Contrastive Predictive Coding for Human Activity Recognition

HARISH HARESAMUDRAM, School of Electrical and Computer Engineering, Georgia Institute of Technology, USA
IRFAN ESSA, School of Interactive Computing, Georgia Institute of Technology, USA
THOMAS PLÖTZ, School of Interactive Computing, Georgia Institute of Technology, USA

Feature extraction is crucial for human activity recognition (HAR) using body-worn movement sensors. Recently, learned representations have been used successfully, offering promising alternatives to manually engineered features. Our work focuses on effective use of small amounts of labeled data and the opportunistic exploitation of unlabeled data that are straightforward to collect in mobile and ubiquitous computing scenarios. We hypothesize and demonstrate that explicitly considering the temporality of sensor data at representation level plays an important role for effective HAR in challenging scenarios. We introduce the Contrastive Predictive Coding (CPC) framework to human activity recognition, which captures the long-term temporal structure of sensor data streams. Through a range of experimental evaluations on real-life recognition tasks, we demonstrate its effectiveness for improved HAR. CPC-based pre-training is self-supervised, and the resulting learned representations can be integrated into standard activity chains. It leads to significantly improved recognition performance when only small amounts of labeled training data are available, thereby demonstrating the practical value of our approach.

Additional Key Words and Phrases: human activity recognition, representation learning, contrastive predictive coding

1 INTRODUCTION

Body-worn movement sensors, such as accelerometers or full-fledged inertial measurement units (IMU), have been extensively utilized for a wide range of applications in mobile and ubiquitous computing, including but not limited to novel interaction paradigms [67, 82, 84], gesture recognition [83], eating detection [2, 7, 73, 87], and health and well-being assessments in general [24, 54, 76]. They are widely utilized on commodity smartphones, and smartwatches such as Fitbit and the Apple Watch. The ubiquitous nature of these devices makes them highly suitable for real-time capturing and analysis of activities as they are being performed.

The workflow for human activity recognition (HAR), i.e., the all encompassing paradigm for aforementioned applications, essentially involves the recording of movement data after which signal processing and machine learning techniques are applied to automatically recognize the activities. This type of workflow is typically supervised in nature, i.e., it requires the labeling of what activities have been performed and when after the data collection is complete [8]. Streams of sensor data are segmented into individual analysis frames using a sliding window approach, and forwarded as input into feature extractors. The resulting representations are then categorized by a machine learning based classification backend into the activities under study (or the NULL class).

The availability of large-scale annotated datasets has resulted in astonishing improvements in performance due to the application of deep learning to computer vision [34, 42], speech recognition [3, 26] and natural language tasks [18, 52]. While end-to-end training has also been applied to activity recognition from wearable sensors [27, 29, 59], the depth and complexity is limited by a lack of such large-scale, diverse labeled data. However, due to the ubiquity of sensors (e.g., in phones and commercially available wearables such as watches etc.) the data recording itself is typically straightforward, which is in contrast to obtaining their annotations, thereby resulting in potentially large quantities of unlabeled data. Thus, in our work we look for approaches that can make economic use of the limited labeled data and exploit unlabeled data as effectively as possible.

Authors' addresses: Harish Haresamudram, hharesamudram3@gatech.edu, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA; Irfan Essa, irfan@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA; Thomas Plötz, thomas.ploetz@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.

Previous works such as [31, 63, 69] have demonstrated how unlabeled data can be utilized to learn useful representations for wide ranging tasks, including identifying kitchen activities [11], activity tracking in car manufacturing [71], classifying every day activities such as walking or running [4, 10, 51, 66], and medical scenarios involving identifying freeze of gait in patients suffering from Parkinson's disease [57]. In many such applications, the presence of complex and often sparsely occurring movement patterns coupled with limited annotation makes it especially hard for deriving effective recognition systems. The promising results delivered in these works without the use of labels have resulted in a general direction of integrating unsupervised learning as-is into conventional activity recognition chains (ARC) [8] in the feature extraction step. In this work, we follow this general direction of utilizing (potentially large amounts of) unlabeled data for effective representation learning and subsequently construct activity recognizers from the representations learned.

Recent work towards such unsupervised pre-training has gone beyond the early introduction using Restricted Boltzmann Machines (RBMs) [63], involving (variants of) autoencoders [31, 74], and self-supervision [32, 69]. While they result in effective representations, most of these approaches do not specifically target a characteristic inherent to body-worn sensor data – temporality. Wearable sensor data resemble time-series and we hypothesize that incorporating temporal characteristics directly at the representation learning level results in more discriminative features and more effective modeling, thereby leading to better recognition accuracy for HAR scenarios with limited availability of labeled training data – as they are typical for mobile and ubiquitous computing scenarios.

Previous work on masked reconstruction [32] has attempted to address temporality at feature level in a self-supervised learning scenario by regressing to the zeroed sensor data at randomly chosen timesteps. This incorporates local temporal characteristics into a pretext task that forces the recognition network to predict missing values based on immediate past and future data. It was shown that the resulting sensor data representations are beneficial for modeling activities, which provides evidence for our aforementioned hypothesis of temporality at feature level playing a key role for effective modeling in HAR under challenging constraints.

In this paper we present a framework that rigorously follows the paradigm of modeling temporality at representation level. We propose to utilize Contrastive Predictive Coding (CPC) [58] for unsupervised representation learning of windowed body-worn sensor data. The key insight behind this approach is that predicting just the next future timestep involves exploiting the local smoothness of the signal, whereas predicting multiple subsequent timesteps requires inferring the global structure of time-series data. This results in representations that encode the high-level information between temporally separated parts of the time-series signal [58], thereby leading to improved downstream recognition performance. Sliding windows of movement data are passed through a non-linear encoder to map the data into a latent space, where an autoregressive network such as gated recurrent units (GRUs) is subsequently utilized to provide the context representation. Using this context, multiple subsequent future timesteps from the encoder are predicted and noise contrastive estimation (NCE) [28] is utilized to train the model. Pre-trained weights are subsequently used in the Activity Recognition Chain at the feature extraction step. We apply our framework for learned representations of activities on four benchmark datasets and demonstrate improved recognition performance. In summary, our contributions are as follows:

- We introduce Contrastive Predictive Coding as an effective self-supervised pre-training scheme for HAR.
- Through a series of experiments, we analyze the network architecture choices and optimal training settings.
- We demonstrate economic use of available annotations by fine-tuning pre-trained models with limited labeled data, resulting in significant improvements over end-to-end training on the available annotations.

2 RELATED WORK ON REPRESENTATIONS OF SENSOR DATA IN HUMAN ACTIVITY RECOGNITION

In our work, we focus on learning effective representations for movement data recorded from body-worn sensing platforms through unsupervised pre-training. This follows the general direction of reducing reliance on annotated

data, by utilizing approaches that perform learning on unlabeled data. As such, relevant prior works involve the following: *(i)* feature learning for activity recognition in general; *(ii)* self-supervised learning in general, i.e., not limited to the mobile and ubiquitous computing domain; and *(iii)* self-supervised learning for HAR.

The activity recognition chain (ARC) [8] details a five step pipeline for human activity recognition. We are interested in the fourth step in the process – feature extraction – which focuses on the computation of representative features. Earlier works utilized hand crafted features for activity recognition [23]. Recently, however, end-to-end training has been increasingly adopted for HAR since it offers integrated representation learning capabilities. There is not yet a consensus on the gold standard feature representation for HAR using wearables [31]. However, they can broadly be broken down into three categories: *(i)* Statistical features, e.g., mean and standard deviation of raw data [38]; *(ii)* Distribution-based representations, including those based on empirical cumulative distribution functions of the raw data [30]; and *(iii)* Learned features, that directly utilize the data itself to derive representations, involving supervised or unsupervised learning and dimensionality reduction techniques [31, 59, 63].

2.1 Feature Engineering

More traditional approaches aim at finding compact representations for the sensor data. A wealth of heuristics, both in the time and frequency domain, have been developed towards extracting such representations [23]. The distribution-based representations, involving the quantiles of the (inverted) empirical cumulative density function of sensor windows present the state-of-the-art for conventional feature extraction [30]. Improvements to this representation technique include adding structure [43] and specializing window lengths for human activity recognition [46]. One type of learned features includes dimensionality reduction techniques such as the principal component analysis, which has been utilized as a feature for activity recognition [63].

2.2 Feature Learning

Methods for feature learning aim at optimizing dedicated objective functions in order to learn useful representations from raw sensor data. Supervised learning requires annotated datasets and does not explicitly differentiate between the representation learning and classification steps of the ARC. As such, feature extraction (learning) is performed implicitly by the particular model that is being trained. Convolutional networks have been utilized to classify multi-variate sensor data [29, 79, 81]. Leveraging the temporal correlations in wearable sensor data at modeling level, (variants of) recurrent neural networks (RNNs) have been applied previously [27, 29, 80]. Guan *et al.* [27] setup ensemble training with recurrent networks in order to improve classification performance. Continuous attention mechanisms over both the sensory and time channels have been applied in [80] in order to learn both the 'important' timesteps as well as channels in windows of data. A combination of both convolutional and recurrent networks is utilized by Ordonez *et al.* [59] where 2D convolutional features are fed to a long short-term memory [36] network to perform activity recognition. Subsequent work on this architecture to improve performance includes temporal attention [55]. More recently, Transformer [75] encoder networks have been utilized for activity recognition [32, 50]. They model sequential information via the use of self-attention mechanisms and utilize only dense layers. As they dispense of both convolutions and recurrence, they are more parallelizable and require less time to train [75]. Overall, supervised learning remains the de facto approach towards activity recognition in recent years and also comprises the state-of-the-art.

