Applied Intelligence
https://doi.org/10.1007/s10489-024-05322-3

SelfPAB: large-scale pre-training on accelerometer data for human activity recognition

Aleksej Logacjov[1] · Sverre Herland[1] · Astrid Ustad[2] · Kerstin Bach[1]

Accepted: 6 February 2024

Abstract
Annotating accelerometer-based physical activity data remains a challenging task, limiting the creation of robust supervised machine learning models due to the scarcity of large, labeled, free-living human activity recognition (HAR) datasets. Researchers are exploring self-supervised learning (SSL) as an alternative to relying solely on labeled data approaches. However, there has been limited exploration of the impact of large-scale, unlabeled datasets for SSL pre-training on downstream HAR performance, particularly utilizing more than one accelerometer. To address this gap, a transformer encoder network is pre-trained on various amounts of unlabeled, dual-accelerometer data from the HUNT4 dataset: 10, 100, 1k, 10k, and 100k hours. The objective is to reconstruct masked segments of signal spectrograms. This pre-trained model, termed SelfPAB, serves as a feature extractor for downstream supervised HAR training across five datasets (HARTH, HAR70+, PAMAP2, Opportunity, and RealWorld). SelfPAB outperforms purely supervised baselines and other SSL methods, demonstrating notable enhancements, especially for activities with limited training data. Results show that more pre-training data improves downstream HAR performance, with the 100k-hour model exhibiting the highest performance. It surpasses purely supervised baselines by absolute F1-score improvements of 7.1% (HARTH), 14% (HAR70+), and an average of 11.26% across the PAMAP2, Opportunity, and RealWorld datasets. Compared to related SSL methods, SelfPAB displays absolute F1-score enhancements of 10.4% (HARTH), 18.8% (HAR70+), and 16% (average across PAMAP2, Opportunity, RealWorld).

Keywords Accelerometer · Human activity recognition · Machine learning · Physical activity behavior · Self-supervised learning · Transformer

1 Introduction

Objective measurement-based human activity recognition (HAR) is a research field focusing on predicting human postures and physical activities from sensor data [43]. In contrast to subjective data, based on questionnaires, objective measurements are less susceptible to bias and misclassification. Accelerometers are among the most commonly used sensors to record human movement for HAR [8]. The main reasons are their small form factor, the low cost, and their ability to provide accurate measurements needed for recognizing human physical activity [8, 11]. Nowadays, supervised machine learning (ML) is one of the most successful techniques to facilitate objective measurement-based HAR due to its ability to learn complex patterns in the data [37]. At the same time, deep learning (DL), a sub-category of ML, excels in many research fields like computer vision [48], natural language processing [2], and speech recognition [12], among others. One of the main reasons for this success is that DL approaches can extract rich features from raw data like images or sensor signals. However, DL methods tend to rely on large, primarily labeled datasets to achieve good results [26, 39]. For accelerometer-based HAR, this means anno-

✉ Aleksej Logacjov
aleksej.logacjov@ntnu.no

Sverre Herland
sverre.herland@ntnu.no

Astrid Ustad
astrid.ustad@ntnu.no

Kerstin Bach
kerstin.bach@ntnu.no

[1] Department of Computer Science, Norwegian University of Science and Technology, Sem Sælands vei, Trondheim, Trøndelag 7034, Norway

[2] Department of Neuromedicine and Movement Science, Norwegian University of Science and Technology, Edvard Griegs gate, Trondheim, Trøndelag 7030, Norway

Published online: 28 March 2024

tated acceleration recordings, where the performed activity (ground truth or gold standard) is known at each point in time. Unfortunately, creating high-quality activity annotations is a tedious task that is costly and time-consuming, especially if working with humans in a free-living setting. Free-living recordings are important since ML approaches trained on data recorded in a controlled laboratory setting perform poorly on data recorded outside a laboratory [30]. This demanding labeling process is one of the main reasons annotated accelerometer-based HAR datasets are relatively small. If they are large, they tend to be of low quality regarding the annotations [27]. As a result, it is more common to use less data-intensive ML approaches for HAR instead of DL [8].

Unlabeled physical activity data, i.e., raw acceleration signals without activity annotations, on the other hand, can be collected more cost-efficiently. The ubiquitous nature of sensors, e.g., integrated into smartwatches or smartphones, allows for collecting large amounts of unlabeled data [15]. Unlabeled accelerometer data was also recorded in the fourth iteration of the Trøndelag Health Study (HUNT4) [32]. The data corpus contains acceleration data of approximately 35,000 subjects, each recorded for up to seven days, resulting in millions of hours of physical activity data. Two body-worn, three-axial accelerometers attached to the lower back and thigh were used. This dataset contains a large amount of information about human physical activity behavior, which cannot be used to train purely supervised ML models.

More recently, self-supervised learning (SSL) gained much attention in the ML community due to its ability to extract useful representations from unlabeled data [18]. In general, SSL consists of two steps. First, a model is pre-trained on unlabeled data by defining an objective (auxiliary task) the model has to solve (upstream training). Second, the learned representations of the upstream training are leveraged to solve tasks that rely on annotated data (downstream training), like HAR [29]. The main goal is to improve the downstream performance, compared to purely supervised learning (i.e., no upstream training), based on the representations learned through self-supervised pre-training. SSL achieved state-of-the-art performances in many research fields. Examples are computer vision [48], natural language processing [9], and speech representation learning [24]. But also, the HAR research community started to investigate different SSL approaches (see Section 2). However, most works in SSL-based HAR use small labeled datasets for both upstream and downstream training. It has been shown in other research fields, like natural language processing, that the amount of training data plays a crucial role in a neural network's performance, with more data leading to better results [20]. A similar observation was made for single-accelerometer-based HAR [49]. However, none of the existing SSL-based HAR literature investigates large-scale, dual-accelerometer datasets for pre-training and their influence on HAR performance. We fill this gap by making the following contributions:

1) We implement a self-supervised physical activity behavior representation learning method (SelfPAB). It is based on the speech representation learning approach TERA [24]. We pre-train a transformer encoder network [45] on the unlabeled HUNT4 data corpus. The auxiliary task during pre-training is to reconstruct masked time windows and frequency bands in six spectrograms, each referring to one axis of the two three-axial accelerometers used in HUNT4. The pre-trained transformer encoder network is used as a feature extractor during downstream HAR training. For the downstream part, we use the five labeled HAR datasets, HARTH [27], HAR70+ [44], PAMAP2 [34], Opportunity [4], and RealWorld [38].

2) We experiment with different amounts of unlabeled data for pre-training. In particular, 10 hours, 100 hours, 1,000 hours (1k hours), 10,000 hours (10k hours), and 100,000 hours (100k hours) of the HUNT4 dataset. We show that only 10 hours of acceleration signals, less than many supervised datasets contain, are sufficient to achieve similar (on HARTH) and higher (on HAR70+) performances than purely-supervised methods. Using 100 hours shows better results in both datasets. Similar to related work on single accelerometers [49], our experiments indicate that increasing the number of hours leads to further performance improvements. Hence, the amount of hours used for pre-training seems to scale with the downstream performance. The best model is pre-trained on 100k hours.

3) Besides HARTH, there is currently no labeled HAR dataset with the same sensor setup as HUNT4 publicly available. Therefore, we create and publish the HAR70+ dataset together with this paper. In contrast to HARTH, HAR70+ contains only subjects over 70 years old, showing potentially different movement patterns for the same activities. We further publish our experiments, and pre-trained model[1].

This paper is the full version of our previous work [28]. Building upon the foundations, we provide a more extensive analysis, detailed experiments, and offer a better understanding of SelfPAB and its applications.

This paper is organized as follows. Section 2 gives an overview of related self-supervised learning approaches in HAR. Section 3 describes the proposed methodology, including upstream and downstream training. Our experimental setup is described in Section 4. The corresponding results are shown in Section 5 and discussed in Section 6. Finally, the conclusion is given in Section 7.

2 Related work in self-supervised learning for human activity recognition

Self-supervised learning (SSL) gained much attention in recent years due to its great success in other research fields

[1] https://github.com/ntnu-ai-lab/SelfPAB (accessed on 2024-02-13)

[9, 24, 48]. One of the first attempts to apply SSL to improve accelerometer-based HAR was made by Saeed et al. [35]. Since then, different works have investigated different SSL strategies. These strategies can be grouped into three categories:

1) **Multi-task self-supervision**: In this category, multiple auxiliary tasks are defined at once. Related works focused on transformation-based multi-task self-supervision, hence, identifying what kind of transformation(s), if any, is applied to the input signal. Saeed et al. [35] applied one of eight different transformations to the time signals, and the overall objective was to identify which of the eight was performed. Tang et al. [40] proposed a combination of transformation recognition (as in [35]) with a knowledge distillation paradigm. Hence, the tasks were to identify one of eight transformations applied to the signals and reproduce the predictions of a pre-trained teacher model. Yuan et al. [49] used three transformations of Saeed et al. [35] to pre-train a ResNet-V2 with 18 layers [16] on the large-scale UK-Biobank single-accelerometer recordings [10].

2) **Contrastive learning**: In contrastive learning, input representations are learned through comparing input samples [22]. The representations of "similar" samples (positive samples) need to be closer together than the representations of "dissimilar" samples (negative samples)[22]. How "similar" / "dissimilar" and the distance are defined depends on the used algorithm. In [14], the objective was to predict k future time steps starting from t inside a given time frame W_i while the positive sample was the interval t to $t+k$ in W_i and the negative samples t to $t+k$ in other windows $W_{j\neq i}$. Tonekaboni et al. [42] defined a neighborhood function on the sensor signals. Positive samples were part of the neighborhood of an input sample (anchor), and negative samples were not part of the neighborhood. Liu and Abdelzaher [25] utilized both labeled and unlabeled data during pre-training, while negative and positive samples were generated depending on the prediction of the unlabeled data. Saeed et al. [36] defined positive samples as the scalograms of the anchor time signals created using wavelet transform and negative samples as scalograms of other time signals. The pre-training objective of Khaertdinov et al. [21] was to increase the cosine similarity of two augmented versions of the same signal frame and decrease it for augmented versions of different signal frames. Wang et al. [47] also used augmentation strategies to create negative and positive samples. They tested different augmentation approaches and presented a new augmentation strategy simulating changing sampling frequencies. Jain et al. [17] considered time-aligned signal frames of different sensors/devices as natural transformations of each other. Hence, positive (time-aligned) and negative (not time-aligned) samples are generated from various sensors. Wang et al. [46] used a combination of augmentation and clustering to define negative and positive samples.

