Temporal Action Localization for Inertial-based Human Activity Recognition

MARIUS BOCK, University of Siegen, Germany
MICHAEL MOELLER, University of Siegen, Germany
KRISTOF VAN LAERHOVEN, University of Siegen, Germany

As of today, state-of-the-art activity recognition from wearable sensors relies on algorithms being trained to classify fixed windows of data. In contrast, video-based Human Activity Recognition, known as Temporal Action Localization (TAL), has followed a segment-based prediction approach, localizing activity segments in a timeline of arbitrary length. This paper is the first to systematically demonstrate the applicability of state-of-the-art TAL models for both offline and near-online Human Activity Recognition (HAR) using raw inertial data as well as pre-extracted latent features as input. Offline prediction results show that TAL models are able to outperform popular inertial models on a multitude of HAR benchmark datasets, with improvements reaching as much as 26% in F1-score. We show that by analyzing timelines as a whole, TAL models can produce more coherent segments and achieve higher NULL-class accuracy across all datasets. We demonstrate that TAL is less suited for the immediate classification of small-sized windows of data, yet offers an interesting perspective on inertial-based HAR – alleviating the need for fixed-size windows and enabling algorithms to recognize activities of arbitrary length. With design choices and training concepts yet to be explored, we argue that TAL architectures could be of significant value to the inertial-based HAR community. The code and data download to reproduce experiments is publicly available via github.com/mariusbock/tal_for_har.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**; • **Computing methodologies → Neural networks**.

Additional Key Words and Phrases: Deep Learning, Inertial-based Human Activity Recognition, Body-worn Sensors, Temporal Action Localization

ACM Reference Format:
Marius Bock, Michael Moeller, and Kristof Van Laerhoven. 2024. Temporal Action Localization for Inertial-based Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 4, Article 174 (December 2024), 19 pages. https://doi.org/10.1145/3699770

1 INTRODUCTION

The recognition of performed activities through wearable sensors such as Inertial Measurement Units (IMUs) has shown to be of significant value in areas such as health care or the support of complex work processes [8]. With the works of many researchers exploring lightweight architectures [70], much of the success of inertial-based models has stemmed from applying them in an online-fashion on wearable edge devices [42, 56]. As shown to work on a multitude of HAR-related scenarios, it has established itself within the inertial-based community to employ a window-based prediction approach using a predefined window size and overlap. With classification algorithms being tasked to assign a label to each window individually, this approach has enabled inertial-based architectures to classify newly incoming data at will. As a too large window size may include multiple fast actions

Authors' addresses: Marius Bock, marius.bock@uni-siegen.de, University of Siegen, Ubiquitous Computing, Computer Vision, Siegen, Germany; Michael Moeller, michael.moeller@uni-siegen.de, University of Siegen, Computer Vision, Siegen, Germany; Kristof Van Laerhoven, kvl@eti.uni-siegen.de, University of Siegen, Ubiquitous Computing, Siegen, Germany.

within a single window, and a too short window may not be enough to capture a complete lengthier action, the chosen size of the fixed window has become a crucial parameter within traditional recognition systems.

A persistent trend in Deep Learning has been the applicability of machine learning concepts such as self-attention [55] to other areas and application scenarios than originally introduced for. With significant progress having been made since the introduction of deep neural architectures such as DeepConvLSTM [38], researchers have followed this trend and have continuously worked on improving the architectural design of networks by incorporating newly introduced techniques (see, e.g., [70]). A promising recent approach in video-based Human Activity Recognition (HAR) is Temporal Action Localization (TAL), which aims to locate activity segments, defined by a class label, start, and end point, within an untrimmed video. Even though introduced architectures have almost doubled in performance over the last 5 years on existing datasets like THUMOS-14 [23], results on large corpora such as EPIC-KITCHENS-100 [14] and Ego4D [18] show that the prediction problem is far from being saturated.

Despite sharing a mutual goal, the TAL community as opposed to the inertial community does not follow an online-based prediction approach, but aims to classify recorded times sequences in an offline fashion, taking advantage of learnable temporal dependencies along the complete timeline. Furthermore, by following a segment-based prediction with activity boundaries being learned via regression-based loss, the "window size problem" is alleviated with methods being flexible enough to recognize both short and long lasting activities. TAL models have recently been shown to be capable of being trained using raw inertial data [7], marking the first instance of such vision-based models being applied in the context of inertial-based HAR. When compared with popular inertial models, results on one particular dataset have already shown that TAL models can produce more coherent and less fragmented predictions while maintaining performance in terms of traditional classification metrics. In light of activity recognition systems having also been deployed to provide offline analysis of streams of prerecorded data [4, 5, 15, 22, 54, 65], this work sets out to further examine the applicability of Temporal Action Localization for inertial-based HAR. Our contributions in this paper are three-fold:

(1) We demonstrate the capabilities of a novel approach to inertial-based HAR, being single-stage TAL, to outperform popular, window-based inertial models on a multitude of wearable activity recognition benchmark datasets in an offline-prediction scenario.
(2) Complementing traditional, window-wise classification metrics, we introduce a set of unexplored, segment-based evaluation metrics for inertial-based HAR, which are based around the scalar evaluation metric mean Average Precision [2].
(3) Though our results demonstrate the superiority of inertial-based models in an online prediction scenario, we show that TAL models can be applied in a near-online fashion, functioning e.g. a server-side prediction tool which provides prediction with a lag of 1 minute.

2 RELATED WORK

Inertial-based Human Activity Recognition. With on-body sensors providing a robust and non-intrusive way to monitor participants along long stretches of time, research conducted in the area of inertial-based HAR has worked towards the automatic interpretation of one or multiple sensor streams to reliably detect activities e.g. in the context of providing medical support or providing guidance during complex work processes [8]. With deep neural networks having established themselves as the de facto standard in inertial-based HAR, DeepConvLSTM [38] as well as the models introduced at a later point follow a similar prediction scenario design applying a sliding window approach which groups concurrent data points for classification. In their original publication Ordóñez and Roggen [38] have found a combination of convolutional and recurrent layers to produce competitive results, with the idea being to model temporal dependencies amongst automatically extracted discriminative features within a sliding window in order to classify it correctly as one of the N activity classes or, if applicable, NULL-class.