Prior work into unsupervised learning has involved RBMs [63] and variants of autoencoder models [31, 74]. In [31] the authors investigate the role of representations in activity recognition, by evaluating different features on a common classification backend. Varamin *et al.* [74] define HAR as a set prediction problems within the autoencoder setup. A more recent development includes the use of self-supervised learning for unsupervised

pre-training. Along with Haresamudram *et al.*'s convolutional autoencoder (CAE) [31], these self-supervised learning approaches form our unsupervised baselines.

2.3 Self-Supervised Learning

Self-supervised learning involves defining a 'pretext' task from the data itself, that provides supervisory signals beneficial for downstream tasks. Numerous such tasks have been designed for both computer vision and natural language processing problems. For example, colorization [86] involves a model that is trained to colorize grayscale images. Predicting image rotations [25] allows the model to learn concepts of the objects in the images, such as their location, pose and type. This approach is shown to be beneficial towards a series of downstream tasks including classification and object recognition. Other approaches such as in-painting [61] mask portions of the image, and predicting relative positions of patches [19] have also been shown to perform well for representation learning. Self-supervision involving spatio-temporality has been effectively applied for learning representations for videos. Identifying the odd or unrelated video sub-sequence [22], sequence verification of temporal order [53], and learning the arrow of time [78] have been used as pretext tasks.

In the case of natural language processing, context-based self-supervised learning has been applied to obtain word and sentence level embeddings for approaches such as Word2Vec [52], GloVe [62] and universal sentence encoder [9]. Most recently, transformer-based [75] approaches such as GPT [65] and BERT [18] have been developed. BERT involves predicting masked tokens from surrounding context followed by predicting the next sentence. GPT similarly learns useful representations by pre-training a language model and then discriminatively fine-tuning on each specific task.

A more related field to activity recognition from wearable sensing involves speech and audio processing. In [77], Wang *et al.* perform self-supervision by reconstructing randomly masked out groups of timesteps as well as frequencies on log mel filterbank energies by using bidirectional recurrent encoders. In a similar vein, Liu *et al.*[47] utilize reconstruction of time, channel and magnitude altered inputs for self-supervision (see also [48, 88].) Autoregressive predictive coding has also been explored for speech self-supervision in [15, 17] and [16]. Another major family of self-supervised learning approaches involves contrastive learning. The intuition behind contrastive learning is to compare semantically similar (positive) and dissimilar (negative) pairs of data points, and to encourage the distance between similar pairs to be close while the distance between dissimilar pairs is encouraged to be more orthogonal [13]. This framework has resulted in excellent representation learning performance across domains such as computer vision [12, 33, 35], natural language [49], speech recognition [6, 58] and time-series data [39]. For example, Schneider *et al.* [6] utilize a contrastive loss on raw speech signals from a large unlabeled corpus to improve speech recognition performance on smaller labeled datasets. Contrastive Predictive Coding (CPC) [58], which is the focus of this work, performs contrastive learning over multiple future timestep predictions.

2.4 Self-Supervised Learning for HAR

Self-supervision for activity recognition from wearables has been explored from a multi-task learning context. In [69], eight data transformations were applied separately to the input data. These transformations were applied randomly, i.e. with 50% probability, and the task consisted of identifying if each transformation was applied or not in a multi-task setting. The architecture involved a transformation prediction network (TPN) consisting of 1D convolutional layers common to all tasks along with task specific fully connected layers. The learned TPN weights were used for representation learning, transfer learning and semi-supervised learning with limited labeled data. Multi-task self-supervised learning [69] utilizes accelerometer signals while our work is agnostic to sensor modalities. Therefore, it is applicable to a wider range of scenarios and environments.

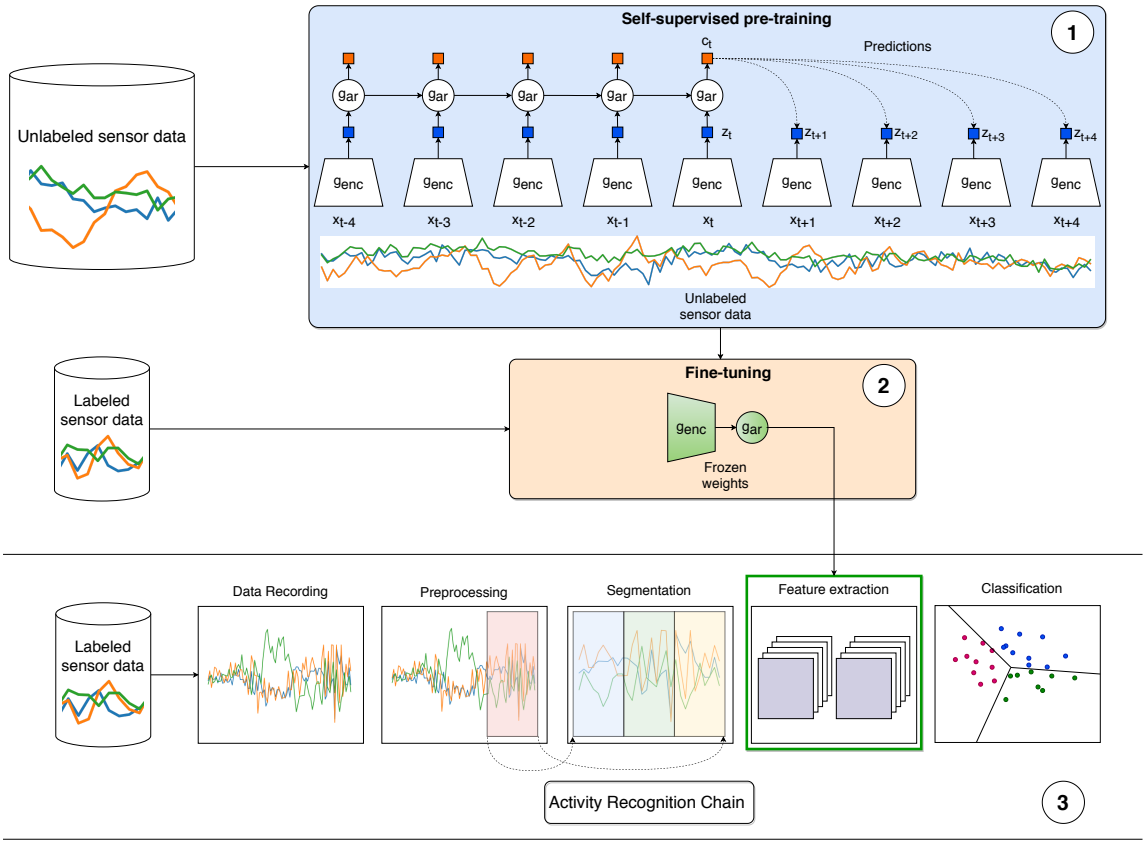

Fig. 1. Overview of proposed Human Activity Recognition framework based on Contrastive Predictive Coding (CPC) – details explained in main text.

A more recent work involves the use of Transformer [75] encoders to reconstruct randomly masked timesteps in windows of sensor data [32]. Here, sensor data at 10% of randomly chosen timesteps is masked, or set to zero. The transformer encoder layers are subsequently trained to reconstruct only the missing data and learned weights are used for activity recognition, transfer learning and fine-tuning with limited labeled data. While [32] utilizes both accelerometer and gyroscope data, it delivers mixed results relative to DeepConvLSTM [59] when fine-tuning the pre-trained weights using limited labeled data. In comparison, our work results in consistent improvement in performance on all benchmark datasets over DeepConvLSTM and thereby reduces reliance on the quantity of labeled data required for training systems.

3 SELF-SUPERVISED PRE-TRAINING WITH CONTRASTIVE PREDICTIVE CODING

In this paper, we introduce the Contrastive Predictive Coding (CPC) framework to human activity recognition from wearables. Fig. 1 outlines the overall workflow, which includes: *(i) pre-training* (part 1 in Fig. 1), where unlabeled data are utilized to obtain useful representations (i.e., learn encoder weights) via the pretext task; and, *(ii) fine-tuning*, which involves performing activity recognition on the learned representations using a classifier (part 2 in Fig. 1). During pre-training, the sliding window approach is applied to large quantities of unlabeled

data to segment it into overlapping windows. They are utilized as input for self-supervised pre-training, which learns useful unsupervised representations. Once the pre-training is complete, weights from both g_{enc} and g_{ar} are frozen and used for feature extraction (part 2 in Fig. 1). This corresponds to the feature extraction step in the ARC (part 3 in Fig. 1).

The frozen learned weights are utilized with the backend classifier network (see Sec. 3.2), a three-layer multi-layer perceptron (MLP), in order to classify windows of labeled data into activities. This corresponds to the classification step in the ARC. The learned weights from CPC are frozen and only the classifier is optimized on (potentially smaller amounts of) labeled datasets. The resulting performance directly indicates the quality of the learned representations.

In what follows, we first detail our Contrastive Predictive Coding framework as it is applied to HAR, and then describe the backend classifier network used to evaluate the unsupervised representations.

3.1 Contrastive Predictive Coding

Contrastive predictive coding (CPC) involves predicting future timesteps in the latent space using autoregressive modeling. The intuition behind such a task is to extract high-level information from the signal while discarding the more local noises [58]. Predicting one future timestep is an established approach in signal processing [5, 21] and exploits the local smoothness of signals. However, as the model is forced to predict farther into the future, it needs to infer more global structure [58]. This results in the incorporation of long-term temporal characteristics into the representation learning itself. We hypothesize that this process is beneficial towards learning effective representations.