3) **Masked reconstruction**: In masked reconstruction-based SSL, parts of the input signals are masked out (e.g., replaced with zeros), and the pre-training objective is the reconstruction of these parts to learn local temporal dependencies [15]. Haresamudram et al. [13] first masked out random 10% time-domain samples from the raw accelerometer signals and second trained an upstream architecture to reconstruct these samples. The mean squared error loss on the masked proportions was used to compare the model output with the original signal. A transformer encoder architecture with following fully-connected layers was trained during pre-training. During downstream training, the former was used as a feature extractor for a multilayer perceptron. Masked reconstruction was also applied by Taghanaki et al. [39]. Given a 2.08 sec long time frame of a three-axial accelerometer measurement, the last 0.48 sec of the z-axis were masked. The task of the upstream model (a convolutional neural network with subsequent feed-forward layers) was to reconstruct the masked part of the z-axis. The pre-trained convolutional neural network was used for feature extraction during downstream training. The downstream architecture was a four-layer multilayer perceptron.

The mentioned related works for masked reconstruction-based SSL show some limitations. First, they consider only a single sensor even though many studies show that using more than one can increase the HAR performance [7, 31]. Second, the former two works ([13, 39]) use the labeled HAR datasets for both pre-training and downstream training. Due to the datasets' limited sizes, this aspect makes it difficult to investigate whether more pre-training data can improve the HAR results. Haresamudram et al. [15] and Yuan et al. [49] are the only two works studying large-scale datasets for pre-training HAR models. Haresamudram et al. [15] studied different pre-training data quantities with the unlabeled Capture-24 dataset. However, they focused on only one sensor, and Capture-24 has its limitations of 4000 hours and 151 participants. The authors showed that $\approx$ 222 hours out of maximal $\approx$ 4000 hours achieved the best results. Hence, more pre-training data did not necessarily improve the HAR downstream performance. In our work, we want to investigate whether this statement holds for even more hours of data. With the UK-Biobank, Yuan et al. [49] used an even larger dataset for pre-training than Haresamudram et al. [15]. It contains over 700,000 person-days of wrist-accelerometer data, with around 100,000 subjects. The authors investigated the effect of the amount of UK-Biobank pre-training data on the downstream performance in more detail. They observed that the downstream results scale with the number of subjects and not necessarily with the number of samples per subject. However, only a single accelerometer was investigated. It remains an open question whether the same behavior can be achieved with a dual-accelerometer setup and masked reconstruction instead of multi-task SSL. We show that increasing

the amount of pre-training data to up to 100,000 hours can lead to better downstream performance using our architecture and auxiliary task, as well as a dual-accelerometer setup, although just marginally. This matches with empirical studies about the influence of training dataset size on transformer-based language model performance [20] and the findings of Yuan et al. [49].

3 Methods

The self-supervised physical activity behavior representation learning method (SelfPAB) used in this work is illustrated in Fig. 1. It is based on TERA [24], a speech representation learning technique, for two main reasons: First, TERA is successfully applied to automatic speech recognition, but more importantly, it performs well in classification tasks like phoneme classification. Hence, it showed good performance on time series classification. HAR based on dual accelerometers is a multivariate time series classification problem. Second, since TERA uses masked reconstruction instead of contrastive learning, it does not suffer from the so-called sampling bias [6]. SelfPAB consists of two parts, an upstream (left) and a downstream part (right). First, an upstream network is pre-trained on unlabeled data to acquire potentially useful physical activity representations. Second, the resulting model is utilized as a feature extractor for downstream training (e.g., HAR). The goal is to improve the performance of a downstream model by leveraging the representations the upstream model acquires from the unlabeled data.

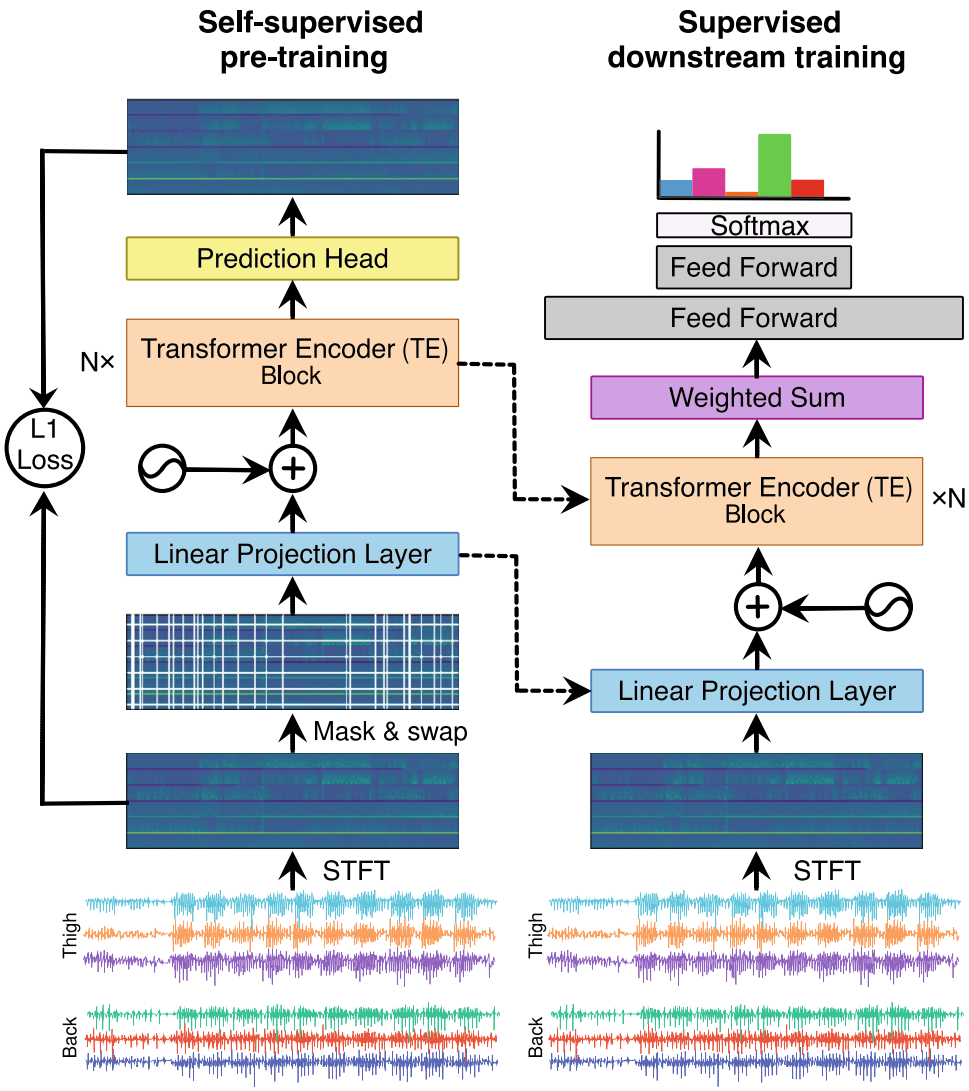
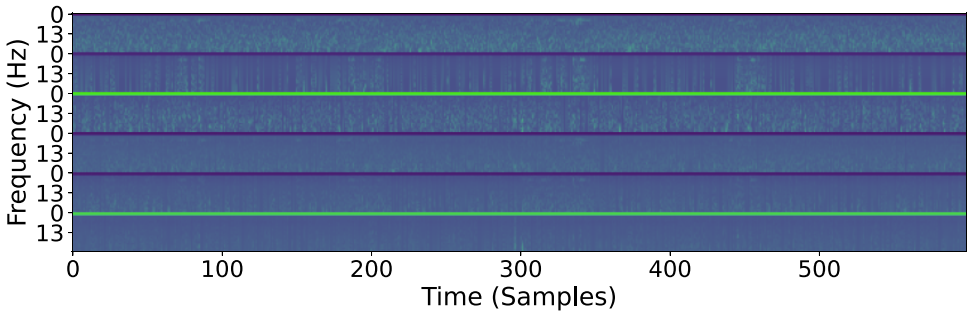
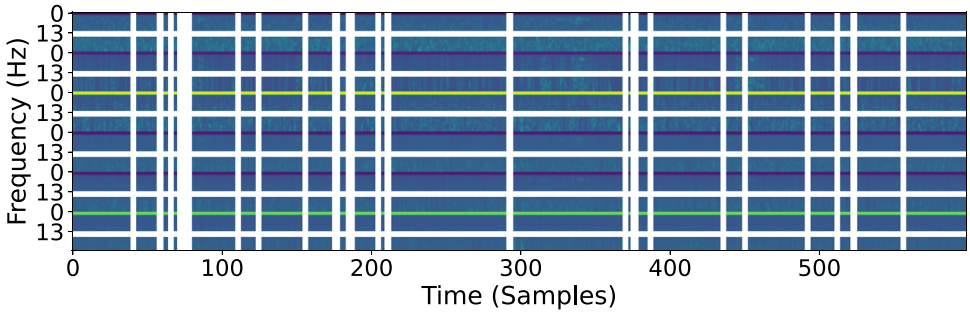


Fig. 1 Illustration of the SelfPAB method, consisting of two parts, the self-supervised pre-training (left) and the supervised downstream training (right)



(a) Original spectrograms



(b) Time and frequency masking

Fig. 2 Stacked five-minute long (a) spectrograms of each sensor axis, created with the short-time Fourier transform. Masking with zeros is performed (b) in the time domain (vertical white lines) and the frequency domain (horizontal white lines). The frequency bands (y-axis) are defined from $0Hz$ to $25Hz$, and the time stamps (x-axis) are from 0 to 599 samples

3.1 Upstream

3.1.1 Acceleration signals

We use two Axivity AX3 (Axivity Ltd., Newcastle, UK) [2] accelerometers attached to the participants' lower back and thigh. Each sensor records the acceleration in three spatial dimensions, resulting in six time signals. We compute spectrograms of each signal using the short-time Fourier transform (STFT) to get the frequency content over time. This is inspired by the research field of automatic speech recognition (ASR), where spectrograms are successfully utilized for pre-training instead of raw time signals [23, 24]. Additionally, we stack the six resulting spectrograms on top of each other. In all our experiments, we work with a sampling rate of $50Hz$, resulting in a maximum frequency of $25Hz$ in the spectrograms due to the Nyquist-Shannon sampling theorem.

3.1.2 Signal alteration and auxiliary task

Defining a proper auxiliary task for the pre-training is crucial in self-supervised learning [24]. It determines what information the upstream model learns from the unlabeled data and whether it is helpful for downstream training. We decide to utilize the masked reconstruction auxiliary task as it allows learning of temporal dependencies without much effort [15], which leads to useful representations of accelerometer signals for downstream tasks. Another benefit of masked reconstruction is that, compared to contrastive approaches, it does not suffer from the sampling bias [6]. Masking and reconstructing parts of the input data has already been successful in self-supervised learning for HAR [13, 39] and other research fields, like natural language processing (NLP) [9] and ASR [24]. The primary strategy is to mask certain parts of the input and let the model learn to reconstruct these parts using the unmasked parts. As a result, we perform two alteration techniques on the input spectrograms.