Building on the idea of combining these two types of layers, researchers have worked on extending the original DeepConvLSTM or have introduced their own architectural designs [1, 6, 11–13, 31, 35, 36, 40, 53, 58–61, 67, 70]. A simple modification of the DeepConvLSTM is the shallow DeepConvLSTM [6]. Contradicting the popular belief that one needs at least two recurrent layers when dealing with time series data [24], Bock et al. [6] demonstrated that removing the second LSTM layer within the original DeepConvLSTM architecture results in significant increases in predictive performance on a multitude of HAR benchmark datasets while also decreasing the number of learnable parameters. Furthermore, with the original DeepConvLSTM only being able to learn temporal dependencies within a sliding window, the shallow DeepConvLSTM applies the LSTM across batches, effectively making the batches the sequence which is to be learned by the LSTM. This dimension flip, along with a non-shuffled training dataset, enables the architecture to learn temporal dependencies amongst a batched input. The same year as the shallow DeepConvLSTM, Abedin et al. [1] introduced Attend-and-Discriminate, a deep neural network architecture following the idea of the original DeepConvLSTM by combining both convolutional and recurrent layers, yet further adding a cross-channel interaction encoder using self-attention as well as attention mechanism to the recurrent parts of the network. In 2022 Zhou et al. [70] proposed TinyHAR, a lightweight HAR model that uses a transformer encoder block along with means of cross-channel fusion via a fully connected layer and a final self-attention layer which aims to learn the contribution of each outputted time step produced by the recurrent parts individually.

Video-based Human Activity Recognition. Classifying videos in the context of Human Activity Recognition can be broadly categorized into three main application scenarios: Action Recognition, which aims to classify trimmed videos into one of C activity classes; Action Anticipation, which aims to predict the next likely activities after observing a preceding video sequence; and Temporal Action Localization (TAL), which seeks to identify and locate all activity segments within an untrimmed video. With the inertial-based benchmark datasets consisting of a multitude of activities, TAL is to be considered most comparable to inertial-based HAR. Unlike popular intertial architectures though, TAL models aim to predict all segments within an untrimmed video at once. Existing TAL methods can broadly be categorized into two categories: two-stage and one-stage recognition systems. Two-stage recognition system [3, 17, 27, 29, 32, 41, 49, 51, 62, 64, 68, 69, 71] divide the process of identifying actions within an untrimmed video into two subtasks. First, during proposal generation, candidate segments are generated, which are then, during the second step, classified into one of C activity classes and iteratively refined regarding their start and end times. Contrarily, single-stage models [10, 28, 30, 33, 34, 37, 47, 48, 52, 63, 66] do not rely on activity proposals, directly aiming to localize segments along their class label and refined boundaries. In 2022 Zhang et al. [66] introduced the ActionFormer, a lightweight, single-stage TAL model which unlike previously introduced single-stage architectures does not rely on pre-defined anchor windows. In line with the success of transformers in other research fields, Zhang et al. [66] demonstrated their applicability for TAL, outperforming previously held benchmarks on several TAL datasets [14, 20, 23] by a significant margin. Surprisingly, a year later, TemporalMaxer suggested removing transformer layers within the ActionFormer, arguing that feature embeddings are already discriminative enough [52]. Though being more lightweight than the ActionFormer, the TemporalMaxer showed to outperform its precedent during benchmark analysis. Similarly to the TemporalMaxer, Shi et al. [47] introduced TriDet, which suggested altering ActionFormer in two ways. First, to mitigate the risk of rank-loss, self-attention layers are replaced with SGP layers. Second, the regression head in the decoder is replaced with a Trident head, which improves imprecise boundary predictions via an estimated relative probability distribution around each timestamp's activity boundaries.

3 TEMPORAL ACTION LOCALIZATION FOR INERTIAL-BASED HAR

As the inertial-based HAR and TAL communities deal with inherently different modalities, both communities have developed distinct predictive pipelines and algorithms tailored to the challenges and characteristics of their

respective modalities (see Figure 1). The objective of both inertial activity recognition and TAL is to predict all activities present in an untrimmed timeline. Given an input data stream X of a sample participant, both the inertial and TAL communities start by applying a sliding window approach which shifts over X, dividing the input data into windows, e.g. of one second duration with a 50% overlap between consecutive windows. This process results in $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ being discretized into $t = \{0, 1, ..., T\}$ time steps, where T is the number of windows for each participant. It is important to note that the TAL models do not use raw data as input but are instead trained using feature embeddings extracted from each individual sliding video clip, which are extracted using pre-trained methods such as [9].

Given all sliding windows associated with an untrimmed sequence, inertial activity recognition models aim to predict an activity label a_t for each sliding window $\mathbf{x}_t$, where a belongs to a predefined set of activity labels, $a_t \in 1, ..., C$. To do so, the sliding windows are batched and fed through, e.g., a deep neural network, such as the DeepConvLSTM [38]. The resulting activity labels for each window are then compared to the true activity labels from the ground truth data, and classification metrics like accuracy or F1-score are calculated. Contrarily, TAL models aim to identify and localize segments of actions within the untrimmed data stream, which can span across multiple windows. To achieve this algorithms are trained to predict activity segments $Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, where N varies across participants. Each activity instance $\mathbf{y_i} = (s_i, e_i, a_i)$ is defined by its starting time s_i (onset), end time e_i (offset) and its associated activity label a_i, where $s_i \in [0, T]$, $e_i \in [0, T]$, and $a_i \in \{1, ..., C\}$.