The architecture for CPC is detailed in part 1 of Fig. 1. As detailed in [58], a non-linear encoder g_{enc} is utilized to map a window of sensor data x_t to a sequence of latent representations given by $z_t = g_{enc}(x_t)$. Subsequently, an autoregressive model g_{ar} is used to summarize all $z_{\leq t}$ into a latent space, providing a context latent representation $c_t = g_{ar}(z_{\leq t})$. Instead of predicting the future samples x_{t+k} directly using a generative model, a density ratio which preserves the mutual information between x_{t+k} and c_t is modeled as follows:

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}. \tag{1}$$

The density ratio f is unnormalized and a simple log-bilinear model is used for scoring:

$$f_k(x_{t+k}, c_t) = exp(z_{t+k}^T W_k c_t) \tag{2}$$

A linear transformation $W_k c_t$ is used to predict the k timesteps, with a separate W_k being used for each step. Using the density ratio $f_k(x_{t+k}, c_t)$ and inferring z_{t+k} using an encoder relieves the network from modeling the high dimensional distribution x_{t_k} and instead allows for sampling $p(x)$ or $p(x|c)$ directly using Noise Contrastive Estimation (NCE). We utilize the InfoNCE loss detailed in [58] to update the network parameters:

$$L_N = -\mathbb{E}_X \left[log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right] \tag{3}$$

Here, one positive sample from $p(x_{t+k}|c_t)$ and $N-1$ negative samples from the 'proposal' distribution $p(x_{t+k})$ are utilized for optimizing the loss. The pre-training using Eq. 3 involves classifying the positive sample correctly from a combined set of one positive and $N-1$ negative samples. After the pre-training is complete, we utilize g_{enc} and g_{ar} weights to extract the representations.

From an implementation standpoint, given a window of T timesteps and k future step predictions, we pass the entire window through the encoder. We choose a random timestep $t \in [0, T-k]$ and predict k subsequent timesteps. Sensor data present between 0 and t are utilized as input to the autoregressive network g_{ar}, which

results in the context representation c_t. For each future timestep prediction, the negatives comprise of same timestep from all other windows in the batch.

3.2 Backend Classifier Network

In Sec. 3.1, we detailed the contrastive predictive coding framework, which constitutes the Feature Extraction step in the ARC. The subsequent step in the chain – Classification – involves the evaluation of the representations learned via self-supervision. The classification backend consists of a 3 layer feedforward network with the first two layers comprising of 256 and 128 units respectively. The last layer constitutes the softmax layer, the size of which equals the number of classes in the dataset. Between the layers, we apply batch normalization [40], the ReLU activation function [56] and dropout [70] with p=0.2. This network is identical to the classifier used in Haresamudram *et al.*[32], thereby making the results obtained directly comparable.

4 HUMAN ACTIVITY RECOGNITION BASED ON CONTRASTIVE PREDICTIVE CODING

In the previous section we have introduced our representation learning framework for movement data based on contrastive predictive coding. This pre-training step is integrated into an overarching human activity recognition framework, that is based on the standard Activity Recognition Chain (ARC) [8]. Addressing our general goal of deriving effective HAR systems from limited amounts of annotated training data, as it is a regular challenge in mobile and ubiquitous computing settings, we conducted extensive experimental evaluations to explore the overall effectiveness of our proposed representation learning approach.

In what follows we provide a detailed explanation of our experimental evaluation, which includes descriptions of: *i)* Application scenarios that our work focuses on; *ii)* Implementation details; *iii)* Evaluation metrics used for quantitative evaluation; and *iv)* Overall experimental procedure. Results of our experiments and discussion thereof are presented in Sec. 5.

4.1 Application Scenarios

In our work, we focus on studying self-supervised pre-training primarily for locomotion style activities such as walking, running, and sitting, as these broadly represent a significant portion of activities performed daily and are of significant interest for the mobile and ubiquitous computing community [1]. The effectiveness of our CPC-based pre-training is evaluated on four representative benchmark datasets, which mainly contain data recorded from smartphones (see Tab. 1 for an overview). In particular, recording movement data from smartphones has the advantage of being unobtrusive, inexpensive, and available to large portions of the population. This also allows for easier in-the-wild data collection. The chosen datasets cover diverse participants, environments of study, and data collection protocols for body-worn sensors including accelerometers and gyroscopes and as such are representative for the targeted application scenarios. They were also studied in previous works on self-supervised learning in human activity recognition [32, 69]. Additionally, we include the USC-HAD dataset into our explorations, as an example for an activity recognition scenario using body-worn sensing platforms that are not smartphones.

Unless specified differently, dataset splits for training, validation and testing are performed based on participant IDs. 20% of the participants are randomly chosen to form the test dataset. Of the remaining data, 20% are randomly chosen once again to form the validation set, while the rest are utilized for training. Detailed descriptions of the datasets are given below.

4.1.1 Mobiact. Movement data from inertial measurement units were collected using a Samsung Galaxy S3 smartphone placed freely in a trouser pocket. The dataset covers eleven activities of daily living and four types of falls [10]. We used v2 of the dataset and subset data that cover only daily living activities, resulting in a total

Table 1. Overview of the datasets used in the evaluation.

Dataset	# of users	# Activity classes	Recording Device	Activities
Mobiact	61	11	Samsung Galaxy S3	Sitting, walking, jogging, jumping, stairs up, stairs down, stand to sit, sitting on a chair, sit to stand, car step-in, and car step-out
Motionsense	24	6	iPhone 6S	Walking, jogging, going up and down the stairs, sitting and standing
UCI-HAR	30	6	Samsung Galaxy S2	Standing, sitting, walking, lying down, walking downstairs and upstairs
USC-HAD	14	12	MotionNode platform	Walking - forward, left, right, upstairs, and downstairs, running forward, jumping, sitting, standing, sleeping, and riding the elevator up and down

of 61 participants. The activities include: sitting, walking, jogging, jumping, stairs up, stairs down, stand to sit, sitting on a chair, sit to stand, car step-in, and car step-out.

4.1.2 Motionsense. The Motionsense dataset [51] consists of 24 participants of different ages, gender, weight and height. The dataset was collected to propose a representation learning model that offers flexible and negotiable privacy-preserving sensor data transmission. The data were collected using an iPhone 6s and comprises accelerometer, gyroscope and attitude information. The activities covered include walking, jogging, going up and down the stairs, sitting and standing.

4.1.3 UCI-HAR. The UCI-HAR dataset [4] contains data collected from 30 participants using a waist-mounted Samsung Galaxy S2 smartphone. The subjects performed six activities including standing, sitting, walking, lying down, walking downstairs and upstairs (the transition classes are not included). We use raw data from both the accelerometer and gyroscope in our study (also called the HAPT dataset [1]).

4.1.4 USC-HAD. The USC-HAD dataset [85] was collected on the MotionNode sensing platform and consists of data from 14 subjects. Twelve activities were recorded, including walking–forward, left, right, upstairs, and downstairs–, running forward, jumping, sitting, standing, sleeping, and riding the elevator up and down. Following the protocol from Haresamudram *et al.* [31], participants $1-10$ form the training set, while participants 11 and 12 form the validation set, and participants 13 and 14 comprise the test set.

4.2 Implementation Details

We implemented our framework using PyTorch [60]. Source code will be shared when the paper is published. In what follows we provide details of parameter choices for the overall processing framework, which shall allow the reader to replicate our experiments.

4.2.1 Data Preparation. We use raw accelerometer and gyroscope data from the benchmark datasets detailed in Sec. 4.1. As deep networks can effectively learn abstract representations from raw data itself, we perform no further filtering / denoising on the datasets [45]. The datasets have different sampling rates and we downsample them to 30Hz in order to maintain uniformity. Further, we also normalize the individual channels of the train dataset split to zero mean and unit variance. These means and variances are subsequently applied to the validation

[1] http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions

and test splits. Similar to prior works, sliding window based segmentation is applied to obtain windows of 1s with 50% overlap between subsequent windows [27, 29, 32].

4.2.2 CPC Pre-Training. The pre-training is performed for 150 epochs with the learning rate being tuned over $\{1e-3, 5e-4\}$ and $k \in \{2, 4, 8, 12, 16\}$. The network weights are optimized using the Adam [41] optimizer. For the g_{enc} network, we primarily evaluate a 1D Convolutional Encoder, containing three blocks with 1D convolutional layers of 32, 64 and 128 channels respectively with a kernel size of 3. Each block consists of a 1D convolutional layer with reflect padding, followed by the ReLU activation function and dropout with p=0.2. For the autoregressive network g_{ar}, we utilize a two-layer gated recurrent units (GRU) network [14] containing 256 units and dropout with p=0.2. The prediction networks, W_k, are linear layers with 128 units.

4.2.3 Activity Recognition. The classification backend is trained with labeled data for 150 epochs using cross entropy loss. Learning rate is tuned over $\{5e-4, 1e-4\}$ and is decayed by a factor of 0.8 every 25 epochs. The network parameters are updated using the Adam [41] optimizer.

4.3 Performance Metric

The test set mean F1-score is utilized as the primary metric to evaluate performance. The datasets used in this study show substantial class imbalance and thus experiments require evaluation metrics that are less affected negatively by such biased class distributions [64]. The mean F1-score is given by:

$$F_m = \frac{2}{|c|} \sum_{c} \frac{prec_c \times recall_c}{prec_c + recall_c} \tag{4}$$

where $|c|$ corresponds to the number of classes while $prec_c$ and $recall_c$ are the precision and recall for each class.

4.4 Experimental Procedure

As the main focus of this work is on deriving effective HAR representations without relying on labels, we study the self-supervised pre-training from two perspectives:

(1) First, we compute the activity recognition performance, which describes the quality of the representations learned during pre-training. It indicates the raw ability of the representations to be discriminative towards the activities under study.
(2) Then, we extend this evaluation to scenarios with limited availability of annotated data and compare performance of the learned weights to end-to-end training. This considers practical situations where very limited labeling is possible from users after the deployment of recognition systems.