1) Time domain alteration: As in [24], we define a time alteration percentage P_T, which determines the maximal amount of time frames to be altered in all six spectrograms. First, a number $T_{num} = \lfloor \frac{P_T \cdot L_T}{W_T} \rfloor$ of start indices are randomly chosen without replacement. L_T is the total number of input time frames, and W_T is a predefined window width to be altered. With a probability of 80%, the selected frames are replaced with zeros, with a probability of 10%, they are swapped with other frames in the input, and with a probability of 10%, they are not altered at all. Liu et al. [24] argue that the latter case tackles the train-test inconsistency problem by feeding a not-altered input into the model. The white vertical lines in Fig. 2b illustrate the masking of time frames with zeros in all six spectrograms. Note that altered windows can overlap, leading to larger consecutive masked/swapped areas.

[2] http://www.axivity.com/ (accessed on 2023-12-19)

2) Frequency domain masking: Like in time domain masking, we compute a number of start indices $F_{num} = \lfloor \frac{P_F \cdot L_F}{W_F} \rfloor$ using frequency masking percentage P_F, frequency window width W_F, and the number of frequency bins of one sensor L_F. The same consecutive frequency bands are masked (zeroed out) in all six spectrograms. The white horizontal lines in Fig. 2b illustrate this masking. The idea is to let the model learn frequency band reconstruction using the other frequency bands.

Note that time domain alteration and frequency domain masking are combined in most cases. Hence, our upstream model receives masked spectrograms as input, and its objective is to reconstruct the masked areas, shown in the top left of Fig. 1. Like in the work of Liu et al. [24], we use the L1 reconstruction loss $l_1 = M \cdot |y - \hat{y}|$ between the upstream model's output $\hat{y}$ and the unmasked spectrograms y. M is a matrix with the same dimension as y and $\hat{y}$ and contains ones where alteration (masking or swapping) is applied and zeros anywhere else. The multiplication with M ensures that the L1 loss is computed for the altered parts only.

3.1.3 Upstream architecture

The masked spectrograms are forwarded to a linear input projection layer. It is a single trainable feed-forward layer, mapping the input to a predefined embedding of dimension d_{model}. Sinusoidal positional encoding is used to preserve information about the order of the input sequence. Input embedding and positional encoding are summed together and forwarded to a transformer encoder network consisting of N transformer encoder layers. Transformer models, proposed by Vaswani et al. [45], can learn relationships between a set of input vectors without the usage of recurrent or convolutional layers, making them efficient to train. A transformer encoder layer consists of a multi-head self-attention block and a feed-forward layer. Residual connections with layer normalizations are applied on each of the two sub-layers. The multi-head self-attention block uses scaled dot-product attention to learn the temporal relations between given input samples, i.e., in our case, between time windows in the stacked spectrograms. Stacking multiple transformer encoder layers results in a transformer encoder (network). For more details about the transformer architecture, we refer to the original paper of Vaswani et al. [45].

The output of the last transformer encoder layer is forwarded to the prediction head, a feed-forward layer mapping the d_{model}-dimensional vectors back to the input dimension d_{input} to make the model's output comparable to the unmasked spectrogram. The prediction head is not used during downstream learning. Despite the similarity of SelfPAB to TERA of Liu et al. [24], we want to highlight the differences here. 1) We work with standard spectrograms in contrast to log Mel spectrograms, used by Liu et al. [24]. 2) We consider a six-dimensional time series (two sensors, each having three axes) instead of a univariate time series as in [24]. 3) Liu et al. [24] applied magnitude alteration by adding noise to the spectrograms, which we do not. The reason is that it did not provide a strong benefit in classification tasks [24].

3.2 Downstream

We use the pre-trained linear input projection layer and the transformer encoder upstream network to extract features from the input spectrograms. The goal is to improve the HAR downstream performance using these features instead of raw spectrograms as input to the downstream model. Like Liu et al. [24], we use the weighted sum $\mathbf{F} = \sum_{l=1}^{L} \mathbf{F}_l \cdot w_l$ of each transformer encoder layer's output F_l as input to the downstream network. L is the number of transformer encoder layers and w_l a trainable weight scalar. This technique allows the model to learn which layer in the upstream network is most important for the downstream training. It is inspired by Chi et al. [5], who showed that using internal transformer encoder layers for feature extraction can lead to better speaker recognition and phoneme classification results.

The downstream architecture is a multilayer perceptron (MLP) with one hidden layer. It receives the upstream model's d_{model}-dimensional output as input. The output layer has the same dimension as the number of classes/activities in the HAR downstream training. A ReLU activation is applied to the hidden layer's output and a softmax activation function to the output layer to model a categorical distribution over all activities. Initially, the weights of the upstream model are frozen, and only the weighted-sum layer and downstream MLP are trained to prevent the initially large gradients from altering the carefully set parameters of the upstream model too much. After initial training of the classifier, we perform fine-tuning. Hence, we unfreeze the upstream model's weights after a predefined number of downstream steps. Fine-tuning upstream models for downstream training showed promising results in related works [5, 24] and better performance in our initial experiments.

4 Experiments

We test our approach in experiments with six different datasets, the HUNT4 [32], the HARTH v1.2 [27], the HAR70+, the PAMAP2 [34], the Opportunity [4], and the RealWorld [38]. HUNT4 is an unlabeled dataset, and it is utilized for pre-training only. After pre-training, we investigate the HAR performance on the latter five datasets, which are all labeled. The pre-training part is also known as upstream

training, and the following supervised training on the five labeled datasets is also called downstream training.

4.1 Pre-training / Upstream

4.1.1 HUNT4 dataset (unlabeled)

We use the unlabeled HUNT4 [32] dataset for pre-training our upstream model. HUNT4 is the fourth round of Norway's biggest health study, the Trøndelag Health Study. Accelerometer data of approximately 35,000 participants were recorded. Each participant wore two three-axial Axivity AX3 accelerometers for up to seven days. The sensors were attached to the participants' lower back and thigh, and recordings were made with a sampling rate of $50Hz$. Up to the time of writing this paper, the HUNT4 is, to the best of our knowledge, the largest dual-accelerometer-based physical activity data corpus in the world. HUNT4 consists of around 230 times more subjects with significantly more hours of data than the Capture-24 dataset [3]. Hence, it is a good candidate to investigate a large variety of hours used for pre-training.

4.1.2 Data pre-processing

Five-minute time windows ($15,000$ samples at $50Hz$), a frame length of $1sec$ ($= 50$ samples), an overlap of half a second ($= 25$ samples), and the Hann window function are used for STFT computation. This results in 26 frequency bins and 599 time frames for each axis. The six sensor spectrograms are stacked, resulting in 156×599-dimensional input matrices. We use the upstream dataset's mean and variance to normalize the input before pre-training.

4.1.3 Parameters

Table 1 shows the hyperparameter assignments used during pre-training. After initial experiments with different hyperparameter assignments, these achieved the best results. The linear projection layer transforms the input of dimension $d_{input} = 156$ to $d_{model} = 1500$. The transformer encoder network consists of four transformer encoder layers each having six attention heads, and a 2048-dimensional feed-forward layer.

We use AdamW with a weight decay factor of $1e^{-5}$ as the optimizer. Like in the work of Liu et al. [24], we perform a linear learning rate warm-up in the first 7% of training steps, leading to a peak learning rate of $1e^{-4}$. Afterward, a linear learning rate decay is applied with a final learning rate of $1e^{-6}$ in the last epoch.

We experiment with a large variety of different amounts of unlabeled data for pre-training. In particular, we compare the downstream performance when using upstream models trained on 10, 100, 1k, 10k, and 100k hours of acceleration data. We refer to these models as SelfPAB 10, SelfPAB 100, SelfPAB 1k, SelfPAB 10k, and SelfPAB 100k, respectively. To ensure comparability, we train the SelfPAB 10 model for 500,000, the SelfPAB 1k model for 50,000, the SelfPAB 1k model for 5,000, the SelfPAB 10k for 500, and the SelfPAB

Table 1 Hyperparameter assignments of the pre-training

Type	Hyperparameter	Value
Architectural	Number of TE layers	4
	Embedding dim. (d_{model})	1500
	Number of heads	6
	Feed-forward dimension	2048
Training	Peak learning rate	$1e^{-4}$
	End learning rate	$1e^{-6}$
	lr schedule	warm-up + linear decay
	Optimizer	AdamW
	Weight decay factor	$1e^{-5}$
	Epochs	500k, 50k, 5k, 500, 50
	Batch size	64
	Loss	(Masked) L1
Masking	Time alter. perc. (P_T)	0.15
	Time alter. width (W_T)	3
	Freq. masking perc. (P_F)	0.2
	Freq. masking width (W_F)	3

It is divided into architectural, training, and masking hyperparameters. The terms dim, TE, lr, Freq., alter., and perc. are the abbreviations for dimension, transformer encoder, learning rate, Frequency, alteration, and percentage, respectively

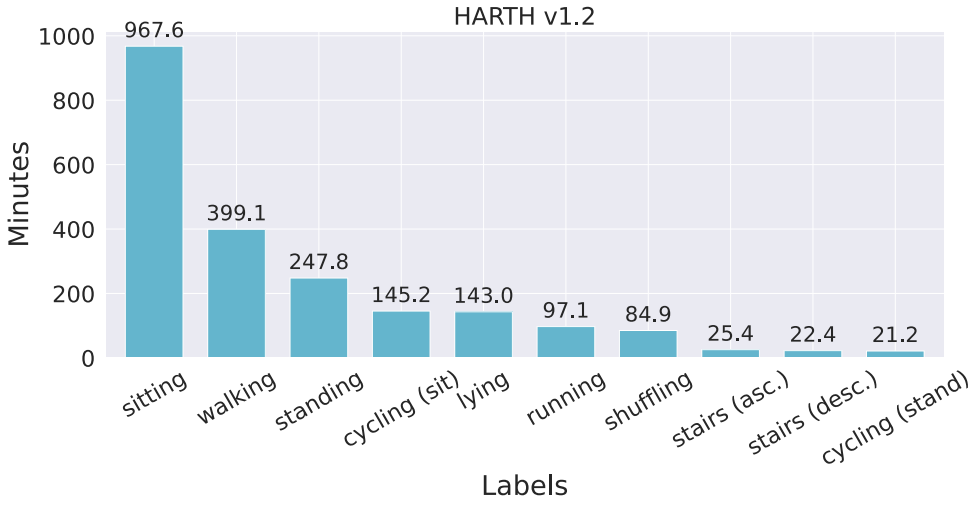


Fig. 3 HARTH v1.2 activity distribution in minutes per activity. The inactive cycling labels are combined with the active cycling labels. The illustration is based on [27]

100k for 50 epochs, all with a batch size of 64. Hence, all five models receive the same number of samples and take the same number of gradient steps. The only difference is the number of unique samples. We randomly select five-minute time windows of the HUNT4 data corpus to collect the required amount of data. A seed of 256 is used to ensure comparability between experiments. This results in 108 subjects for 10 hours, 1062 subjects for 100 hours, 9277 for 1000 hours, 33932 for 10k hours, and 35650 for 100k hours. Ten percent of the data is used as a validation set. Hence, SelfPAB 10 is actually trained on 9 hours and validated on 1 hour. SelfPAB 100 is trained on 90 hours and validated on 10 hours, SelfPAB 1k is trained on 900 hours and validated on 100 hours, and so on. We use checkpoints during training such that the model iteration with the lowest validation loss is always saved to disk.