As the input data used to train TAL models is a collection of 1-dimensional feature embeddings, the 2 d-dimensional, windowed inertial data commonly found in the inertial-based HAR community, cannot directly be used to train TAL models. The following will thus describe two preprocessing methods which can be used to convert the inertial data into a format such that it can be used as input for TAL models.

Vectorization of raw inertial data. Since both communities employ a sliding window approach but feed data to their models using different dimensions, Bock et al. [7] proposed a simple, yet effective preprocessing step. This step converts the 2-dimensional, windowed inertial data, as used by inertial architectures, into a 1-dimensional feature embedding suitable for training TAL models. The preprocessing method as detailed in Figure 2 involves concatenating the different sensor axes of each window, converting the input data to be a collection of 1-dimensional feature embedding vectors $\mathbf{x}_t \in \mathbb{R}^{1 \times WS}$ where W is the number of samples within a window, and S is the number of sensor axes. More specifically, given a sliding window $\mathbf{x}_{SW} \in \mathbb{R}^{W \times S}$, we vectorize the two-dimensional matrix as follows:

$$\mathbf{x}_{SW} = \begin{bmatrix} x_{11} & \dots & x_{1S} \\ \vdots & \ddots & \\ x_{W1} & & x_{WS} \end{bmatrix} \rightarrow \text{vec}(\mathbf{x}_{SW}) = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1S} \\ x_{2S} \\ \vdots \\ x_{WS} \end{bmatrix} \tag{1}$$

Two-stage training via prepended inertial models. In order to encode videos and come up with a discriminative, latent feature representation, TAL models usually resort to using extracted feature embeddings from models pretrained on large vision corpuses like Kinetics-400 [25]. Inspired by this, we propose a second variant on how to use inertial data as input to TAL models (see Figure 3) which involves using features extracted from a separately trained inertial model as latent representation of each sliding window. Specifically, a LOSO cross-validation step within the two stage training consists of 1. training the inertial model on all but the validation data, 2. use

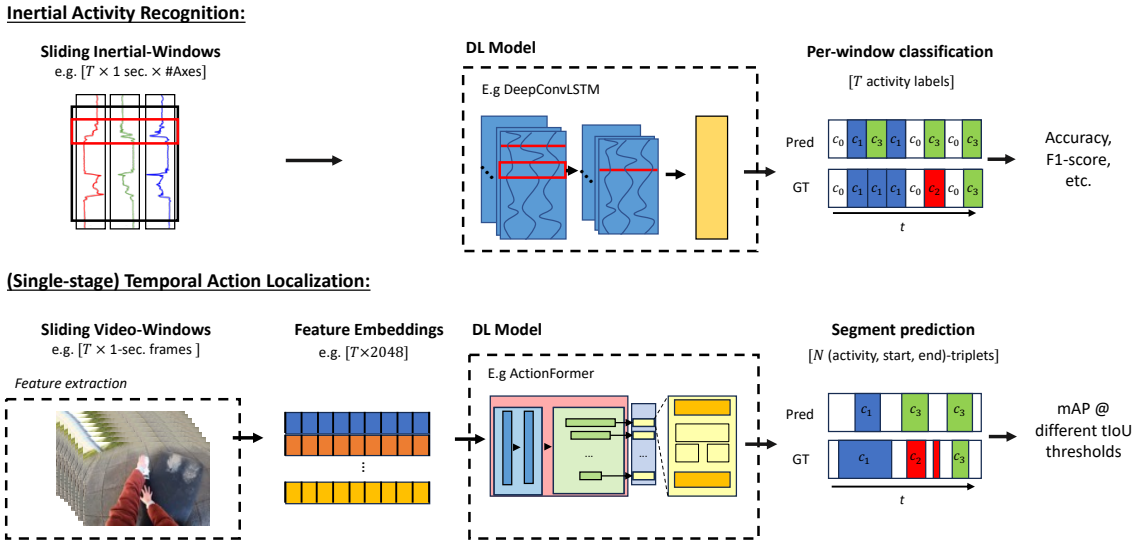

Fig. 1. Overview of the prediction pipelines applied in inertial-based activity recognition and single-stage Temporal Action Localization (TAL). Both apply a sliding window to divide input data into windows of a certain duration (e.g. one second). TAL models do not use raw data as input but are applied on per-clip, pre-extracted feature embeddings. Inertial activity recognition models predict activity labels for each sliding window, which are used for calculating classification metrics such as accuracy and F1. TAL models predict activity segments, defined by a label, start and end points, and are evaluated with mean Average Precision (mAP) applied at different temporal Intersection over Union (tIoU) thresholds.

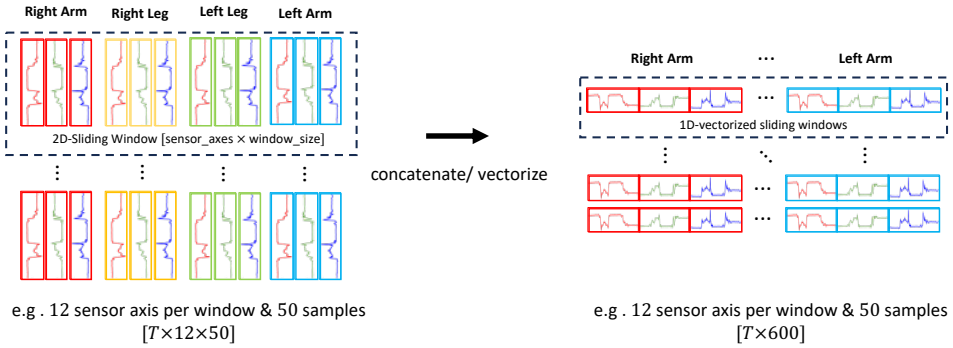

Fig. 2. Visualization of the applied vectorization on top of windowed inertial data assuming four 3D-inertial sensors and a sliding window size of 50 samples. Each 2D-sliding-window of size $[50 \times 12]$ is vectorized by concatenating each of the axes one after another. Resulting 1D-embedding vectors, being of size $[1 \times 600]$, can be used to train TAL models.

the trained inertial network to extract latent feature representations of each window within the training and validation data, and 3. using the extracted features as clip-wise feature embeddings to train a TAL model.

