2020, pp. 1332–1339.

[34] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran, 2017.

[35] L. van der Maaten and G. Hinton, "Viualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[36] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1747–1756, May 2020.

[37] Y. Jain, C. I. Tang, C. Min, F. Kawsar, and A. Mathur, "ColloSSL: Collaborative self-supervised learning for human activity recognition," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–28, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3517246

Bulat Khaertdinov (Graduate Student Member, IEEE) received the bachelor's degree from Kazan Federal University, Russia, and the M.Sc. degree from The University of Manchester, U.K. He is pursuing the Ph.D. degree with the Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands. His research is mainly focused on applications of deep metric learning and self-supervised learning to the problem of human activity recognition using different data modalities.

Stylianos (Stelios) Asteriadis is an Associate Professor with the Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands, where he is currently the Coordinator of the Cognitive Systems Research Group, and teaches the courses of human–computer interaction and affective computing, artificial intelligence, and computer vision. He has published over 80 peer reviewed papers in the fields of computer vision, automated human behavior recognition, and artificial intelligence. He is conducting research in ambient assisted living and automated human emotion recognition, with particular focus on in-the-wild applications, making use of various sensorial cues and deep learning. Large focus of his work is placed on collaborations with end-users in real-operating environments, through international, and EU funded large scale projects.

Esam Ghaleb received the Ph.D. degree in bimodal audio-video emotion recognition from Maastricht University, The Netherlands, where he is a Postdoctoral Researcher with the Department of Data Science and Knowledge Engineering. He is working in the Horizon 2020 Project, ProCare4Life, where his research focuses on personalized and dynamic multimodal human behavioral analysis for people with neurodegenerative diseases. His research has been published in various peer-reviewed conference proceedings and high-impact journals. His research interests revolve around human-centered AI, in the fields of computer vision, machine learning, affective computing, explainable AI (XAI), and human behavioral analysis, aiming at advancing having AI solutions to have a societal impact, considering scientific insights into humans' behaviors and knowledge. He co-organized the first AI and Sensor-Supported Integrated care Solutions (ASSIST) workshop and served as a reviewer for many international conferences and high-impact journals.