For creating the altered time frames, we define a time alteration percentage of $P_T = 0.15$, meaning that up to 15% of the time frames can be altered. The amount of consecutive time frames to alter is set to $W_T = 3$. We set the frequency masking percentage to $P_F = 0.2$ and the frequency masking width to $W_F = 3$. Hence, we create a $3Hz$ mask, which is replicated across all six sensor axes' spectrograms. This strategy avoids masking too much information out of the frequency domain. These masking hyperparameter assignments are mainly inspired by related works in self-supervised learning with masked reconstruction [9, 23, 24].

4.2 Downstream

4.2.1 Datasets (labeled)

This work considers five publicly available and labeled datasets for downstream training (i.e., HAR). The first two (HARTH v1.2 and HAR70+) are, to the best of our knowledge, the only two labeled and publicly available HAR datasets with the same sensor setup as HUNT4. Therefore, our main focus of this work is on these two datasets. The remaining three (PAMAP2, Opportunity, and RealWorld) consist of recordings from multiple sensors and sensor placements. In our experiments, we take two accelerometers from each dataset, which are positioned closest to the thigh and lower back. We test SelfPAB on the latter three datasets, to investigate its robustness against sensor displacement.

1) HARTH v1.2: The first is the HARTH v1.2 [27][3]. Twenty-two subjects were recorded for around 1.5 to 2 hours during their regular working hours in a free-living setting. They had an average BMI of 23.1 ± 2.3 (min : 19.2, max : 28.4) $\frac{kg}{m^2}$ and were on average 38.6 ± 14 (min : 25, max : 68) years old. Hence, it shows a wide variety of ages. Professional annotations were created using video recordings. HARTH v1.2 has twelve different activities: walking, running, shuffling (i.e., standing with leg movement), stairs (ascending), stairs (descending), standing, sitting, lying, cycling (sit), cycling (stand), cycling (sit, inactive), and cycling (stand, inactive) (i.e., cycling (sit/stand) without leg movement). The dataset contains around 2221.6 min ($\approx$ 37 hours) of acceleration data. Figure 3 shows the activity distribution of the HARTH v1.2. The dataset is highly imbalanced, making HAR a challenging task for ML approaches. In the original HARTH experiments [27], as well as in the following validation study [1], specific activity labels were merged due to their similarity in the actual physical motion and to mitigate class imbalances. We perform a similar class merging to increase the comparability to the original works. Hence, for the HARTH v1.2, we combine cycling (sit, inactive) and cycling (sit) as well as cycling (stand, inactive) and cycling (stand), resulting in ten activities.

[3] Dataset available at https://github.com/ntnu-ai-lab/harth-ml-experiments/tree/v1.2/harth (accessed on 2023-12-19)

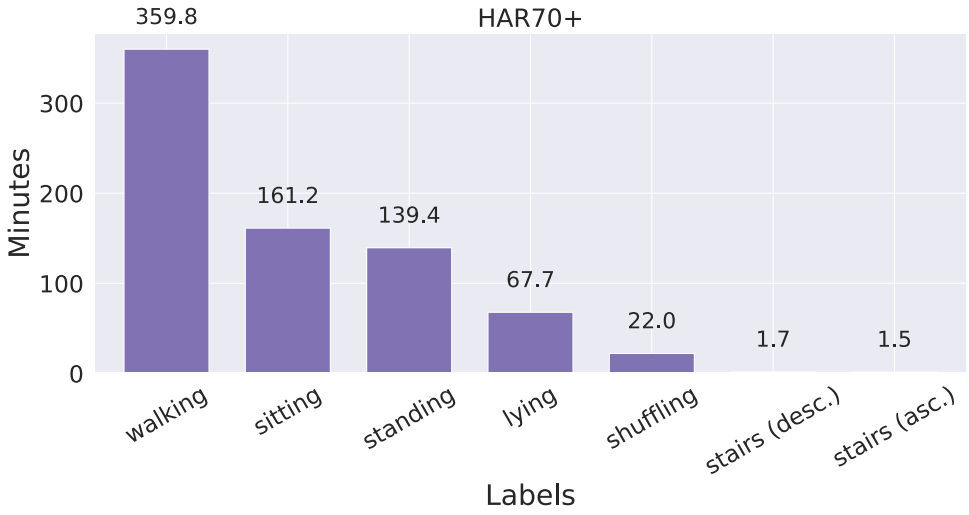


Fig. 4 HAR70+ activity distribution in minutes per activity. The illustration is based on [27]

2) HAR70+: We created the HAR70+ dataset. In contrast to HARTH v1.2, only subjects over 70 ($min : 70, max : 95$) years were recorded[4], allowing us to investigate the same activities with potentially distinct movement patterns. Eighteen subjects with a BMI of $26.8 \pm 2.7 \frac{kg}{m^2}$ participated in the data collection. Seven activities were professionally annotated by us: standing, shuffling, walking, sitting, lying, stairs (descending), and stairs (ascending). HAR70+ consists of around 756 min (= 12.6 hours) accelerometer recordings. The activity distribution is shown in Fig. 4. As in HARTH v1.2, a high imbalance is observable, with the most common activity walking and the least frequent activities stairs (descending/ascending). Each participant provided written informed consent, and we obtained ethical approval from the Norwegian Centre for Research Data (NSD).

3) PAMAP2: The PAMAP2 dataset [34] was recorded with three inertial measurement units (IMUs) attached to the chest, wrist, and ankle of nine subjects. We use only the chest and ankle accelerometer recordings in our experiments. Furthermore, we focus on the same 12 activities as Haresamudram et al. [15].

4) Opportunity: The Opportunity dataset [4] was recorded with seven IMUs and twelve accelerometers. We focus only on the back and the top right knee acceleration recordings. Four subjects were recorded, each in five runs of activities of daily living and one drill run. Five different activities/modes of locomotion are utilized: standing, sitting, walking, lying, and transition/null. We added the null class to maintain continuous signals.

5) RealWorld: In the RealWorld dataset [38], 15 subjects were equipped with seven wearable devices containing accelerometers. We use the waist and thigh recordings, as well as all eight annotated activities: downstairs, upstairs, jumping, lying, standing, sitting, running/jogging, and walking.

We resample PAMAP2 and Opportunity to $50Hz$ using the Fourier method to match the HUNT4 sampling rate. PAMAP2 is downsampled from $100Hz$ to $50Hz$ by truncating high frequencies in the spectrum. Opportunity is upsampled from $30Hz$ to $50Hz$ through zero-padding of high frequencies in the spectrum. Additionally, if applicable, sensor reorientation was performed on the latter three datasets to match the orientation of the HUNT4 dataset.

4.2.2 Downstream training

We investigate four different settings to show the benefits of our two-stage approach. The hyperparameters of the downstream experiments are found through initial hyperparameter optimizations in the form of grid searches and summarized in Table 2.

(1) SelfPAB: The weights of the pre-trained transformer encoder network and the linear projection layer are frozen and used for feature extraction, as shown on the right side of Fig. 1. The weighted sum of each transformer encoder layer's output is forwarded to a two-layer MLP, with hyperparameters shown in Table 2. Four weight scalars w_l are trained for weighted sum computation, one for each transformer encoder layer. The downstream MLP's hidden layer has a dimension of 1028, and the output layer has a dimension depending on the number of activities in the dataset used. Furthermore, we unfreeze the weights of all transformer encoder layers and the linear projection layer after 3/4 of the total training steps. Learning rate decay is performed to ensure a small enough learning rate during fine-tuning.

(2) Spectrograms + MLP: In the second setting, we skip the pre-trained model and train the mentioned MLP directly on the stacked spectrograms. This allows us to investigate whether the upstream model learns signal representations

[4] https://github.com/ntnu-ai-lab/harth-ml-experiments/tree/main/har70plus (accessed on 2023-12-19)

Table 2 Hyperparameter assignments of the downstream experiments

Type	Hyperparameter	Value
Architectural (MLP)	Dim. hidden layer	1028
	Dim. output layer	12, 7, 12, 5, or 8
Architectural (TE)	Number of TE layers (L)	4
	Embedding dim. (d_{model})	1500
	Number of heads (h)	6
	Feed-forward dimension	2048
	Dimension prediction head	12, 7, 12, 5, or 8
Training	Learning rate	$1e^{-4}$
	Learning rate schedule	exponential
	Learning rate decay factor	0.1
	Optimizer	Adam
	Steps	2000
	Batch size	32
	Loss	L1
	Start fine-tuning step	1500 (not for linear evaluation)

The table is divided into architectural and training hyperparameters. The multilayer perceptron (MLP) architecture is used in settings (1) and (2), and the transformer encoder network in setting (3). The terms dim. and TE are the abbreviations for dimension and transformer encoder, respectively

that improve HAR performance compared to raw spectrograms. The same architectural hyperparameters as in setting (1) are used (see Table 2).

(3) Spectrograms + TE: In the third setting, we also skip the upstream model and use spectrograms as inputs. However, we use the upstream model's, not pre-trained, architecture for the downstream training. Hence, instead of an MLP as in setting (2), a four-layer transformer encoder network with a linear projection layer, positional encoding, and prediction head is used. Therefore, we train the transformer encoder (TE) network end-to-end with random weight initialization. Except for the prediction head, all hyperparameters are the same as in the upstream model (see Tables 1 and 2). The prediction head has a dimension corresponding to the number of activities in the downstream dataset. A softmax activation follows the prediction head to get a distribution of the activities. Setting (3) allows us to investigate whether a potential HAR performance increase of SelfPAB is caused only by the transformer encoder architecture or the combination of architecture and pre-training.

(4) SelfPAB (linear eval.): As commonly done in SSL research, we make a linear evaluation [15, 40]. In a linear evaluation, we use our upstream model as a feature extractor for a single feed-forward layer with softmax activation instead of a two-layer MLP as in setting (1). Only the feed-forward layer is trained, and no fine-tuning is performed here.

Settings (2) and (3) will answer the question of whether the pre-training objective in combination with the proposed architecture is helpful for HAR or not. Setting (4) will answer the question of how well the activities are linearly separable in the learned latent space. Thus, it will give us an impression of how well the upstream model learned to create distinct latent representations for different activities. In all four settings, the spectrograms of all subjects are computed first. As before, the frame length is set to 50 samples ($= 1sec$), and the frame shift to 25 samples (half a second). Since we normalize the HUNT4 data during pre-training, we do the same for the downstream datasets using the upstream (HUNT4) dataset's mean and variance before downstream training. This strategy showed a considerable performance improvement in previous work [15].