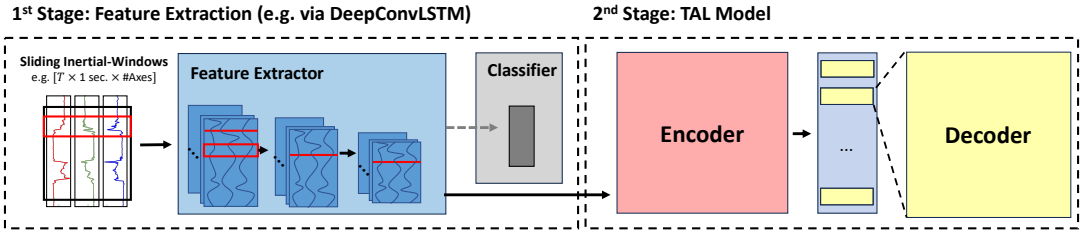

Fig. 3. Visualization of the applied two-stage training process. The first stage involves training e.g. a classic DeepConvLSTM as introduced by Ordóñez and Roggen [38]. Once the first-stage training has finished, the classifier is omitted from the model such that latent features can be extracted. The 1-dimensional, window-wise features are then used as input embeddings for the second stage, i.e. training a TAL model.

3.1 TAL Architectures Overview

In light of the recent success of transformer-based models in e.g. Natural Language Processing and Computer Vision, Zhang et al. [66] proposed the *ActionFormer*, is an end-to-end trainable transformer-based TAL architecture. Unlike other single-stage TAL approaches, it does not rely on pre-defined anchor windows. The architecture, as illustrated in Figure 4, combines multiscale feature representations with local self-attention and is trained through a classification and regression loss calculated by a light-weighted decoder. Building up on the ActionFormer architecture, Tang et al. [52] and Shi et al. [47] proposed the *TemporalMaxer* and *TriDet* model respectively. Within the *TriDet* model projection and transformer layers of the ActionFormer are replaced with fully-convolutional SGP layers and the regression head is replaced by a trident regression head which claims to improve imprecise boundary predictions. Contrarily, the TemporalMaxer suggests modifying the encoder of the ActionFormer to employ solely max pooling and remove all transformer-based layers, as, according to the authors, this does not come at the cost of a lost in information and predictive performance.

In order to predict activity segments $Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ within an input video, the ActionFormer, TemporalMaxer and TriDet model all follow the same sequence labeling problem formulation for action localization. That is, given a set of feature input vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, a model aims to classify each timestamp as either one of the activity categories C or as background (or null) class and regress the distance towards the timestamp's corresponding segment's start and end point. More specifically, given the input vectors X a model aims to learn to label the sequence as

$$X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\} \rightarrow \hat{\mathbf{Y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_T\}, \tag{2}$$

where $\hat{\mathbf{y}}_\mathbf{t} = (p(a_t), d_t^s, d_t^e)$ at timestamp t is defined by a probability vector $p(a_t)$ indicating the class-wise probability of the timestamp being classified as one of the activity categories C and $d_t^s > 0$ and $d_t^e > 0$ being the distance between the current timestamp t and the current segment's onset and offset. Note that d_t^s and d_t^e are not defined if the timestamp is to be classified as background. The sequence labeling formulation can then be easily decoded to activity segments with:

$$a_t = \arg\max p(a_t), \quad s_t = t - d_t^s, \quad \text{and} \quad e_t = t + d_t^e \tag{3}$$

The authors of the ActionFormer, TemporalMaxer, and TriDet models attribute much of their models' performance to the constructed multi-layer feature pyramid. The feature pyramid within each of the three models downsamples the input sequence of sliding windows multiple times to create representations of a participant's data stream at different temporal granularities [47, 52, 66]. Using this method, the TAL models can learn both short and long temporal dependencies across a participants's timeline, as the temporal distance between two

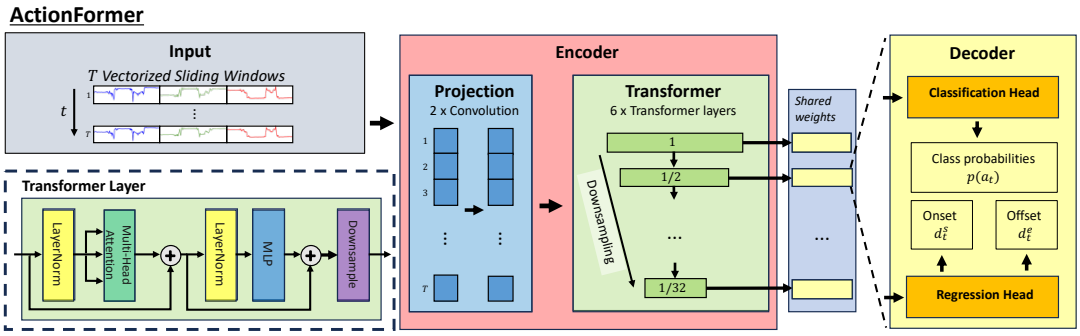

Fig. 4. Architecture overview of the ActionFormer proposed by Zhang et al. [66]. The architecture follows a encoder-decoder structure. The encoder encodes input sequences into a feature pyramid, which captures information at various temporal scales. The decoder, consisting of a classification and regression head, then decodes each timestamp within the feature pyramid to sequence labels, i.e. a class probability vector and the timestamp's activity onset and offset distance. The TriDet[47] and TemporalMaxer[52] both follow the same encoder-decoder structure, yet suggest architectural changes.

embedding vectors within the feature pyramid increases the further we move down the pyramid. This use of downsampling to capture different length temporal patterns is comparable to how inertial models concatenate convolutional layers without padding to increase the receptive field of each convolutional kernel. However, both types of models differ in the types of patterns learned. The feature pyramid of the TAL models enables them to learn cross-window, temporal patterns of arbitrary length, while the convolutional parts of the inertial models enable them to learn in-window temporal patterns of arbitrary length.