Put together, these evaluations provide a rounded understanding of the effectiveness of the self-supervised pre-training and its performance for representation learning.

5 RESULTS AND DISCUSSION

5.1 Activity Recognition

We perform CPC-based self-supervised pre-training and integrate the learned weights as a feature extractor in the activity recognition chain. In order to evaluate these learned representations, we compute their performance on the classifier network (Sec. 3.2). The performance obtained by CPC is contrasted primarily against previous unsupervised approaches including multi-task self-supervised learning [69], convolutional autoencoders [31], and masked reconstruction [32]. For reference, we also compare the performance relative to the supervised baseline–DeepConvLSTM [59]– and a network with the same architecture as CPC, albeit trained end-to-end from scratch. Once the model was pre-trained using CPC, the learned weights (from both g_{enc} and g_{ar}) were

Table 2. We compare the representation learning performance of the proposed approach against both supervised learning and to state-of-the-art unsupervised learning baselines. Performance for the approaches with † have been taken from [32]. CPC (end-to-end) refers to the setup wherein the same network architecture as CPC (1D Conv Encoder) is utilized. However, all weights of the network are initialized randomly and trained end-to-end from scratch using labeled data. Results marked in **green** correspond to the best performing models including both supervised and unsupervised learning approaches. For easy comparison, results in **bold** include the best performing unsupervised learning techniques.

Approach	Method type	Mobiact	Motionsense	UCI-HAR	USC-HAD
DeepConvLSTM† [59]	Supervised	82.40	85.15	**82.83**	44.83
CPC (end-to-end, 1D Conv Encoder)	Supervised	**83.68**	86.66	79.79	49.09
Multi-task self-supervised learning† [69]	Unsupervised	75.41	83.30	80.20	45.37
Convolutional autoencoder† [31]	Unsupervised	79.58	82.50	80.26	48.82
Masked reconstruction† [32]	Unsupervised	76.81	88.02	**81.89**	49.31
CPC (1D Conv Encoder)	Unsupervised	**80.97**	**89.05**	81.65	**52.01**

frozen and used with the classifier network. Labeled data was utilized to train the classifier network using cross entropy loss and the test set mean F1-score was detailed in Tab. 2.

We first compare the performance of the CPC-based pre-training to state-of-the-art unsupervised learning approaches. We note that all unsupervised learning approaches are evaluated on the same classifier network (Sec. 3.2), which is optimized during model training for activity recognition. On Mobiact, Motionsense and USC-HAD, CPC-based pre-training outperforms **all** state-of-the-art unsupervised approaches. For UCI-HAR, the performance is comparable to masked reconstruction. This clearly demonstrates the effectiveness of the pre-training thereby fulfilling one of the goals of the paper – which is to develop effective unsupervised pre-training approaches. It also validates our hypothesis that explicitly incorporating temporality at the representation level itself is beneficial towards learning useful representations.

Relative to DeepConvLSTM, i.e., the supervised baseline, the proposed approach shows increased performance by approximately 4% on Motionsense and over 7% on USC-HAD. On UCI-HAR and Mobiact, the performance is comparable to the 1D Conv Encoder-based CPC model (last row). Relative to end-to-end training on a network with the same architecture as CPC, using the pre-training results in improved performance for Motionsense, which obtains the highest F1-score across both supervised and unsupervised learning approaches. In contrast, CPC-based pre-training performs comparably on UCI-HAR, while it shows reduced results for both USC-HAD and Mobiact with the difference being around 3% at worst. This strong performance showcases the quality of the weights learned as they outperform end-to-end training on some of the datasets, even when the number of trainable parameters are only a fraction (as only the final classifier parameters are updated).

5.2 Semi-Supervised Learning on Limited Labeled Data

Here we consider a very important scenario where only a small portion of the collected data is labeled. This is a practical and rather common situation in mobile and ubiquitous computing scenarios as real-world deployment may allow for acquiring only a small labeled dataset from the users with minimal interruptions. This scenario is important because data collection is typically straightforward yet annotation is often challenging.

As in Sec. 5.1, we first pre-train the network using the CPC task and utilize the weights learned on both the encoder and autoregressive network for feature extraction. The classifier network is initialized randomly and trained from scratch whereas the learned weights are frozen. For each dataset, we utilize the entire dataset (unlabeled) for pre-training. During the fine-tuning stage, we randomly sample x labeled samples per class where $x \in \{1, 2, 5, 10, 25, 50, 100\}$ and train the classifier from scratch. We perform five runs and plot the test set

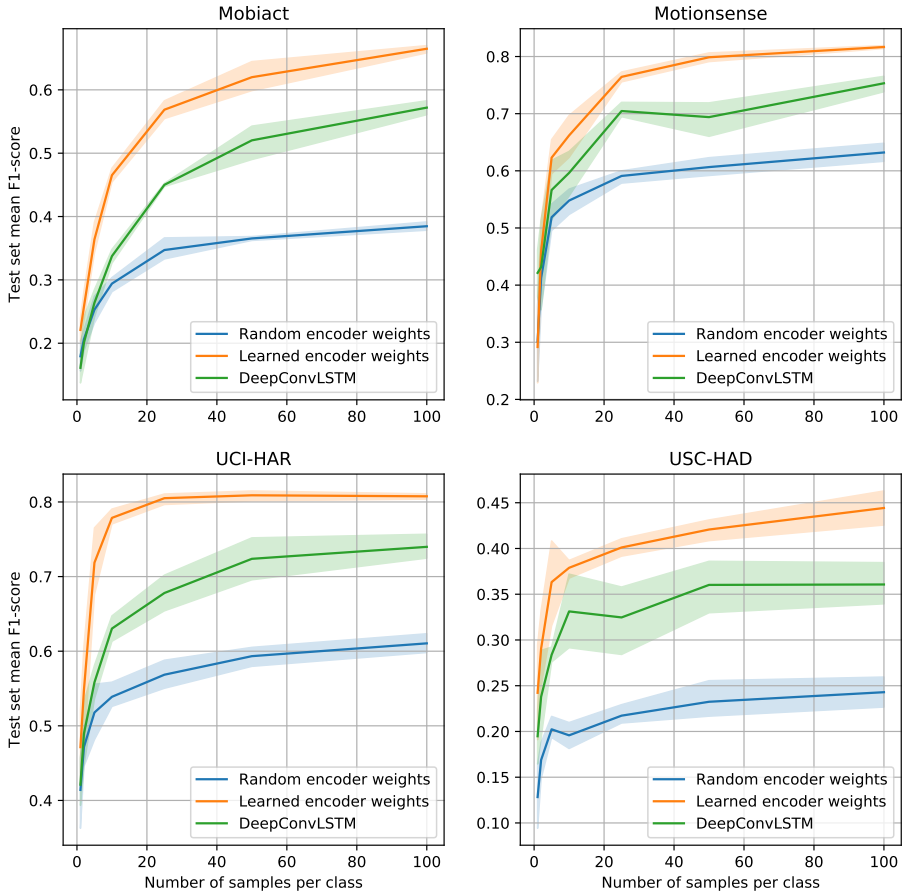

Fig. 2. Semi-supervised learning on limited labeled data. The network is first pre-trained in an unsupervised manner using CPC, and the learned weights are frozen. The backend classifier is initialized randomly and trained from scratch using {1, 2, 5, 10, 25, 50, 100} randomly sampled labelled frames. We perform five runs and report the test set mean f1-score. The curve in orange corresponds to CPC where the weights are learned while the blue line refers to the scenario with the randomly initialized feature extractor. The supervised baseline, DeepConvLSTM, is depicted in green. We observe significant improvements over DeepConvLSTM on all datasets.

mean F1-score in Fig. 2 for the 1D Conv encoder models detailed in Tab. 2. The performance of the CPC-based pre-training technique is contrasted against the DeepConvLSTM network, which is the supervised learning baseline. In order to make the contribution of the learned weights clearer, we also compute the performance when the feature extractor is initialized randomly and frozen during fine-tuning.

On Mobiact, which contains the largest number of participants, we immediately notice the improvement in performance even when only one labeled sample per class is available for fine-tuning. Relative to DeepConvLSTM,

the performance of the proposed method is consistently higher by over 5% throughout. The difference is even clearer over the random initialized feature extractor, as the pre-training results in an improvement of over 15% when more than ten samples are available per class.

In the case of Motionsense, DeepConvLSTM outperforms CPC when there is only one labeled samples per class. Beyond that, CPC shows increased performance, resulting in a maximum improvement of approx. 10% when 50 labeled samples are available. The learned weights result in sustained improvements of around 15% over the random weights when more than ten labeled samples per class are available.

For UCI-HAR, CPC results show increased performance over DeepConvLSTM even when just one labeled sample per class is available. The difference peaks at over 10% when we have access to ten samples per class. Similarly, CPC shows improved performance even when there is only one sample per class for USC-HAD. As more labeled data is available, the performance of CPC rises further over DeepConvLSTM. There is also a consistent difference in performance over using a randomly initialized feature extractor.

The consistently improved performance over end-to-end training with DeepConvLSTM makes a compelling indicator that incorporating temporal characteristics into representation learning results in improved representations. We also note that the largest unlabeled dataset, Mobiact, demonstrates the most significant improvement in performance over random weights due to the self-supervised pre-training. Thus, using an even larger unlabeled dataset might further improve the quality of the representations learned using self-supervision. Additionally, the confidence interval for the proposed technique, CPC, is generally narrower than DeepConvLSTM for all datasets. This indicates that the pre-training makes the activity recognition performance more robust to variability in the input windows.