The first 20% of each subject's spectrogram is used as the validation set and the remaining 80% as the training set. Splitting the data this way leads to roughly the same activity distribution of the validation and training set. We randomly cut 32 (batch size) five-minute windows (599 time bins) out of the training set in each training step. The models are trained on 2000 steps in total. We utilize the Adam optimizer, a learning rate of $1e^{-4}$, and exponential learning rate decay with a decay factor of 0.1. The categorical cross-entropy is used as the loss function. Every tenth step, we compute the loss, accuracy, recall, and precision of the validation set. For HARTH v1.2 and HAR70+, a leave-one-subject-out cross-validation (LOSO) is performed. Hence, the model is trained on $S - 1$ subjects and tested on 1 after training, with S being the number of subjects. This is repeated S times, each time with a different test subject. A LOSO has the advantage of having less subject-based bias than other cross-validation methods [31] and allows us to compare the performance on different subjects independently. We perform five-fold cross-validations for the remaining three datasets (PAMAP2, Opportunity, and RealWorld) instead to

make them comparable to related work [15]. Averaged across all activities, we compute the average F1-score, average precision, and average recall for each test subject. We further compute the overall accuracy for each test subject. Note that the activity distribution output of the downstream model is given for one second with half a second overlap. The reason is that we define the frame length of the input spectrograms to be one second and the overlap half a second. The actual prediction for one second is the maximum of the softmax output. For overlapping windows, the output distributions are averaged, and the maximum is taken afterward. To make the results comparable, we unfold the resulting one-second predictions to 50 samples per second, hence, back to the original time domain dimension.

4.2.3 Baselines and comparison to related work

We compare our method to purely supervised baselines and SSL methods. The first two purely supervised methods are the support vector machine (SVM) and extreme gradient boost (XGB). These two approaches achieved the best results in the original experiments of Logacjov et al. [27] for the HARTH dataset. We ensure comparability by retraining an XGB and an SVM on HARTH v1.2 and HAR70+, respectively, using a hyperparameter optimization followed by a LOSO. We compute the same $F = 161$ features of five-second time frames the authors in [27] used for training. We use the radial basis function, with parameter $\gamma = \frac{1}{F \cdot \sigma_X^2}$, as the kernel function of the SVM, with σ_X^2 being the variance of the training set X. A regularization parameter of $C = 10$ and no class weighting lead to the best results. For XGB, the best hyperparameters of our hyperparameter optimization are learning rates of 0.5 and 0.3, number of estimators 1024 and 512, and maximum depths of 3 and 5, for the datasets HARTH v1.2 and HAR70+, respectively. The multi-class classification error rate is used as the loss function. For the PAMAP2, Opportunity, and RealWorld, we also perform hyperparameter optimizations, followed by five-fold cross-validations. The baseline predictions are given for five seconds. Hence, similar to the downstream experiments, we unfold these five-second predictions to 50 samples per second, ensuring comparability. In addition to the XGB and SVM, we experiment with a well-established baseline method for HAR, the DeepConvLSTM [33]. Architectural hyperparameters are the same as in the original DeepConvLSTM paper to maintain comparability. The remaining hyperparameters are found through hyperparameter optimizations. We experiment with time series data as input (TS), as well as spectrograms (Spectr.).

Furthermore, we compare SelfPAB to two SSL methods, the SimCLR [41] and SelfHAR [40]. SimCLR achieved the best SSL results in the large study of Haresamudram et al. [15] about SSL in HAR, making it a good candidate to compare our method to. SelfHAR is a promising method that combines self-training and SSL in a three-stage approach not investigated by Haresamudram et al. [15]. We pre-train SimCLR and SelfHAR using 100k hours of the HUNT4 dataset and perform downstream training on the five labeled datasets afterward. For SimCLR, the same hyperparameter optimization as in [15] is performed for a fair comparison. For SelfHAR, the default hyperparameters provided in the authors' source code are utilized due their model's higher training complexity. We use the original code provided by the authors[5][6].

5 Results

5.1 HARTH v1.2 overall downstream performance

Table 3 shows the average F1-score, precision, recall, and accuracy across all test subjects of the HARTH v1.2 LOSO, together with the corresponding standard error. The first six methods are the four purely supervised baselines, as well as setting (2) and (3). The last four rows show the self-supervised methods, SelfHAR, SimCLR, SelfPAB (linear eval.), and our proposed SelfPAB pre-trained on 100k hours of HUNT4 data. The best results are written in bold letters, and the second-best are underlined. SelfPAB 100k shows the best results in all four metrics. The XGB has the second-best F1-score, recall, and accuracy, closely followed by the SVM. However, the accuracy results should be treated with caution since the activities of HARTH v1.2 are highly imbalanced, which is not considered when computing the accuracy. The F1-score takes imbalances into account and is a trade-off between precision and recall. It is therefore considered the most important metric in this work. With the highest F1-score of 81.3%, SelfPAB 100k is the best model in our experiments. The DeepConvLSTM (TS) has the worst performance with an F1-score of 51.2%, followed by DeepConvLSTM (Spectr.) with 60.2%. Furthermore, setting two (Spectrograms + MLP) has the third-worst performance, with an average F1-score of 60.5%, precision of 70.9%, and recall of 66.5%. The accuracy is similar to the ones of the baseline models. This is followed by our linear evaluation with 64.9% F1-score, as well as setting three (Spectrograms + TE) with an average F1-score, precision, and recall of 66.1%, 76.5%, and 67.8%, respectively. The two self-supervised methods, SelfHAR and SimCLR, are better than most purely supervised baselines but not better than the SVM, the XGB, or our SelfPAB 100k.

[5] SimCLR: https://github.com/iantangc/ContrastiveLearningHAR (accessed on 2023-11-28)

[6] SelfHAR: https://github.com/iantangc/SelfHAR (accessed on 2023-11-28)

Table 3 The F1-score, precision, recall, and accuracy results of the leave-one-subject-out cross-validation (LOSO) averaged (with standard error) across all 22 subjects of the HARTH v1.2 dataset

Approach	F1-score	Precision	Recall	Accuracy
Purely supervised				
SVM	71.7 ± 2.0	76.5 ± 2.0	75.7 ± 1.5	91.9 ± 0.7
XGB	74.2 ± 1.9	76.9 ± 2.2	78.5 ± 1.2	92.1 ± 1.0
DeepConvLSTM (TS)	51.2 ± 6.2	65.3 ± 3.7	58.1 ± 6.3	81.3 ± 4.6
DeepConvLSTM (Spectr.)	60.2 ± 2.2	69.6 ± 2.5	64.5 ± 1.7	88.1 ± 1.6
Spectr. + MLP	60.5 ± 2.4	70.9 ± 2.7	66.5 ± 1.4	92.0 ± 0.7
Spectr. + TE	66.1 ± 2.0	76.5 ± 1.9	67.8 ± 1.8	91.6 ± 0.9
Self-supervised				
SelfHAR [40]	69.7 ± 2.1	76.3 ± 1.8	73.7 ± 1.7	91.1 ± 1.1
SimCLR [41]	70.9 ± 1.9	74.8 ± 1.8	73.2 ± 1.6	89.6 ± 1.0
SelfPAB (linear eval.)	64.9 ± 1.9	80.0 ± 1.8	64.9 ± 1.3	87.8 ± 1.2
SelfPAB 100k	**81.3 ± 1.3**	**83.7 ± 1.3**	**82.6 ± 1.2**	**94.6 ± 0.5**

The F1-score, the precision, and the recall are also averaged across the different activity labels. The best results are shown in bold letters and the second-best are underlined. Spectr. is the abbreviation for Spectrograms

5.2 HAR70+ overall downstream performance

The average F1-score, precision, recall, and accuracy of the HAR70+ LOSO are shown in Table 4. SelfPAB 100k achieves the best F1-score, recall, and accuracy. It has an average F1-score of 78.5%, precision of 86.5%, a recall of 78.7%, and an accuracy of 93.8%. The linear evaluation exhibits the best precision of 91.2% but the worst recall and F1-score. Again, the DeepConvLSTM experiments are among the worst results, with F1-scores of 54.7% and 59.3%. For the HAR70+, the two self-supervised methods perform worse than almost all purely supervised approaches. SelfHAR has an F1-score of 58.8%, and SimCLR 59.7%. This is followed by setting two (Spectrograms + MLP) with an F1-score of 61.9%. Setting three (Spectrograms + TE) outperforms the XGB in the F1-score (64.5%) but has the lowest precision (80.2%). All investigated methods are considerably worse than our proposed SelfPAB in F1-score and recall.

5.3 Activity recognition performance

Figure 5a shows the average F1-score (with standard error) for each activity in the HARTH v1.2 dataset. The SelfPAB 100k, the XGB, and the Spectrograms + TE experiments are visible. The XGB is either the best or second-best among the purely supervised methods; hence, we focus on it here. The Spectrograms + TE results illustrate the difference between supervised and self-supervised training of the proposed architecture. The shown activities are ordered according to the number of samples in the dataset, with sitting being the most common activity (see Fig. 3). The well-represented classes (sitting, walking, standing, cycling(sit), lying, and running) are largely dominated by good but similar results for all models. Although, Spectrograms + TE shows slightly worse results in sitting, lying, and running. Cycling (sit) has a similar average performance across the models, considering the high standard error. Shuffling is an exception here. It has almost the same amount of samples in the dataset as running (see Fig. 3) but a much lower performance across all models. Nevertheless, SelfPAB has considerably better results than the baseline XGB for shuffling. The rare classes (stairs (ascending), stairs (descending), and cycling(stand)) have, in general, much poorer performance. Despite this, we observe that the SelfPAB model performs comparably well on stairs (ascending) and stairs (descending). Cycling (stand), on the other hand, is similarly poor predicted by all models and shows a high standard error.