Due to the TAL community's training objective differing from that of the inertial community, TAL models are trained not only on a classification loss, but also a regression loss, which optimizes each timestamp's corresponding activity onset and offset. The use of self-attention based layers has shown to improve results in both communities, yet variations of the ActionFormer show that these layers are not necessarily needed. More specifically, the use of fully-convolutional SGP layers in the TriDet model show that, rather than the type of technique being important, focus needs to be put specifically on learning both local and global temporal information for each timestamp.

4 METHODOLOGY

4.1 Datasets

We evaluate each algorithm featured in this benchmark analysis using 6 popular HAR datasets, namely the Opportunity [45], SBHAR [43], Wetlab [46], WEAR [7], Hang-Time [21] and RWHAR dataset [50]. The datasets, all covering different application and recording scenarios, provide us with a challenging set of prediction problems to properly assess the strengths and weaknesses of each model. Table 1 summarizes the key characteristics of each dataset. In addition to vital information such as participant count, activity count, sampling rate and sensor axes count, the table also includes details on the scenario and type of activities found in each dataset. We classify activities into four types: (1) periodic activities, characterized by recurring periodic patterns; (2) non-periodical activities consisting of non-occurring, non-periodical patterns; and (3) complex activities, defined by an arbitrary sequence of non-periodic and periodic activities. Lastly, following the works of Alwassel et al. [2], we provide average count of segments across all subjects, categorizing each segment within each dataset into five bins: XS: (0 seconds, 3 seconds], S: (3 seconds, 6 seconds], M: (6 seconds, 12 seconds], L: (12 seconds, 18 seconds], and XL: more than 18 seconds.

Table 1. Key characteristics of the datasets used in this benchmark analysis. The table provides: participant count (#Sbjs), activity count (#Cls), sampling rate (SR), sensor axes count (#Axes), average number of segments per participant based on absolute length of the segment (#Segments), overall scenario (e.g. activities of daily living (ADL)) and type of activities found in the dataset.

Dataset	#Sbjs	#Cls	SR	#Axes	#Segments [2]					Scenario	Type of Activities
					XS	S	M	L	XL		
Opportunity [45]	4	17(+1)	30	113	403 ± 63	185 ± 48	48 ± 10	3 ± 2	0 ± 0	ADL	non-periodic, complex
SBHAR [43]	30	12(+1)	50	3	3 ± 2	9 ± 2	6 ± 4	10 ± 4	13 ± 3	Locomotion	(non-)periodic
Wetlab [46]	22	8(+1)	50	3	7 ± 5	7 ± 3	7 ± 2	5 ± 2	13 ± 2	Laboratory	complex
WEAR [7]	18	18(+1)	50	12	1 ± 1	1 ± 1	1 ± 1	1 ± 1	30 ± 9	Sports	periodic
Hang-Time [21]	24	5(+1)	50	3	154 ± 53	22 ± 7	10 ± 7	3 ± 3	3 ± 2	Sports	(non-)periodic, complex
RWHAR [50]	15	8	21	50	0 ± 0	0 ± 0	0 ± 0	0 ± 0	9 ± 1	Locomotion	periodic

4.2 Training Pipeline

Prediction Scenarios. All experiments were initiated employing a Leave-One-Subject-Out cross-validation. This type of validation involves iteratively training on all but one participant's data and using the hold-out participant's data during validation until all participants have been evaluated, ensuring that each network architecture is assessed based on its capabilities to generalize across unseen participants. All datasets were windowed using a sliding window of one second with a 50% overlap across windows. In order to allow for fair comparison, our benchmark analysis assesses the TAL models employing two prediction scenarios, with the differences in their predicted output between inertial-based and TAL models. Being the scenario TAL models were intended to be used for, within the first prediction scenario (*Offline Activity Recognition*) the TAL models are tasked to predict each participant's data as a whole, i.e. one batch representing the full data stream available of one participant. Nevertheless, to also assess the models' capabilities to provide window-level predictions, within the second prediction scenario (*Online Activity Recognition*) we artificially chunk participants' timelines making the models predict the timelines in chunks (e.g., 30 seconds). As inertial-based models unlike TAL models are designed to make window-based predictions, the second prediction scenario will assess the TAL models online capabilities while also benchmarking them with removed temporal context. Note that in case of the *Offline Activity Recognition* we test the TAL models using both a single-stage as well as two-stage training (as described in Chapter 3).

Hyperparameters. For all inertial architectures [1, 6, 38, 70] we employed a similar optimization as proposed with the release of the shallow DeepConvLSTM, namely an Adam optimizer paired with a learning rate of $1e^{-4}$, a weight decay of $1e^{-6}$, and Glorot weight initialization [16]. To allow each model to converge more properly, we increase the number of epochs to 100 and employ a step-wise learning-rate schedule that multiplies the learning rate by a factor of 0.9 after every 10 epochs. For all architectures, we fixed the hidden dimension of the recurrent layers to employ 128 units and scaled kernel sizes of the convolutional filters according to the relative difference in sampling rate among the different input datasets. In line with how the Attend-and-Discriminate architecture was first introduced, we optimized said architecture using center-loss [57] as opposed to a weighted cross-entropy loss, which was used during training of all other inertial architectures. Lastly, as proposed by the authors, we do not shuffle batches during the training of the shallow DeepConvLSTM.