6 DISCUSSION

The main hypothesis of our work is that explicitly targeting the temporal characteristics of movement data for representation learning is beneficial towards learning discriminative features for human activity recognition using body-worn sensors. We accomplish this by predicting multiple future timesteps using contrastive learning. Such a pre-training scheme fits directly into the traditional activity recognition chains at the feature extraction step, allowing for easy integration into activity recognition workflows. In what follows, we first analyze contrastive predictive coding as it is applied to HAR and derive insights on the performance of the pre-training. Then we look at the practical implications of our work on existing and future applications in mobile and ubiquitous computing and outline the research agenda related to representation learning for human activity recognition.

6.1 Analyzing Contrastive Predictive Coding

Contrastive predictive coding involves correctly identifying the positive sample from distractors for k future timesteps. In this section, we design experiments to better understand the pre-training process. We begin by studying the encoder (g_{enc}) and evaluate different choices, including feedforward, convolutional and recurrent networks and analyze how difficult the pretext task training is, and how it affects downstream activity recognition. Subsequently, we also analyze the number of steps to predict during the pre-training. Finally, we examine if all the weights learned during self-supervision are useful, by selectively utilizing only portions of the learned weights. These experiments help us to build better self-supervised models.

6.1.1 Choice of encoder. The encoder g_{enc} is used to map samples from the input data into a latent space on which future predictions are made. Therefore, the architecture of g_{enc} has impact on the pre-training as well as the subsequent activity recognition. We study three categories of networks for g_{enc} and report the downstream activity recognition performance in Fig. 3:

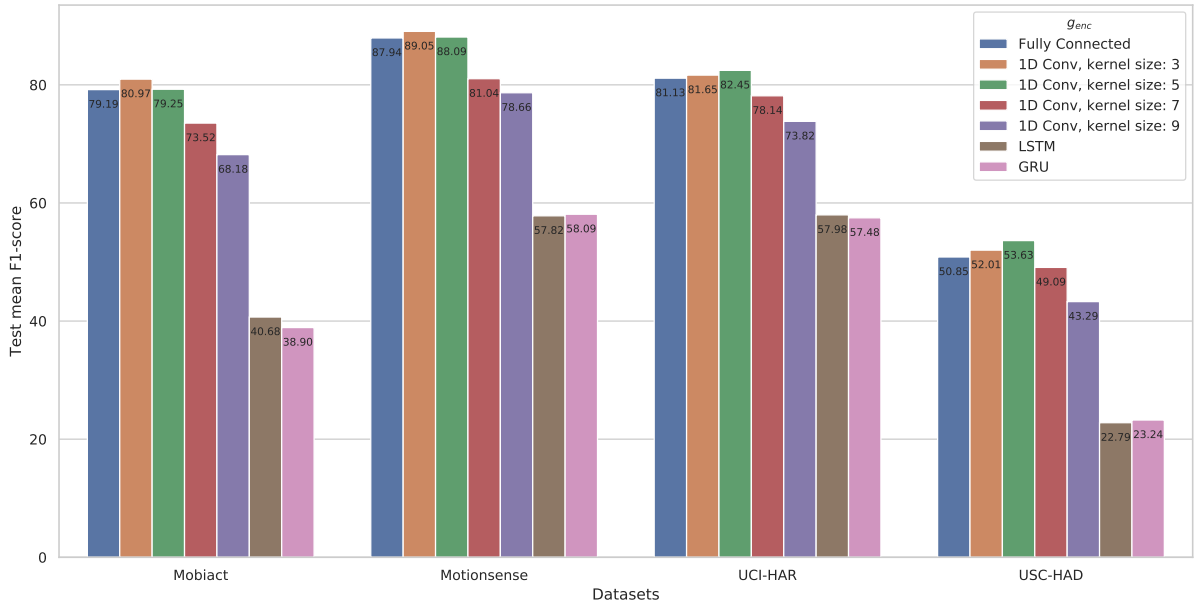

Fig. 3. Studying the effect of using different g_{enc} networks for CPC pre-training: we utilize three categories of encoder networks including fully connected layers, 1D convolutional layers with kernel sizes $\in \{3, 5, 7, 9\}$, and recurrent layers for pre-training. The learned g_{enc} and g_{ar} weights are frozen and used to extract representations for classifying activities. We observe that the fully connected encoder or the 1D convolutional encoders with smaller kernel sizes (i.e. 3 or 5) demonstrate the highest activity recognition performance.

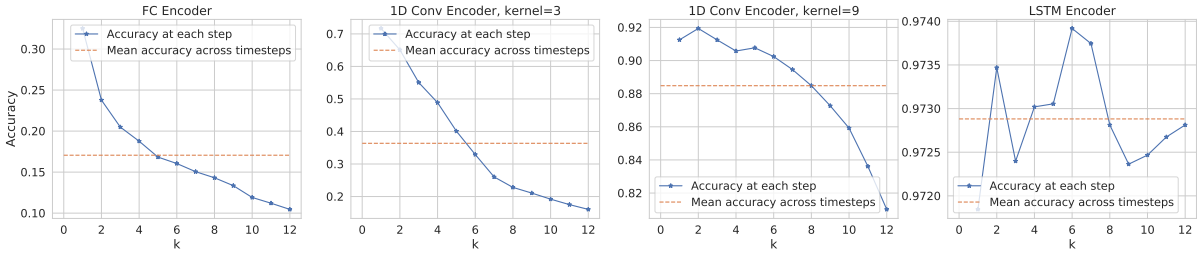

Fig. 4. Accuracy of correctly predicting the positive sample from negatives in the contrastive loss, across multiple future timesteps ($k = 12$) for test split of the Mobiact dataset.

Fully Connected: which consists of a feedforward network comprising of three fully connected layers with 32, 64 and 128 units, along with dropout [70] of p=0.2 and the ReLU activation function [56] applied between consecutive fully connected layers.

1D Convolutional: containing three 1D convolutional blocks with 32, 64 and 128 channels respectively. Each block consists of a 1D convolutional layer with reflect padding, followed by the ReLU activation and dropout with p=0.2. We vary the kernel sizes $\in \{3, 5, 7, 9\}$.

Recurrent: consisting of one layer long short-term memory (LSTM) or GRU network with 128 units.

For each type of encoder, we utilize the same training settings as used in Tab. 2 for the 1D Conv Encoder.

Fig. 3 shows that the choice of the encoder network has a significant impact on the activity recognition performance. For the 1D Conv. encoders, we see that having shorter kernel sizes (i.e., 3 or 5) is generally more preferable to longer filter lengths such as 7 or 9. The convolution operation over the input data results in overlaps over the future timesteps to be predicted. Correctly predicting the first few future timesteps, which potentially fall under the overlap, thereby becomes trivial. This results in reduced activity recognition performance as the pretext task becomes ineffective. It also explains how the activity recognition performance reduces as the kernel size is increased.

Larger overlaps on the future timesteps mean that the overlapped steps are easier to predict, reducing the discriminative ability of the representations learned. This can be clearly seen in Fig. 4 where the 1D Conv. Encoder with a kernel size of 9 has a considerably higher average accuracy of predicting the positive sample relative to using the encoder with a kernel size of 3. It indicates that the pretext task is easier to solve. Correspondingly, the downstream activity recognition performance for kernel size 9 is around 13% lower than the F1-score for kernel size 3. This effect is more prominent for the recurrent encoders as the input data is parsed step-by-step and the output at any given timestep is dependent on previous data. Thus, the pretext task becomes trivial to solve, resulting in a reduction of around 42% compared to the 1D Conv. Encoder with kernel size 3.

The fully connected (FC) encoder represents the other extreme where there is no overlap across the window of data. The average accuracy of predicting the future timesteps is lower relative to the 1D Conv. Encoder with kernel size 3. This results in reduction of nearly 2% for activity recognition. From Fig. 4 and Fig. 3 we can see that the difficulty of solving the pretext task has significant impact on the activity recognition performance. If the task is trivial (as in the case of using large kernel sizes or recurrent encoders), the downstream recognition task also suffers correspondingly. Therefore, the encoder architecture must be carefully considered before being utilized for pre-training, as rendering the pretext task trivial results in poor representations.

6.1.2 Number of Future Steps to Predict. CPC learns effective representations by predicting multiple future timesteps, rather than just the next timestep. This allows the model to learn features that incorporate long term temporal characteristics. It also leads us to the question: *How many future timesteps need to be predicted to learn useful representations?* For the CPC-based pre-trained models in Tab. 2, we study the activity recognition performance for increasing values of $k \in \{2, 4, 8, 12, 16\}$, and report the test mean F1-scores.

As can be seen in Fig. 5, increasing the number of predicted timesteps generally results in improved activity recognition. For Mobiact, Motionsense and UCI-HAR, the performance peaks when k=12. Beyond that, the performance starts reducing again. The trend is similar for USC-HAD, with the peak occurring at k=8. Thus, predicting approximately 400ms into the future results in the best performance for Mobiact, Motionsense and UCI-HAR, whereas USC-HAD requires predicting around 270ms into the future.

As discussed in Sec. 3.1, the input to the autoregressive network g_{ar} is the sensor data between timesteps 0 and t. With increasing k, the input to g_{ar} becomes shorter thereby making it more difficult to learn the context representation c_t which can reliably predict many subsequent future timesteps. We hypothesize that this results in the reduction in performance after obtaining the peak. Looking at the 1D Conv. Encoder with kernel size=3 in Fig. 4, we can also observe that predicting multiple timesteps into the future is considerably more challenging than predicting the immediate future. The farther into the future we predict, the lower the mutual information between what the model already knows (i.e., the context representation) and the target, making it more difficult to correctly predict the positive sample. The accuracy of prediction drops almost exponentially indicating the difficulty of predicting far into the future. However, Fig. 5 also shows that predicting the immediate future (i.e., $k = 2$ in Fig. 5) results in poorer representations as the model only learns short term noises. Therefore, predicting multiple future timesteps is vital towards extracting the long-term temporal signal which exists in the sensor data windows.