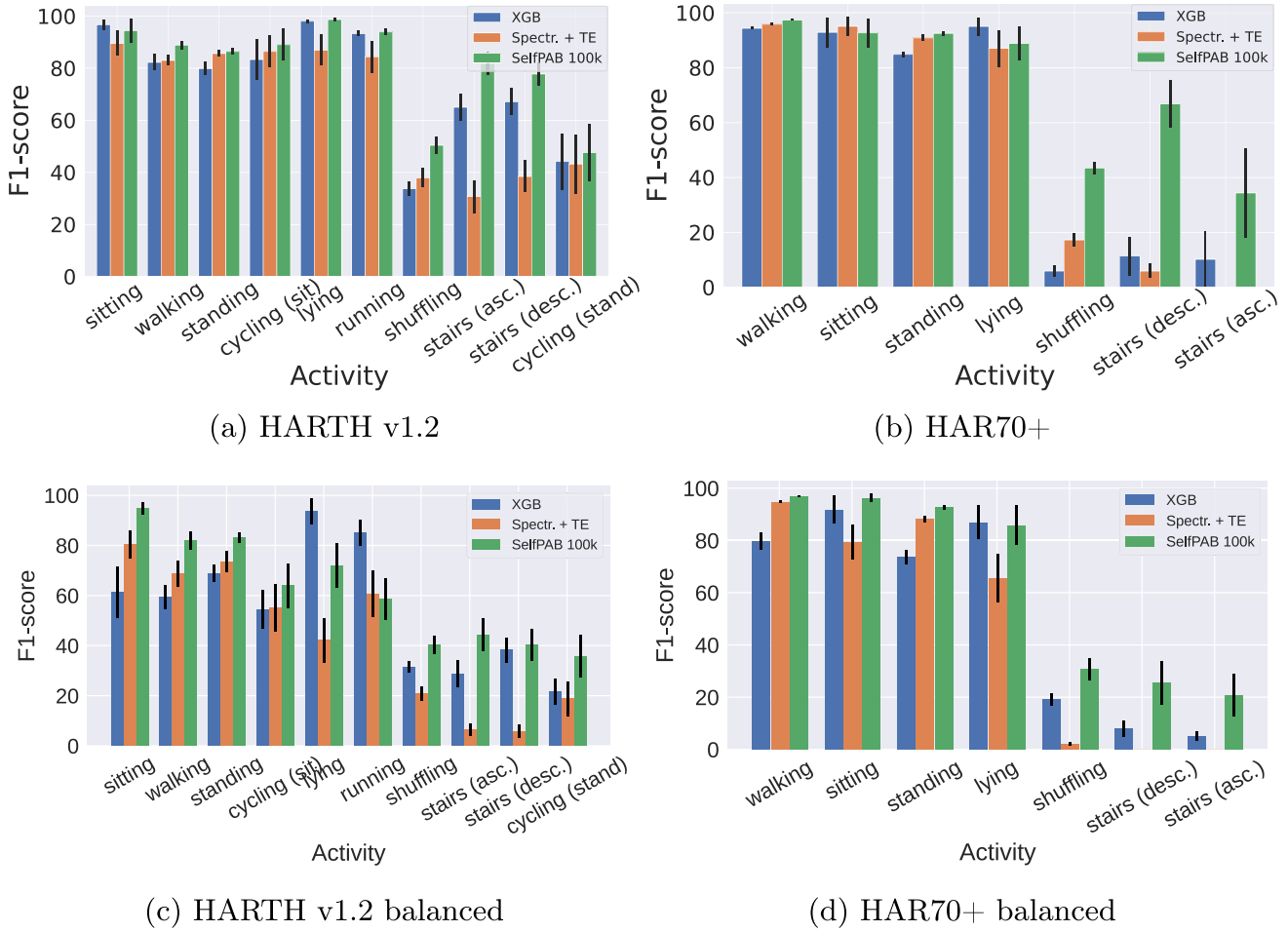
Figure 5d shows the F1-score performance for each activity of HAR70+ separately. The activities are ordered in descending order from left to right so that the left-most activity (walking) is the most common in HAR70+. The well-represented classes (walking, sitting, standing, and lying) perform similarly well across all three models, while the XGB is slightly better for lying. SelfPAB, on the other hand, has slightly higher F1-scores for walking and standing. For the rare classes (shuffling, stairs (descending), and stairs (ascending)), pre-training considerably improves the performance compared to XGB and Spectrograms + TE.

Most activities benefit from our pre-training in the HARTH v1.2 and HAR70+ datasets. However, while more frequent activities have a generally high performance for all models, less common activities show a strong F1-score increase. Hence, those activities are less often misclassified with other activities, although the amount of labeled data is low. To investigate the reason for that behavior, we conduct a further experiment. We balance the HARTH v1.2 and HAR70+ datasets through under-sampling. We limit the number of minutes per activity to exactly one minute. Hence, every activity becomes non-frequent. We train the XGB, the

Table 4 The F1-score, precision, recall, and accuracy results of the leave-one-subject-out cross-validation (LOSO) averaged (with standard error) across all 18 subjects of the HAR70+ dataset

Approach	F1-score	Precision	Recall	Accuracy
Purely supervised				
SVM	64.3 ± 2.9	80.0 ± 2.4	66.7 ± 2.5	90.6 ± 1.2
XGB	63.7 ± 2.4	85.1 ± 2.0	64.6 ± 2.2	91.3 ± 1.2
DeepConvLSTM (TS)	54.7 ± 3.3	69.8 ± 2.9	58.2 ± 3.3	88.6 ± 2.5
DeepConvLSTM (Spectr.)	59.3 ± 1.9	66.2 ± 2.1	68.8 ± 2.6	91.0 ± 1.5
Spectr. + MLP	61.9 ± 2.5	84.5 ± 2.2	63.4 ± 2.2	92.9 ± 1.2
Spectr. + TE	64.5 ± 2.5	80.2 ± 1.9	65.2 ± 2.1	93.2 ± 1.0
Self-supervised				
SelfHAR [40]	58.8 ± 2.7	87.4 ± 1.4	59.7 ± 2.7	89.4 ± 1.2
SimCLR [41]	59.7 ± 2.4	83.2 ± 1.8	61.1 ± 2.3	89.6 ± 1.2
SelfPAB (linear eval.)	54.0 ± 2.3	**91.2 ± 1.2**	55.9 ± 2.2	90.7 ± 1.2
SelfPAB 100k	**78.5 ± 2.1**	86.5 ± 1.1	**78.7 ± 1.8**	**93.8 ± 1.2**

The F1-score, the precision, and the recall are also averaged across the different activity labels. The best results are shown in bold letters and the second-best are underlined. Spectr. is the abbreviation for Spectrograms



(a) HARTH v1.2

(b) HAR70+

(c) HARTH v1.2 balanced

(d) HAR70+ balanced

Fig. 5 Average F1-scores for each activity in the HARTH v1.2 and HAR70+ datasets. The black lines show the corresponding standard errors. The activities are ordered according to their amount of minutes in the datasets, with left being the most common activity. The bottom two panels show the results when both datasets are balanced through strong under-sampling, where each activity consists of exactly one minute. Spectr. is the abbreviation for Spectrograms

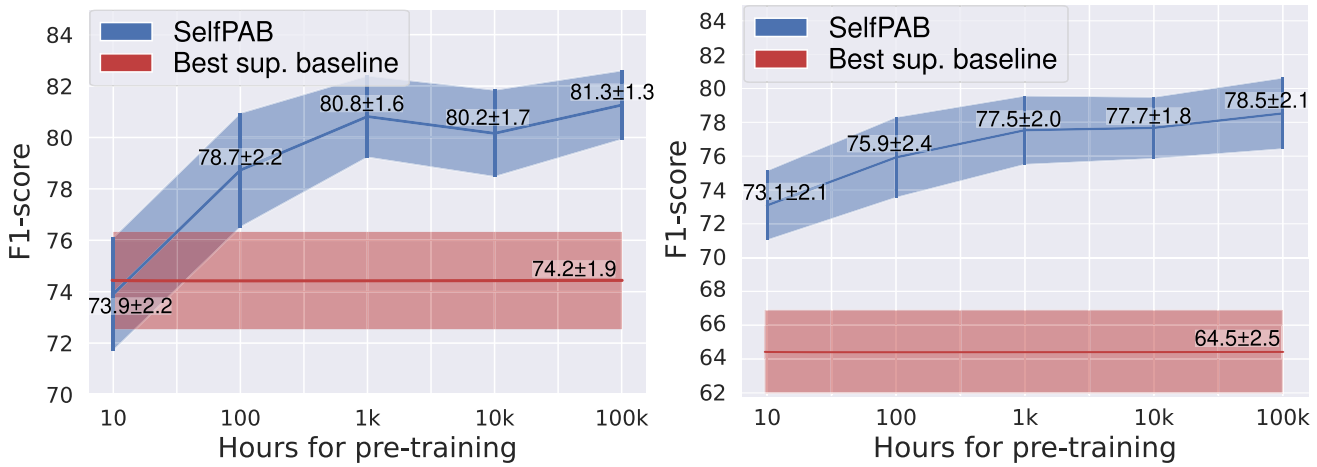
Spectr.+TE, and our SelfPAB on these balanced datasets and visualize the average F1-scores per activity in Fig. 5c and d. We observe that although the datasets are now balanced, there is still a big difference between some activities, similar to the full datasets. For example, for the HAR70+, shuffling, and stair walking are still considerably worse than the other activities, although all activities have the same number of samples. This indicates that those activities are generally hard to learn, independent of the method used. However, we can observe that SelfPAB still outperforms the other methods and exhibits comparably good performance even with only one minute per activity. The F1-score averaged across the HARTH v1.2 activities is $52.7 \pm 2.8\%$ for XGB, $42.7 \pm 2.4\%$ for Spectr. + TE, and $61.5 \pm 2.9\%$ for SelfPAB. For the HAR70+, the averaged F1-scores are $52.3 \pm 1.7\%$ for XGB, $52.9 \pm 3.2\%$ for Spectr. + TE, and $68.6 \pm 3.0\%$ for SelfPAB. Hence, SelfPAB does not need many downstream samples to learn the recognition of activities. Nevertheless, it is still recommended to use more samples, especially for the more complex classes.

5.4 Impact of the amount of unique upstream samples

The overall increase in the average F1-score with an increasing amount of pre-training hours is illustrated in Fig. 6a for the HARTH v1.2 dataset and in Fig. 6b for the HAR70+. The best supervised baselines are shown as references in red. For HARTH v1.2, it is the XGB, and for HAR70+, it is Spectr.+TE. For HARTH v1.2 (Fig. 6a), a strong performance gain is achieved when training on 1k hours compared to 10 or 100 hours. Using more pre-training data improves the performance marginally, with 10k hours being worse than 1k hours. When SelfPAB is pre-trained on at least 100 hours, it shows better performance than the best supervised baseline, with 78.7% compared to XGB's 74.2%. SelfPAB 100k has an F1-score of $81.3 \pm 1.3\%$, which is an improvement of around 7% compared to the baseline XGB. Furthermore, it generally attains a lower standard error than the best baseline when pre-trained on at least 1k hours. It is also observable that SelfPAB 10 reaches a similar high F1-score performance compared to the XGB. In the HAR70+ experiments (Fig. 6b), the F1-score increases with increasing hours used during pre-training, while the performance gain from 10 hours to 1k hours is stronger than from 1k hours to 100k hours. SelfPAB pre-trained on only 10 hours of HUNT4 outperforms the best supervised baseline, with 73.1% compared to 64.5%. The best SelfPAB model (pre-trained on 100k hours) has an improvement of around 14% compared to Spetr.+TE. It is observable that with an increase of hours used for pre-training, an increase in average F1-score occurs in both datasets. Hence, SelfPAB gets better when pre-trained on more data. In both cases, the model pre-trained on the most hours of unique upstream samples, SelfPAB 100k, achieves the best average F1-score.