The three TAL architectures [47, 52, 66], we chose to employ hyperparameters reported by the authors that produced best results on the EPIC-Kitchens dataset [14]. The hyperparamenters, though optimized for a different modality than inertial data (egocentric videos), have shown to produce competitive results on the WEAR dataset [7] and are thus considered a good starting point for evaluating the applicability of the three architectures on other HAR datasets. Nevertheless, given the small size of the tested inertial datasets compared to datasets used by the TAL community, we chose to increase the amount of epochs to 100 throughout all TAL-based experiments.

As TAL models are designed to predict both a regression (boundary prediction) and classification target (segment label), loss calculation of the ActionFormer, TemporalMaxer and TriDet models are performed as a weighted combination of a regression loss (IoU loss [44]) and classification loss (focal loss [26]).

Postprocessing. To compare predictions of both TAL and inertial-based models, segments (TAL) and windowed-predictions (inertial-based) first needed to be translated back to sample-wise predictions. In case of the segments we do so by translating each segment "overwrite" the prediction timeline with its associated label. By starting with the segment associated with the lowest prediction confidence, segments with a high prediction confidence are preferred in case of overlapping segments. In case of the inertial-based architectures, we start by iterating over the windowed predictions in order of occurrence in the timeline having them determine the samples they are associated with. To deal with the overlap amongst windows, preceding windows are allowed to "overwrite" the label of samples of previous windows which they are overlapping with.

As inertial models are tasked to predict on a per-window basis and, the architectures suffer from frequently occurring activity switches ultimately leading to fragmented segments which significantly lower mAP scores being produced as opposed to the TAL models. Therefore, to remove only short lasting switches, predictions of inertial-based architectures mentioned in this paper were smoothed using majority-vote filters. The exact size of the majority filter was chosen dataset-specific, determined via trying out a selection of filters between 2.5 and 40 seconds. Specifically, the filters were chosen as: 2.5 seconds (Opportunity), 20 seconds (Wetlab), 5 seconds (SBHAR), 15 seconds (WEAR), 5 seconds (Hang-Time) and 40 seconds (RWHAR). Similar to the inertial models, the TAL architectures are tasked to predict class probabilities and segment boundaries of each windowed timestamp. Consequently, without applying any confidence threshold, all predicted activity segments are considered during creation of the prediction timeline causing accuracy of the NULL-class to be significantly low. Therefore, to alleviate this, we apply an optimized confidence threshold on predicted segments of all TAL models. Similarly to the majority filter, the score threshold for each architecture was chosen dataset-specific, determined via trying out a selection of thresholds between 0.05 and 0.5 seconds. This eliminates low scoring segments and substantially lowers the confusion of the NULL-class with the other activities. More details on the effect of the majority filter as well as score thresholding can be found in the supplementary material.

5 RESULTS

As part of our experimental evaluation, we provide traditional classification metrics (precision, recall and F1-score), misalignment measures as defined by Ward et al. [56] and mAP averaged across tIoU thresholds 0.3, 0.4, 0.5, 0.6 and 0.7. All results are the unweighted average across all subjects along the LOSO validation. Experiments were repeated three times employing three different random seeds (1, 2 and 3). Classification metrics are calculated on a per-sample basis as the segmented output of the TAL models and windowed output of the inertial-models need to be translated to a common time granularity. To ensure readability of this work, visualization of the per-class analysis will only include confusion matrices of the SBHAR and RWHAR datasets, as we deemed these two datasets to be the most representative in illustrating the strengths and weaknesses of the TAL architectures when applied to inertial data. Please note that the confusion matrices of the other datasets can be found in the supplementary material. Furthermore, all created plots part of a performed DETAD analysis [2] on each dataset can be found in our repository.

5.1 Offline Activity Recognition

Table 2 provides average results of the seven tested architectures across each dataset in an offline prediction setting. One can see that the TAL architectures outperform the inertial architectures across all datasets regarding average mAP. This shows that by being trained to specifically optimize activity boundaries, the different prediction target has resulted in overall more coherent segments which overlap to a larger degree with the ground truth

segments. Even though prediction results of the inertial architectures were further smoothed by a majority filter, average mAP is, except for the WEAR dataset, more than halved when compared to that of the TAL architectures. Regarding traditional classification metrics the TAL architectures are able to outperform inertial architectures for four out of six datasets with only the RWHAR [50] and WEAR dataset [7] being the exception. These improvements range between 5% for the Opportunity and Wetlab dataset, 10% for the Hang-Time and even as much as 25% in F1-score for the SBHAR dataset. Calculated misalignment ratios show that both inertial and TAL architectures have a similar distribution of errors, with architectures producing better overall classification results also showing overall lower misalignment measures. Only for the RWHAR dataset one can see that TAL architectures show a significantly higher Overfill-Ratio. This might be due to the RWHAR dataset not featuring a NULL-class, which introduces an uncommon prediction scenario for the TAL architectures. Nevertheless, the performed DETAD analysis [2], which further differentiates amongst the segment-based errors, reveals that inertial architectures suffer more severely from background errors, i.e. confusing activities with the NULL-class. While this effect is decreased for the shallow DeepConvLSTM, TAL architectures show to be able to more reliably differentiate between activities and the NULL-class, and thus more reliably localize activities within the untrimmed sequences.