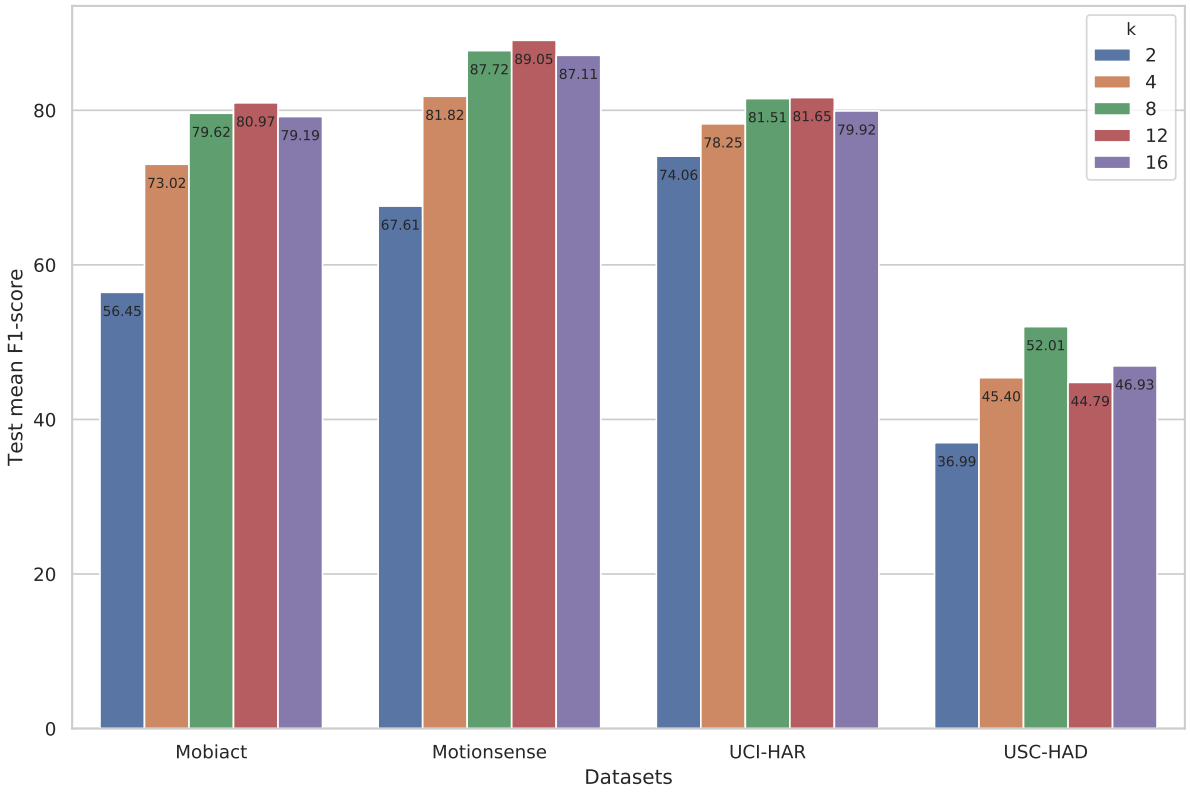

Fig. 5. Studying the effect of the number of future timestep predictions (i.e. k) during CPC pre-training for activity recognition: we utilize the 1D convolution encoder with a kernel size=3 as g_{enc} and vary $k \in \{2, 4, 8, 12, 16\}$ for pre-training. Subsequently, the learned g_{enc} and g_{ar} weights are frozen and used as a feature extractor for activity recognition. We observe that predicting multiple future timesteps ($k \geq 8$) is more advantageous than predicting the near future, i.e. $k \in \{2, 4\}$.

6.1.3 Which Pre-Trained Weights Should be Used? During activity recognition, the learned weights from both g_{enc} and g_{ar} were frozen and utilized with the classifier network (Fig. 1). This brings up the question: *Which learned weights are useful for activity recognition?* To answer this, we progressively utilize learned weights from fewer encoder layers during the classification and report the resulting test mean F1-score in Fig. 6.

The default configuration involves freezing learned weights from both g_{enc} and g_{ar}, as done for Tab. 2. This corresponds to $g_{enc_{\leq 3}} + g_{ar}$ in Fig. 6. $g_{enc_{\leq 3}}$ utilizes frozen learned weights from all three encoder layers, while the autoregressive network g_{ar} is initialized randomly and trained with the classifier. On a similar vein, $g_{enc_{\leq 2}}$ uses frozen learned weights from the first two encoder layers while the third layer and the autoregressive network g_{ar} are both trained with the classifier.

From Fig. 6 we can see that utilizing the default configuration results in the best activity recognition performance for Motionsense and USC-HAD. For these datasets, using learned $g_{enc_{\leq 3}} + g_{ar}$ weights outperforms fully supervised training (purple bars in Fig. 6).

For Mobiact, the default configuration does not result in performance exceeding fully supervised training. However, using the learned $g_{enc_{\leq 2}}$ weights results in a mean F1-score of 85.22%, which is an improvement over

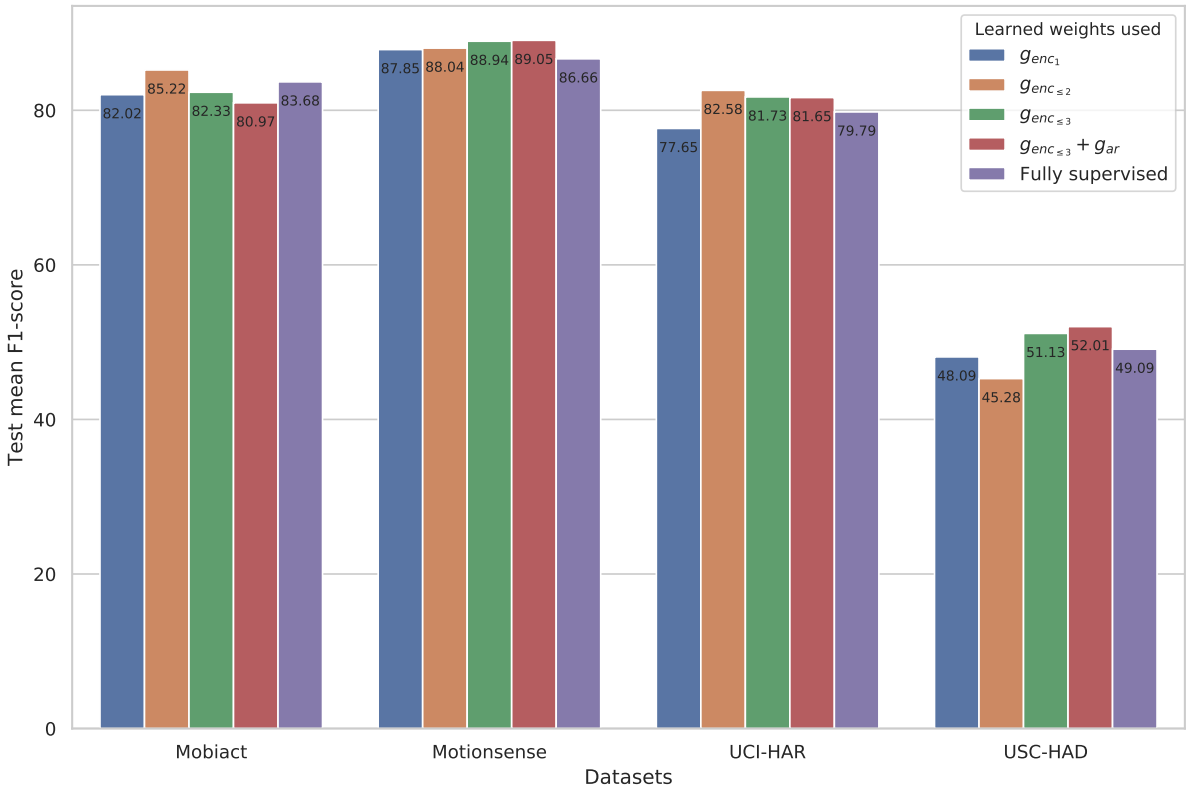

Fig. 6. Studying which learned weights are useful for activity recognition: using learned weights from the CPC model with 1D convolutional encoder and kernel size=3, we progressively utilize learned weights from fewer layers for activity recognition. The rest of the layers are initialized randomly and optimized during activity recognition. For example, using both g_{enc} and g_{ar} weights is denoted by $g_{enc_{\leq 3}} + g_{ar}$ while using only the first two encoder layer learned weights is shown as $g_{enc_{\leq 2}}$ (the rest of the network including the third encoder layer and g_{ar} are not frozen). We observe that using a subset of the learned weights can lead to improvement over using all learned weights.

fully supervised training. Similarly for UCI-HAR, the best performance is obtained when using the learned $g_{enc_{\leq 2}}$ weights. We observe a mean f1-score of 82.58%, which is higher than the best unsupervised representation learned by masked reconstruction (Tab. 2) and comparable to DeepConvLSTM, which exceeds all other approaches. Thus, we can conclude that using a subset of the weights learned via self-supervision could result in improved performance, often comparable to or better than supervised learning baselines such as DeepConvLSTM.