5.5 PAMAP2, Opportunity, and RealWorld Downstream

Table 5 provides the average F1-scores of the PAMAP2, Opportunity, and RealWorld datasets. These are datasets with partly different sensor placements and orientations. SelfPAB



(a) HARTH v1.2 (b) HAR70+

Fig. 6 The average downstream F1-score for (a) HARTH v1.2 and (b) HAR70+ if SelfPAB is trained on 10 hours, 100 hours, 1k hours, 10k hours, and 100k hours of the HUNT4 data. The performance of the best supervised baseline is shown in red as a reference. The shaded areas represent the standard error, and the y-axis range is between 70% and 85% for HARTH v1.2, and between 62% and 85% for HAR70+

Table 5 Average F1-score results, with standard error, of the five-fold cross-validations on PAMAP2, Opportunity, and RealWorld

Approach	PAMAP2	Opportunity	RealWorld	Average
Purely supervised				
SVM	71.47 ± 3.63	80.96 ± 1.27	75.07 ± 6.35	75.83 ± 2.77
XGB	70.81 ± 3.53	82.77 ± 1.07	76.54 ± 3.15	76.71 ± 3.45
DeepConvLSTM (Spectr.)	48.27 ± 11.34	**86.77 ± 1.19**	79.41 ± 6.25	71.48 ± 11.80
Spectr. + MLP	52.63 ± 7.05	77.82 ± 0.88	75.95 ± 1.40	68.80 ± 8.10
Spectr. + TE	69.38 ± 5.11	76.09 ± 1.78	86.09 ± 1.93	77.19 ± 4.85
Self-supervised				
SelfHAR [40]	62.22 ± 5.64	70.68 ± 1.09	72.14 ± 5.85	68.35 ± 3.09
SimCLR [41]	65.29 ± 4.44	69.68 ± 1.95	82.20 ± 3.22	72.39 ± 5.07
SelfPAB (linear eval.)	59.63 ± 4.86	53.17 ± 1.37	82.68 ± 1.75	65.16 ± 12.67
SelfPAB 100k	**85.59 ± 4.15**	86.16 ± 1.27	**93.61 ± 1.75**	**88.45 ± 2.58**

The best results are shown in bold letters, and the second-best are underlined

pre-trained on 100k hours of HUNT4 performs best on the PAMAP2 and the RealWorld datasets. It is the second-best for the Opportunity. The best model for the Opportunity dataset is the purely supervised DeepConvLSTM. SelfPAB is, on average, the best model with 88.45% F1-score and Spetr. + TE is the second-best with 77.19%. Furthermore, SelfPAB performs better than the other SSL methods, SelfHAR and SimCLR, on all three datasets. However, the linear evaluation of SelfPAB exhibits the worst results with an F1-score of 65.16%, averaged across the three datasets. Hence, using a two-layer MLP as the downstream head and applying fine-tuning (SelfPAB 100k) instead of a linear classifier and no fine-tuning (SelfPAB linear eval.) makes a considerable difference in the performance.

6 Discussion

We show that most activities benefit from our proposed method. However, especially the difficult and non-frequent activities benefit from the pre-training. We showed that SelfPAB is downstream data-efficient, as it shows the best performances even under a limited amount of downstream data. Spectrograms + TE performs worse than SelfPAB pre-trained on only 10 hours, although the same architecture is used. Ten hours are less than each of the two labeled datasets HARTH v1.2 and HAR70+ contain. This observation indicates that the pre-training with weight freezing, as in SelfPAB, can be seen as a helpful initialization procedure for downstream training. It initializes the model weights to make the downstream optimization easier compared to randomly initialized weights as in Spectrograms + TE. We further show that increasing the number of unique data samples for pre-training improves the HAR downstream performance, with 100k hours leading to the best results. A similar observation was made by Kaplan et al. [20] on transformer-based language models, where the loss scales with the amount of training data. However, the effort of collecting that amount of physical activity recordings is high compared to the resulting performance gain. Future work can tackle this aspect by developing auxiliary tasks that can learn valuable representations from lower amounts of pre-training data. Nevertheless, we already show that we can achieve better (on HAR70+) and similar (on HARTH v1.2) performances than the purely supervised baselines when only using 10 hours of unlabeled data, which is less than both labeled datasets contain. And we achieve better results on both datasets when using only 100 hours. Hence, our approach can already learn useful representations even from small amounts of unlabeled data. Furthermore, the performance increase slows down after 1k hours, indicating a convergence, similar to the work of Haresamudram et al. [15]. Hence, a limitation of our study is that we do not investigate more than 100k hours to examine whether an actual convergence occurs or not. Considering the observations of Yuan et al. [49], it is expected that a performance increase can be observable when we increase the number of subjects rather than the number of samples per subject. However, in our 100k hours model, all 35,650 subjects are already utilized. Hence, we do not expect a considerable performance improvement with, e.g., 1M hours since the number of subjects will not increase, but rather similar results to the 100k hours model. Nevertheless, a further investigation is considered an interesting direction for future works.

Although SelfPAB outperforms the XGB classifier in all our experiments, the latter still exhibits good performances. An additional advantage of XGB is that it is lightweight compared to the proposed SelfPAB method. It has considerably lower training time and memory requirements. Furthermore, SelfPAB outperforms XGB mainly in non-frequent activities, for which some even SelfPAB has non-acceptable rates for real-world applications. This leads to the question of whether the computational costs of pre-training SelfPAB are worth

it. This decision depends on the goals of the user of HAR models. If the goal is to have the best performance possible, regardless of the training costs, SelfPAB is a good option. However, if a slightly worse performance is acceptable, and a lightweight model is preferred, e.g., in edge devices, we recommend using the XGB model instead. Furthermore, since SelfPAB mainly outperforms XGB on non-frequent activities, the decision of which model to choose also depends on the activities a user is interested. If the user is interested in the frequent activities only, the XGB model is recommended. However, if the non-frequent activities are of interest, we recommend using SelfPAB instead. Even if it has non-acceptable recognition rates for some of these activities, it still performs best on them compared to the other tested methods.

A limitation of our work is that we do not further investigate the influence of the proportion of masked areas on the upstream or downstream performance. Masking too much during pre-training can make the reconstruction task too difficult for the model, so it does not learn useful latent representations. It might further cause generalization issues since the downstream data is not masked compared to the highly masked upstream data. Masking too little during pre-training can make the task too straightforward, again leading to a model that does not learn meaningful latent representations. Hence, a good trade-off for the difficulty of the task has to be found through empirical exploration. Our masking hyperparameters are inspired by related works on SSL with masked reconstruction [9, 23, 24]. Exploring more masking hyperparameter assignments might improve SelfPAB's performance even more, but since SelfPAB already shows the best results in our experiments and since our main focus is the pre-training on a large unlabelled dataset, we consider an investigation in that direction as out of the scope of this work.

Our linear evaluation revealed a further limitation of our method. While a linear classifier can be used to achieve comparably good results for datasets that use the same sensor setup as HUNT4 (HARTH v1.2 and HAR70+), datasets that use different sensor placements (PAMAP2, Opportunity, and RealWorld) perform worst on average using a simple linear classifier. This indicates that the latent representations learned by SelfPAB are biased towards the HUNT4 sensor placements. However, as our other SelfPAB results show, using a two-layer MLP as the downstream head in combination with fine-tuning the upstream model can compensate for the sensor placement difference. The PAMAP2, Opportunity, and RealWorld results increase considerably when doing so, becoming the best results in our experiments. Therefore, our experiments indicate that SelfPAB is generalizable across different sensor placements if fine-tuning is performed during downstream training and a two-layer MLP is used as the downstream head. Nevertheless, we cannot know how well SelfPAB performs on sensor placements that differ considerably from the evaluated ones. More tests on publicly available datasets with diverse placements are required to answer how well SelfPAB generalizes across sensor placements. However, we consider such an investigation to be outside this work's scope.

Furthermore, we observe that SelfPAB has a better F1-score than the other two investigated SSL methods, SelfHAR and SimCLR, in all our experiments. We attribute this to two main differences between our method and the other two. First, SelfPAB uses a transformer encoder as architecture, while SelfHAR and SimCLR use convolutional neural networks. Transformers have the inherent advantage of being able to learn temporal dependencies very well. While SimCLR and SelfHAR have access to a single two-second window, SelfPAB has access to 599 one-second windows at the same time, and due to its architecture, it can learn the dependencies between these 599 windows [45]. Second, transformer models have been shown to scale very well with the amount of training data [20]. One disadvantage of SelfPAB is the number of parameters. SimCLR has around 200k parameters, and SelfHAR has around 400k. SelfPAB, on the other hand, has around 60M parameters, which might be too high depending on the application. We apply our method to epidemiological data and perform all computations on a server. Hence, no real-time computation is required, making SelfPAB a good candidate. Although SelPAB shows the best results in our experiments compared to other SSL methods, it remains an open question of how well it performs compared to upstream models that were trained on even more data, e.g., the one pre-trained by Yuan et al. [49] on the UK-Biobank dataset. However, since their method was trained on a single accelerometer and ours on two accelerometers, it is not possible to compare them directly. Hence, we refer to such an investigation for future work.

With our state-of-the-art results, we lay the foundation for more accurate public health studies using SelfPAB. Therefore, our method not only contributes to general artificial intelligence research with our pre-trained, publicly available model but also has an indirect impact on society through future epidemiological studies based on SelfPAB [19].

7 Conclusion

Inspired by the recent success of self-supervised machine learning and the large-scale HUNT4 data corpus, we make three contributions in this paper.

1) We implement SelfPAB, a self-supervised representation learning approach for human activity recognition. It is trained in two steps. First, a transformer encoder network is pre-trained on the large-scale accelerometer-based physical

activity dataset, HUNT4. The network learns physical activity representations by solving an auxiliary task of reconstructing masked parts of accelerometer signal spectrograms. Second, the pre-trained network is used as a feature extractor for downstream human activity recognition.

2) SelfPAB achieves better results than purely-supervised approaches, and we indicate that increasing the amount of unique pre-training samples leads to an increase in the downstream HAR performance.

3) We make the new HAR70+ dataset and our pre-trained model publicly available.

For future research, further downstream training on datasets with sensor placements that differ considerably from the HUNT4 ones would be interesting to investigate. Such an investigation would reveal in more detail how robust SelfPAB is regarding diverse sensor placements. Furthermore, the fact that two separate sensors record the data can be used to design more innovative pre-training objectives.

The ever-growing community of physical activity behavior research based on accelerometer (attached to the thigh and lower back) measurements will acquire new knowledge about the influence of physical activity behavior on public health by using our SelfPAB method.

Author Contributions Conceptualization: Aleksej Logacjov, Sverre Herland; Methodology: Aleksej Logacjov, Sverre Herland; Model implementation: Aleksej Logacjov, Sverre Herland; Formal analysis and investigation: Aleksej Logacjov, Sverre Herland; Resources: Astrid Ustad, Kerstin Bach; Writing - original draft preparation: Aleksej Logacjov; Writing - review and editing: Sverre Herland, Kerstin Bach, Astrid Ustad; Visualization: Aleksej Logacjov, Sverre Herland; Supervision: Kerstin Bach; Project Administration: Kerstin Bach

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). NTNU Health, Norwegian University of Science and Technology (grant no. 81771516)

Availability of Data and Materials The HAR70+ dataset presented in this article is publicly available on https://github.com/ntnu-ai-lab/harth-ml-experiments/tree/main/har70plus (accessed on 2023-12-19). The pre-trained models are publicly available on https://github.com/ntnu-ai-lab/SelfPAB (accessed on 2024-02-13).