With the improvements on the SBHAR being the largest across all datasets, one can see on a per class level (see Figure 5) that this increase can be attributed to improved performance on transitional, non-periodic classes like *sit-to-stand*. These activities are mostly recognisable by their context and surrounding activities and are thus particularly challenging to predict for models that do not rely on temporal dependencies spanning multiple seconds. By applying the TAL architectures in an offline manner, the architectures are able to leverage both local and global context across the whole timeline and are thus even able to recognize these short-lasting activities. Since the DeepConvLSTM, Attend-and-Discriminate and TinyHAR models are being trained on shuffled training data and have recurrent parts applied on within-window sequences, the three architectures cannot learn to predict these activities based on surrounding context. Among inertial models, only the shallow DeepConvLSTM is able to more reliably predict these context-based classes as it is trained on unshuffeled data and applies a dimensional flip when training the LSTM. Across all datasets part of this benchmark analysis, the RWHAR dataset yields the least performant results for the TAL architectures. We accredit this primarily due to the absence of a NULL-class in the dataset, which introduces a uncommon prediction scenario for the models. Furthermore, the RWHAR consists of the least amount of segments per subject, limiting the amount of training segments which can be used to optimize the TAL models. Results on the RWHAR dataset are nevertheless surprising as the TAL models show confusion among classes which they do not struggle to predict in other datasets (e.g., *lying* in the SBHAR dataset) as well as classes that are not similar to each other (e.g., *lying* and *jumping*). The obtained results raise the question of whether TAL models are primarily suited for being applied on untrimmed sequences, which (1) include breaks and/ or (2) provide a larger amount of segments than the RWHAR dataset.

Nevertheless, apart from the RWHAR dataset, the TAL models deliver the most consistent results across all classes. Even though, in most cases, the individual per-class accuracies are not the highest when compared to results obtained using the inertial architectures, the inertial architectures are frequently not able to predict all classes reliably, with at least one class showing low prediction accuracy. This is especially evident when examining the results obtained on the Hang-Time and Opportunity datasets, where TAL models are capable of correctly predicting challenging non-periodic activities such as *rebounds*, *passes*, *opening door* and *closing door*. Furthermore, as also seen in the DETAD analysis, TAL models are overall more capable of differentiating activities from the NULL-class, showing the highest NULL-class accuracy across all datasets. To summarize, the TAL architectures are more reliable in recognizing any kind of actions within an untrimmed sequence, and are less prone to predict fragmented prediction streams or non-existent breaks.

Table 2. *Offline Activity Recognition:* Average LOSO cross-validation results obtained on six inertial HAR benchmark datasets [7, 21, 43, 45, 46, 50] for four inertial [1, 6, 38, 70] and three TAL architectures [47, 52, 66]. The table provides per-sample classification metrics, i.e. Precision (P), Recall (R), F1-Score (F1), misalignment ratios [56] and average mAP applied at different tIoU thresholds (0.3:0.1:07). All experiments employed a sliding window of one second with a 50% overlap. Results are averaged across three runs employing different random seeds. The TAL architectures are able to outperform the inertial architectures regarding average mAP on all HAR datasets and result in the highest classification metrics on four out of the six datasets. Best results per dataset are in **bold**.

	Model	P (↑)	R (↑)	F1 (↑)	UR (↓)	OR (↓)	DR (↓)	IR (↓)	FR (↓)	MR (↓)	mAP (↑)
Opportunity	DeepConvLSTM	50.22	33.88	34.41	0.30	17.29	0.78	**31.26**	**0.01**	0.23	13.97
	Shallow D.	42.08	27.18	26.46	0.24	14.28	0.97	35.06	**0.01**	**0.15**	10.61
	A-and-D	35.25	48.55	36.35	0.32	15.28	0.45	52.55	0.02	0.35	13.75
	TinyHAR	48.09	54.27	47.09	0.34	19.99	**0.39**	34.01	0.02	0.46	19.78
	ActionFormer	**54.63**	**58.78**	**51.93**	**0.19**	13.34	0.41	33.82	**0.01**	0.42	**51.24**
	TemporalMaxer	44.44	55.63	44.74	0.21	14.13	0.40	43.78	**0.01**	0.55	46.31
	TriDet	48.72	57.69	48.79	0.23	**13.20**	**0.39**	40.07	**0.01**	0.58	49.70
SBHAR	DeepConvLSTM	67.54	63.72	62.31	0.41	7.19	0.59	21.44	0.07	0.10	49.60
	Shallow D.	72.98	75.41	71.13	0.60	10.19	0.46	14.23	0.02	**0.09**	65.15
	A-and-D	68.64	71.07	66.49	**0.31**	9.47	0.45	21.71	0.05	0.11	55.79
	TinyHAR	58.91	63.70	56.29	**0.31**	8.16	0.54	31.67	0.05	0.13	45.38
	ActionFormer	87.02	84.37	84.43	0.36	**5.41**	0.16	5.09	**0.00**	0.20	**95.46**
	TemporalMaxer	86.52	83.37	83.66	0.41	6.02	0.19	4.41	**0.00**	0.24	94.39
	TriDet	**88.95**	**86.15**	**86.45**	0.38	**5.41**	**0.12**	**3.95**	0.01	0.29	94.75
Wetlab	DeepConvLSTM	38.65	47.34	37.87	**0.51**	**7.13**	0.69	48.53	0.21	0.64	11.88
	Shallow D.	39.01	35.42	34.42	**0.51**	9.52	1.57	**34.51**	**0.06**	**0.43**	15.40
	A-and-D	37.75	**55.71**	37.49	0.56	9.69	**0.63**	57.60	0.16	0.57	12.27
	TinyHAR	34.31	50.84	31.48	0.61	8.41	1.00	61.26	0.11	0.59	10.05
	ActionFormer	40.71	49.25	40.71	0.56	9.41	0.79	52.44	0.07	0.79	33.53
	TemporalMaxer	**50.43**	36.65	37.09	0.59	9.37	0.83	53.59	0.10	0.61	**35.72**
	TriDet	44.13	49.15	**42.85**	0.58	8.85	0.75	48.63	0.10	0.84	34.05
WEAR	DeepConvLSTM	80.68	76.25	75.78	0.28	**2.35**	0.52	6.68	0.11	**0.32**	61.03
	Shallow D.	80.78	78.91	77.71	0.27	3.23	0.50	**5.21**	0.04	0.43	67.89
	A-and-D	**82.34**	83.29	**80.61**	**0.20**	4.03	0.33	7.18	0.09	0.52	64.78
	TinyHAR	81.87	**84.23**	80.56	0.21	4.77	**0.27**	9.54	0.10	0.55	63.33
	ActionFormer	71.88	76.70	72.43	0.22	6.48	0.65	7.12	**0.01**	2.06	73.80
	TemporalMaxer	69.54	72.80	69.52	0.23	6.87	0.83	5.50	**0.01**	1.50	69.18
	TriDet	73.57	77.54	73.18	0.28	4.79	0.64	6.21	0.03	1.64	**77.12**
Hang-Time	DeepConvLSTM	44.13	33.95	35.25	**0.28**	10.60	0.88	**20.16**	0.27	1.46	5.44
	Shallow D.	37.97	38.19	36.85	0.35	14.00	1.07	36.38	**0.21**	2.33	5.00
	A-and-D	40.54	43.32	40.39	0.35	14.14	0.72	34.81	0.30	1.44	6.73
	TinyHAR	37.09	41.13	36.89	0.41	11.59	0.75	42.90	0.43	1.26	4.73
	ActionFormer	49.19	**57.57**	**51.23**	0.62	11.63	0.51	47.65	0.48	0.64	**29.26**
	TemporalMaxer	45.01	54.56	47.17	0.71	10.97	**0.45**	52.71	1.13	0.65	27.86
	TriDet	**49.59**	55.14	50.67	0.73	**9.85**	0.52	48.55	0.69	**0.62**	29.24
RWHAR	DeepConvLSTM	79.05	81.93	77.56	0.65	2.05	1.09	13.63	1.07	**0.00**	0.11
	Shallow D.	**88.59**	**89.01**	**86.85**	**0.31**	**1.14**	**0.38**	**6.93**	0.98	**0.00**	0.00
	A-and-D	79.46	82.68	78.09	0.66	2.31	1.17	11.28	0.84	**0.00**	0.04
	TinyHAR	83.59	86.25	82.62	0.53	**1.14**	0.76	11.65	0.89	**0.00**	0.02
	ActionFormer	63.76	67.64	61.24	2.48	11.12	2.06	11.23	**0.09**	**0.00**	65.40
	TemporalMaxer	63.20	67.60	60.59	2.59	11.95	1.81	13.96	0.36	0.05	50.60
	TriDet	69.27	73.04	67.86	1.48	6.88	2.03	10.24	0.23	**0.00**	**69.98**