As mentioned in [58], either c_t (the output of g_{enc}) or z_t (the output of g_{ar}) can be utilized as the representation. If extra context from the past is useful for classification, then c_t can be used; else z_t may be more effective. We hypothesize that certain transition-style activities covered in Mobiact, such as stand to sit, sit to stand, car step-in and car step-out require the extra context provided by the learned weights at the encoder level, rather than the context representation (c_t), which may not be sufficiently encoding previous timesteps to be accurately classified. The confusion matrix in Fig. 7 illustrates this interpretation. For the transition-style activities in particular, using

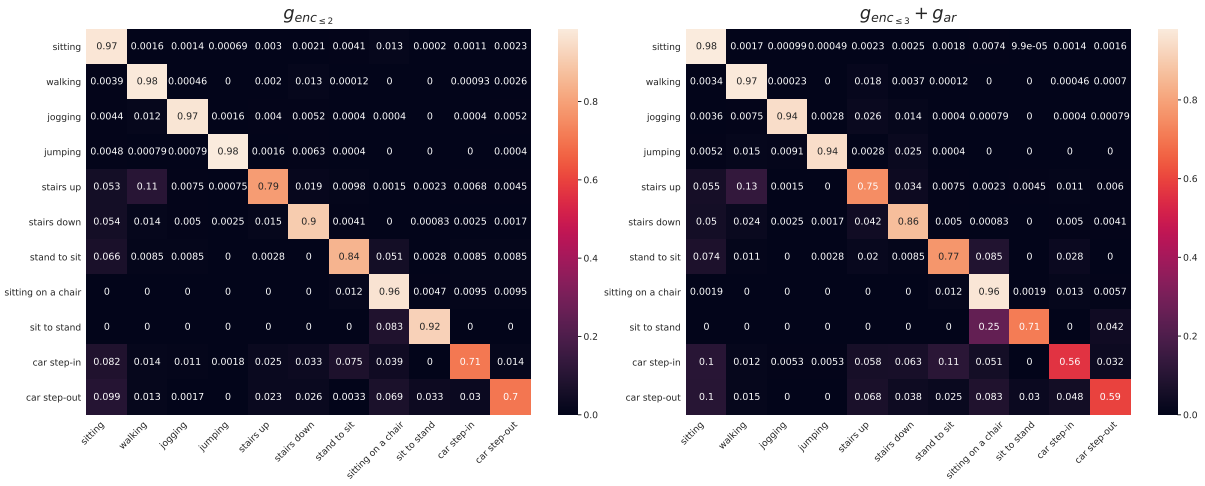

Fig. 7. Confusion matrix for the activity recognition performed after using learned weights of different layers for the Mobiact dataset. The pre-training was performed using 1D convolutional encoder with kernel size=3 and $k = 12$. $g_{enc_{\leq 2}}$ (left) utilizes learned weights from only the first two convolutional layers whereas $g_{enc_{\leq 3}} + g_{ar}$ (right) uses learned weights from all encoder layers as well as g_{ar}. For transition-style activities such as stand to sit, sit to stand, car step-in and car step-out, we see that the extra context provided by $g_{enc_{\leq 2}}$ results in improved performance over $g_{enc_{\leq 3}} + g_{ar}$.

the learned weights from only the first two encoder layers, i.e., $g_{enc_{\leq 2}}$ results in considerable improvements over using $g_{enc_{\leq 3}} + g_{ar}$.

6.1.4 Summary and Guidelines. Studying factors which affect CPC such as the choice of encoder, number of future step predictions and the usability of learned weights at various levels of the network, allow us to develop a set of insights for applying CPC to other mobile and ubiquitous computing scenarios as well:

(1) The encoder architecture has significant impact on the quality of the representations learned via CPC. Overlap over the prediction targets results in poorer representations, and thus, care must be taken to avoid setting up trivial prediction tasks.
(2) Predicting multiple future timesteps results in improved representations over predicting only the immediate future. Such a prediction scheme encodes global structure present in the time-series body-worn sensor data, which is beneficial towards learning richer representations.
(3) A subset of the learned weights could result in improved performance relative to using all learned weights. The representation used depends on the activities under study, as certain activities might require longer context to be accurately recognized. Thus, using learned weights from both the encoder and autoregressive networks maybe sub-optimal than using some parts of those weights.

6.2 Practical Value of CPC for Human Activity Recognition Tasks

6.2.1 Incorporating Temporality at Representation Level results in Improved Recognition Performance. The primary hypothesis of this work is that incorporating temporal characteristics into the representation learning process results in effective representation learning for time-series data. We note that previous unsupervised learning approaches involving Restricted Boltzmann Machines[63] and convolutional autoencoders [31] were trained using signal reconstruction. The more recent multi-task self-supervised learning utilizes binary predictions of randomly applying binary signal transformations in order to learn representations. These approaches however, do not

specifically utilize the temporal characteristics of the data. Masked reconstruction [32] leverages local temporal dependencies by reconstructing zeroed out timesteps from immediate local context. From Tab. 2 we observe that masked reconstruction, using just local low-level information, results in improvements in performance on three out of four of the benchmark datasets over the approaches that do not seek to incorporate any level of temporal information. This demonstrates the importance of using temporal information into the representation learning process itself.

Taking this insight further, we proposed to utilize Contrastive Predictive Coding, which learns to infer global structure in the time-series data, beyond just local low-level information and noise, by predicting multiple future timesteps. From our results in Tab. 2 we obtain strong evidence that such a process results in rich representations that more effectively capture the time-series nature of the body-worn sensor data.

6.2.2 CPC is Flexible and Generic. The CPC framework can be considered generic from three standpoints:

Target prediction setup: The temporal order of prediction is not significant and the pretext task can also be setup to predict the past timesteps or even randomly masked portions of the data (similar to masked reconstruction). However, care needs to be taken while choosing the encoder, the target predictions as well as the negative samples in order to avoid setting up a trivial task.

Architecture: The architecture of CPC also allows for replacing the g_{enc}, g_{ar}, and W_k networks with suitable alternatives. In this work we utilized standard architectures for simplicity. For example, in Sec. 6.1.1 we discussed various encoder networks and their effects. The autoregressive network g_{ar} used in this work is a 2-layer GRU. However, more powerful techniques for processing sequential information such as Transformers [75] or convolution layers could further improve the representation performance. Finally. the future timestep predictions are made by W_k, which in our setup consists of linear layers. They only observe the context representation in order to make the prediction. As explored in [68], alternates to this setup include 1-layer Transformer encoders and recurrent networks.

How negatives are sampled: In this work, the negatives for the contrastive loss are sampled from different windows present in the same batch of data. However, if we had prior knowledge of the downstream task, the negative sampling could be performed in such a way to incentivize relevant features for that particular task. For example, if the downstream task was person identification, we could sample negatives from the rest of the participants for each positive sample. The resulting features are likely to be discriminative towards participants rather than activities.

6.2.3 Self-Supervised Pre-Training Reduces Reliance on Large Amounts of Annotated Data. The lack of large-scale annotated body-worn sensor datasets for human activity recognition limits the application of complex supervised recognition systems. At the same time, the more straightforward data collection process allows for the potential collection of large amounts of unlabeled data. In this work, we demonstrated how pre-training on unlabeled data results in performance exceeding fully supervised training when limited labeled data is available. This allows for leveraging massive movement datasets such as the UK Biobank [20] to learn general movement representations, that can subsequently be fine-tuned for target tasks using labeled data. Such an approach can have serious impact in practical scenarios where very limited data can be collected from users after the deployment of recognition systems, for user adaptation or personalization purposes. By first pre-training on unlabeled data, these limited labels can be economically used to derive accurate recognition.

6.3 Research Agenda: Representation Learning for Human Activity Recognition

The focus of this work is towards utilizing unlabeled movement sensor datasets to perform representation learning. It is motivated by the limited size and the lack of variety in current annotated datasets. Instead, we can leverage massive movement datasets such as the UK Biobank [20] to first learn generic representations

that can be fine-tuned on task specific labeled data. However, specific applications of mobile and ubiquitous computing might require sensor placements which are different than the available source dataset. Therefore, a big challenge to address is the development of approaches that can still learn representations that can be effectively fine-tuned on data collected from sensors placed on different locations on the body. This will allow for improved performance in niche scenarios requiring unique sensor placements.

Recent approaches addressing the lack of annotated datasets involve extraction of virtual movement data from modalities such as mocap [37, 72] and videos [44], which contain a large number of participants and diverse set of actions and environments. Such techniques have the advantage of being capable of extracting large quantities of virtual movement data at arbitrary positions on the body. However, due to the artifacts introduced during the extraction process, the performance of recognizers trained on this data tend to be lower than those trained on real IMU data. Thus, developing effective representation learning approaches that can deal with the domain shift could be beneficial towards improved recognition.

7 CONCLUSION

Feature extraction plays a crucial role in human activity recognition (HAR) using body-worn movement sensors. Accordingly, the community has invested a lot of effort into developing new ways to extract meaningful representations from streams of sensor data that, ultimately, will lead to improved downstream recognition performance. Hand-crafted (engineered) features have long dominated the field, yet with the success of deep learning methods in the wider scientific community, the focus of the HAR community has shifted to learning effective representations directly from sensor data. End-to-end learning can only be employed with some limitations given the typically severe restrictions on the amount of *labelled* sample data that is available for training HAR models from scratch. As such, substantial effort has been devoted to explicit representation learning thereby not necessarily relying on large quantities of labeled sample data but rather making more economic and hence effective use of smaller labeled training sets, and especially aiming at exploiting unlabeled data, which are straightforward to collect in mobile and ubiquitous computing settings.