Code Availibility The source code used for the experiments is publicly available on https://github.com/ntnu-ai-lab/SelfPAB (accessed on 2024-02-13).

Declarations

Conflict of Interest The authors declare no conflict of interest.

Ethics Approval The HAR70+ data collection was approved by the Norwegian Centre for Research Data (protocol code 515701). Date of approval: 16 March 2021.

Consent to Participate Informed consent was obtained from all subjects involved in the study.

Consent for Publication Consent for publication of the dataset was obtained from all subjects involved in the study.

References

1. Bach K, Kongsvold A, Bårdstu H et al (2021) A machine learning classifier for detection of physical activity types and postures during free-living. J Meas Phys Behav -1(aop):1–8 https://doi.org/10.1123/jmpb.2021-0015
2. Brown T, Mann B, Ryder N et al (2020) Language models are few-shot learners. In: Advances in neural information processing systems, vol 33. Curran Associates, Inc., pp 1877–1901
3. Chan Chang S, Doherty A (2021) Capture-24: activity tracker dataset for human activity recognition. University of Oxford
4. Chavarriaga R, Sagha H, Calatroni A et al (2013) The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recognit Lett 34(15):2033–2042. https://doi.org/10.1016/j.patrec.2012.12.014
5. Chi PH, Chung PH, Wu TH et al (2021) Audio ALBERT: a lite BERT for self-supervised learning of audio representation. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, Shenzhen, China, pp 344–350 https://doi.org/10.1109/SLT48900.2021.9383575
6. Chuang CY, Robinson J, Yen-Chen L et al (2020) Debiased contrastive learning. https://doi.org/10.48550/arXiv.2007.00224
7. Cleland I, Kikhia B, Nugent C et al (2013) Optimal placement of accelerometers for the detection of everyday activities. Sensors (Basel, Switzerland) 13(7):9183–9200. https://doi.org/10.3390/s130709183
8. Demrozi F, Pravadelli G, Bihorac A et al (2020) Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey. IEEE Access 8:210816–210836. https://doi.org/10.1109/ACCESS.2020.3037715
9. Devlin J, Chang MW, Lee K et al (2019) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
10. Doherty A, Jackson D, Hammerla N et al (2017) Large scale population assessment of physical activity using wrist worn accelerometers: the UK biobank study. PLoS ONE 12(2):e0169649. https://doi.org/10.1371/journal.pone.0169649
11. Fullerton E, Heller B, Munoz-Organero M (2017) Recognizing human activity in free-living using multiple body-worn accelerometers. IEEE Sens J 17(16):5290–5297. https://doi.org/10.1109/JSEN.2017.2722105
12. Gulati A, Qin J, Chiu CC et al (2020) Conformer: convolution-augmented transformer for speech recognition. arXiv:2005.08100
13. Haresamudram H, Beedu A, Agrawal V et al (2020) Masked reconstruction based self-supervision for human activity recognition. In: Proceedings of the 2020 international symposium

on wearable computers. Association for Computing Machinery, New York, USA, ISWC '20, pp 45–49, https://doi.org/10.1145/3410531.3414306
14. Haresamudram H, Essa I, Plötz T (2021) Contrastive predictive coding for human activity recognition. Proc ACM Interac Mob Wearable Ubiquit Technol 5(2):65:1–65:26 https://doi.org/10.1145/3463506
15. Haresamudram H, Essa I, Plötz T (2022) Assessing the state of self-supervised human activity recognition using wearables. Proc ACM Interac Mob Wearable Ubiquit Technol 6(3):116:1–116:47. https://doi.org/10.1145/3550299
16. He K, Zhang X, Ren S et al (2016) Identity mappings in deep residual networks. In: Leibe B, Matas J, Sebe N et al (eds) Computer vision – ECCV 2016. Springer International Publishing, Cham, Lecture Notes in Computer Science, pp 630–645 https://doi.org/10.1007/978-3-319-46493-0_38
17. Jain Y, Tang CI, Min C et al (2022) ColloSSL: collaborative self-supervised learning for human activity recognition. Proc ACM Interac Mob Wearable Ubiquit Technol 6(1):17:1–17:28. https://doi.org/10.1145/3517246
18. Jaiswal A, Babu AR, Zadeh MZ et al (2021) A survey on contrastive self-supervised learning. Technologies 9(1):2. https://doi.org/10.3390/technologies9010002
19. Jiang Y, Li X, Luo H et al (2022) Quo vadis artificial intelligence? Discov Artif Intell 2(1):4. https://doi.org/10.1007/s44163-022-00022-8
20. Kaplan J, McCandlish S, Henighan T et al (2020) Scaling laws for neural language models. https://doi.org/10.48550/arXiv.2001.08361
21. Khaertdinov B, Ghaleb E, Asteriadis S (2021) Contrastive self-supervised learning for sensor-based human activity recognition. In: 2021 IEEE international joint conference on biometrics (IJCB). IEEE, Shenzhen, China, pp 1–8, https://doi.org/10.1109/IJCB52358.2021.9484410
22. Le-Khac PH, Healy G, Smeaton AF (2020) Contrastive representation learning: a framework and review. IEEE Access 8:193907–193934. https://doi.org/10.1109/ACCESS.2020.3031549
23. Liu AT, Yang Sw, Chi PH et al (2020) Mockingjay: unsupervised speech representation learning with deep bidirectional transformer encoders. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp 6419–6423. https://doi.org/10.1109/ICASSP40776.2020.9054458, arxiv:1910.12638
24. Liu AT, Li SW, Lee Hy (2021) TERA: self-supervised learning of transformer encoder representation for speech. IEEE/ACM Trans Audio Speech Lang Process 29:2351–2366. https://doi.org/10.1109/TASLP.2021.3095662
25. Liu D, Abdelzaher T (2021) Semi-supervised contrastive learning for human activity recognition. In: 2021 17th international conference on distributed computing in sensor systems (DCOSS). IEEE, Pafos, Cyprus, pp 45–53, https://doi.org/10.1109/DCOSS52077.2021.00019
26. Liu X, Zhang F, Hou Z et al (2021) Self-supervised learning: generative or contrastive. IEEE Transactions on Knowledge and Data Engineering p 1. https://doi.org/10.1109/TKDE.2021.3090866
27. Logacjov A, Bach K, Kongsvold A et al (2021) HARTH: a human activity recognition dataset for machine learning. Sensors 21(23):7853. https://doi.org/10.3390/s21237853
28. Logacjov A, Herland S, Ustad A, Bach K (2023) Large-Scale Pre-Training for Dual-Accelerometer Human Activity Recognition. Norsk IKT-konferanse for forskning og utdanning, 1
29. Mao HH (2020) A survey on self-supervised pre-training for sequential transfer learning in neural networks. https://doi.org/10.48550/arXiv.2007.00800
30. Ahmadi MN, Brookes D, Chowdhury A, Pavey T, Trost SG (2020) Free-living evaluation of laboratory-based activity classifiers in preschoolers. Med Sci Sports Exerc 52(5):1227–1234. https://doi.org/10.1249/mss.0000000000002221
31. Narayanan A, Stewart T, Mackay L (2020) A dual-accelerometer system for detecting human movement in a free-living environment. Med Sci Sports Exerc 52(1):252–258. https://doi.org/10.1249/MSS.0000000000002107
32. NTNU (2022) HUNT4 - The Trøndelag Health Study - NTNU. https://www.ntnu.edu/hunt/hunt4. Accessed 04 Aug 2022
33. Ordóñez FJ, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115. https://doi.org/10.3390/s16010115
34. Reiss A, Stricker D (2012) Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th international symposium on wearable computers. IEEE, Newcastle, UK, pp 108–109, https://doi.org/10.1109/ISWC.2012.13
35. Saeed A, Ozcelebi T, Lukkien J (2019) Multi-task self-supervised learning for human activity detection. Proc ACM Interact Mobil Wearable Ubiquit Technol 3(2):61:1-61:30. https://doi.org/10.1145/3328932
36. Saeed A, Salim FD, Ozcelebi T et al (2021) Federated self-supervised learning of multisensor representations for embedded intelligence. IEEE Internet Things J 8(2):1030–1040. https://doi.org/10.1109/JIOT.2020.3009358
37. Stewart T, Narayanan A, Hedayatrad L et al (2018) A dual-accelerometer system for classifying physical activity in children and adults. Med Sci Sports Exerc 50(12):2595–2602. https://doi.org/10.1249/MSS.0000000000001717
38. Sztyler T, Stuckenschmidt H (2016) On-body localization of wearable devices: an investigation of position-aware activity recognition. In: 2016 IEEE international conference on pervasive computing and communications (PerCom), pp 1–9, https://doi.org/10.1109/PERCOM.2016.7456521
39. Taghanaki SR, Rainbow M, Etemad A (2021) Self-supervised human activity recognition by learning to predict cross-dimensional motion. 2021 International symposium on wearable computers, pp 23–27. https://doi.org/10.1145/3460421.3480417, arxiv:2010.13713
40. Tang CI, Perez-Pozuelo I, Spathis D et al (2021) SelfHAR: improving human activity recognition through self-training with unlabeled data. Proc ACM Interac Mob Wearable Ubiquit Technol 5(1):1–30. https://doi.org/10.1145/3448112, arxiv:2102.06073
41. Tang CI, Perez-Pozuelo I, Spathis D et al (2021) Exploring contrastive learning in human activity recognition for healthcare. https://doi.org/10.48550/arXiv.2011.11542
42. Tonekaboni S, Eytan D, Goldenberg A (2021) Unsupervised representation learning for time series with temporal neighborhood coding. arXiv:2106.00750
43. Twomey N, Diethe T, Fafoutis X et al (2018) A Comprehensive study of activity recognition using accelerometers. Informatics 5(2):27. https://doi.org/10.3390/informatics5020027
44. Ustad A, Logacjov A, Trollebø SØ et al (2023) Validation of an activity type recognition model classifying daily physical behavior in older adults: the HAR70+ model. Sensors 23(5):2368. https://doi.org/10.3390/s23052368
45. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. arXiv:1706.03762
46. Wang J, Zhu T, Chen L et al (2022) Negative selection by clustering for contrastive learning in human activity recognition. arXiv:2203.12230
47. Wang J, Zhu T, Gan J et al (2022) Sensor data augmentation by resampling for contrastive learning in human activity recognition. arXiv:2109.02054

48. Yu J, Wang Z, Vasudevan V et al (2022) CoCa: contrastive captioners are image-text foundation models. https://doi.org/10.48550/arXiv.2205.01917
49. Yuan H, Chan S, Creagh AP et al (2023) Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. https://doi.org/10.48550/arXiv.2206.02909