The approach presented in this paper follows the paradigm of explicitly learning data representations, which are then integrated as features into the standard activity recognition chain [8]. The key idea for the presented approach is to focus directly on temporality in the data in order to learn feature representations that lead to improved activity recognition performance especially for challenging scenarios with limited labeled training data. To this end, we have introduced the concept of Contrastive Predictive Coding (CPC) into human activity recognition using body-worn movement sensors. CPC was demonstrated to be effective in, for example, audio analysis scenarios and we have adopted the technique and refined it towards the constraints and requirements of HAR. CPC learns features through a combination of encoder networks that not only target representing individual sample but rather focus on predicting samples in the temporal vicinity and as such explicitly aiming at modeling temporality. We have hypothesized that this temporality is crucial not only at modeling level but especially at representation level. The pre-trained models are then integrated into the activity recognition chain, serving as effective feature extractors.

In our extensive experimental evaluation we have demonstrated the effectiveness of the proposed approach for improved human activity recognition under realistic, challenging requirements. On a range of benchmark scenarios we have shown that CPC-learned features lead to recognition models that outperform all previous approaches to unsupervised representation learning. Furthermore, we have demonstrated that the CPC-based models are on par with supervised learning approaches. Yet, such end-to-end methods are not suitable for real-world application scenarios with limited annotated training sets. We have shown how CPC-based models can overcome these issues by demonstrating that recognition results are significantly better for small, labeled sampled sets when compared to the state-of-the-art in end-to-end learning.

These results are encouraging because they indicate that it is possible to utilize unlabeled data for deriving effective sensor data representations that in turn will lead to more effective recognition systems. Collecting even vast amounts of unlabelled sensor data can be considered straightforward given the ubiquity of mobile sensing platforms such as smartphones or other body-worn sensors. The work presented here adds to the general agenda of deriving robust and effective recognition systems for challenging assessment scenarios as they are common in applications of mobile and ubiquitous computing.

REFERENCES

[1] Gregory D Abowd. 2012. What next, Ubicomp? Celebrating an intellectual disappearing act. *Proc. Int. Conf. on Ubiquitous Computing (UbiComp)* (2012).

[2] Oliver Amft, Holger Junker, and Gerhard Troster. 2005. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*. IEEE, 160–163.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. 173–182.

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones.. In *Esann*.

[5] Bishnu S Atal and Manfred R Schroeder. 1970. Adaptive predictive coding of speech signals. *Bell System Technical Journal* 49, 8 (1970), 1973–1986.

[6] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477* (2020).

[7] Abdelkareem Bedri, Richard Li, Malcolm Haynes, Raj Prateek Kosaraju, Ishaan Grover, Temiloluwa Prioleau, Min Yan Beh, Mayank Goel, Thad Starner, and Gregory Abowd. 2017. EarBit: using wearable sensors to detect eating episodes in unconstrained environments. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1, 3 (2017), 1–20.

[8] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 1–33.

[9] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[10] Charikleia Chatzaki, Matthew Pediaditis, George Vavoulas, and Manolis Tsiknakis. 2016. Human daily activity and fall recognition using a smartphone's acceleration sensor. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*. Springer, 100–118.

[11] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. R. Millán, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* (2020).

[13] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020).

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[15] Yu-An Chung and James Glass. 2020. Generative pre-training for speech with autoregressive predictive coding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3497–3501.

[16] Yu-An Chung and James Glass. 2020. Improved speech representations with multi-target autoregressive predictive coding. *arXiv preprint arXiv:2004.05274* (2020).

[17] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240* (2019).

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*. 1422–1430.

[20] Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christoper G Owen, et al. 2017. Large scale population assessment of physical activity using wrist worn accelerometers: The UK Biobank Study. *PloS one* 12, 2 (2017), e0169649.

[21] Peter Elias. 1955. Predictive coding–I. *IRE Transactions on Information Theory* 1, 1 (1955), 16–24.

[22] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. 2017. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3636–3645.
[23] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. 2010. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing* 14, 7 (2010), 645–662.
[24] James M Fisher, Nils Y Hammerla, Thomas Ploetz, Peter Andras, Lynn Rochester, and Richard W Walker. 2016. Unsupervised home monitoring of Parkinson's disease motor symptoms using body-worn accelerometers. *Parkinsonism & related disorders* 33 (2016), 44–50.
[25] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).
[26] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.
[27] Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–28.
[28] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 297–304.
[29] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016).
[30] Nils Y Hammerla, Reuben Kirkham, Peter Andras, and Thomas Ploetz. 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 international symposium on wearable computers*. 65–68.
[31] Harish Haresamudram, David V Anderson, and Thomas Plötz. 2019. On the role of features in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*. 78–88.
[32] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. 2020. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 International Symposium on Wearable Computers*. 45–49.
[33] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.
[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[35] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. 2019. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272* (2019).
[36] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[37] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.
[38] Tâm Huynh and Bernt Schiele. 2005. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. 159–163.
[39] Aapo Hyvarinen and Hiroshi Morioka. 2016. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*. 3765–3773.
[40] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
[41] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.
[43] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. 2018. Adding structural characteristics to distribution-based accelerometer representations for activity recognition using wearables. In *Proceedings of the 2018 ACM international symposium on wearable computers*. 72–75.
[44] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *arXiv preprint arXiv:2006.05675* (2020).
[45] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
[46] Hong Li, Gregory D Abowd, and Thomas Plötz. 2018. On specialized window lengths and detector based human activity recognition. In *Proceedings of the 2018 ACM international symposium on wearable computers*. 68–71.
[47] Andy T Liu, Shang-Wen Li, and Hung-yi Lee. 2020. Tera: Self-supervised learning of transformer encoder representation for speech. *arXiv preprint arXiv:2007.06028* (2020).
[48] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal*

Processing (ICASSP). IEEE, 6419–6423.
[49] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893* (2018).
[50] Saif Mahmud, M Tonmoy, Kishor Kumar Bhaumik, AKM Rahman, M Ashraful Amin, Mohammad Shoyaib, Muhammad Asif Hossain Khan, and Amin Ahsan Ali. 2020. Human Activity Recognition from Wearable Sensor Data Using Self-Attention. *arXiv preprint arXiv:2003.09018* (2020).
[51] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. 2018. Protecting sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*. 1–6.
[52] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[53] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*. Springer, 527–544.
[54] Mehrab Bin Morshed, Koustuv Saha, Richard Li, Sidney K D'Mello, Munmun De Choudhury, Gregory D Abowd, and Thomas Plötz. 2019. Prediction of mood instability with passive sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–21.
[55] Vishvak S Murahari and Thomas Plötz. 2018. On attention models for human activity recognition. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. 100–103.
[56] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.
[57] John G Nutt, Bastiaan R Bloem, Nir Giladi, Mark Hallett, Fay B Horak, and Alice Nieuwboer. 2011. Freezing of gait: moving forward on a mysterious clinical phenomenon. *The Lancet Neurology* 10, 8 (2011), 734–744.
[58] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[59] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.
[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*. 8026–8037.
[61] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.
[62] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[63] Thomas Plötz, Nils Y Hammerla, and Patrick L Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In *Twenty-second international joint conference on artificial intelligence*.
[64] David MW Powers. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061* (2020).
[65] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
[66] A. Reiss and D. Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring.
[67] Gabriel Reyes, Dingtian Zhang, Sarthak Ghosh, Pratik Shah, Jason Wu, Aman Parnami, Bailey Bercik, Thad Starner, Gregory D Abowd, and W Keith Edwards. 2016. Whoosh: non-voice acoustics for low-cost, hands-free, and rapid input on smartwatches. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 120–127.
[68] Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7414–7418.
[69] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.
[70] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
[71] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. 2008. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 2 (2008), 42–50.
[72] Shingo Takeda, Tsuyoshi Okita, Paula Lago, and Sozo Inoue. 2018. A multi-sensor setting activity recognition simulation tool. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 1444–1448.
[73] Edison Thomaz, Irfan Essa, and Gregory D. Abowd. 2015. A Practical Approach for Recognizing Eating Moments with Wrist-Mounted Inertial Sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan) *(UbiComp '15)*. Association for Computing Machinery, New York, NY, USA, 1029–1040. https://doi.org/10.1145/2750858.2807545

[74] Alireza Abedin Varamin, Ehsan Abbasnejad, Qinfeng Shi, Damith C Ranasinghe, and Hamid Rezatofighi. 2018. Deep auto-set: A deep auto-encoder-set network for activity recognition using wearables. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 246–253.
[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[76] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 3–14.
[77] Weiran Wang, Qingming Tang, and Karen Livescu. 2020. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6889–6893.
[78] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. 2018. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8052–8060.
[79] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition.. In *Ijcai*, Vol. 15. Citeseer, 3995–4001.
[80] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu. 2018. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. 56–63.
[81] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 197–205.
[82] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T Inan, Thad E Starner, and Gregory D Abowd. 2016. Tapskin: Recognizing on-skin input for smartwatches. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. 13–22.
[83] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E Starner, Omer T Inan, and Gregory D Abowd. 2017. Fingersound: Recognizing unistroke thumb gestures using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–19.
[84] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. 2017. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–25.
[85] M. Zhang and A. Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors.
[86] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European conference on computer vision*. Springer, 649–666.
[87] Shibo Zhang, Yuqi Zhao, Dzung Tri Nguyen, Runsheng Xu, Sougata Sen, Josiah Hester, and Nabil Alshurafa. 2020. NeckSense: A Multi-Sensor Necklace for Detecting Eating Activities in Free-Living Conditions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–26.
[88] Yilun Zhao, Xinda Wu, Yuqing Ye, Jia Guo, and Kejun Zhang. 2020. MusiCoder: A Universal Music-Acoustic Encoder Based on Transformers. *arXiv preprint arXiv:2008.00781* (2020).