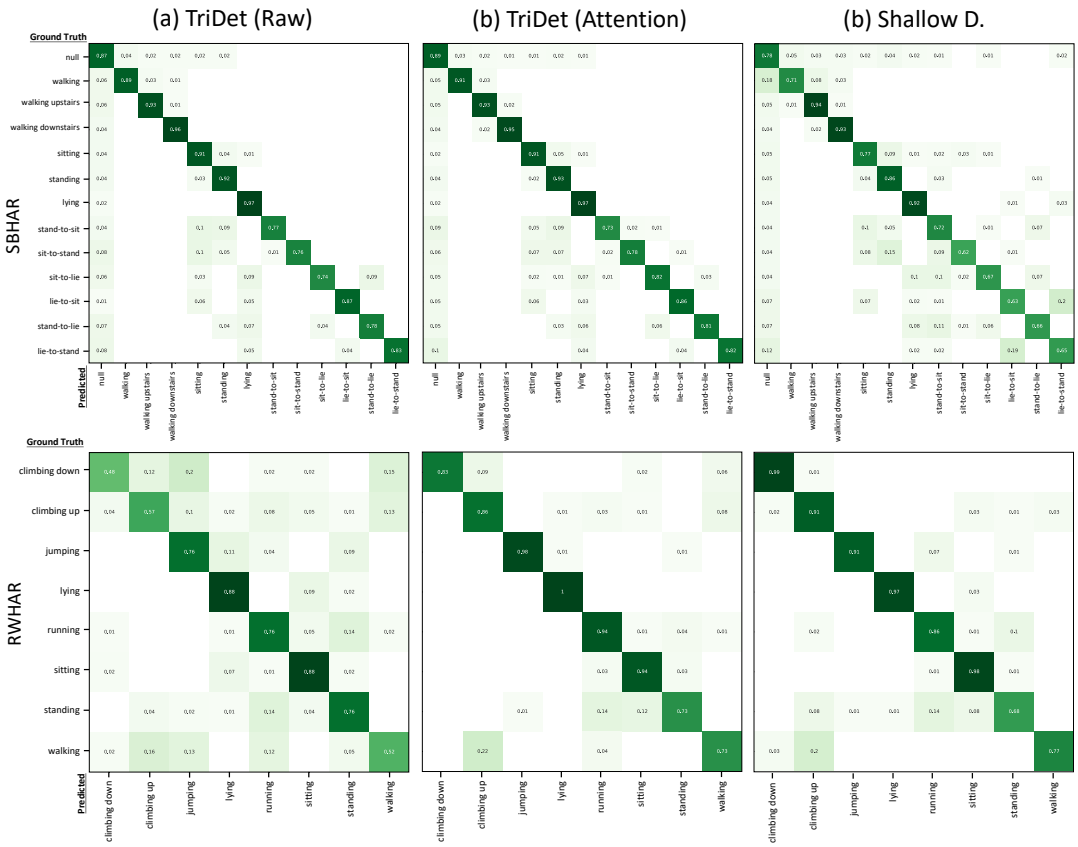

Fig. 5. *Offline Activity Recognition:* Confusion matrices of the (a) the best TAL architecture (TriDet) [47] and (b) inertial model (shallow DeepConvLSTM) being applied on the SBHAR [43] (top) and RWHAR dataset [50] (bottom) with a one second sliding window and 50% overlap. Note that confusions which are 0 are omitted.

Single- vs. Two-Stage TAL Training. As described in Section 3, the TAL models described in this paper were intended to be applied using feature embeddings describing each sliding window, rather than raw data. Nevertheless, our initial results show that TAL models are indeed capable of being applied to raw inertial data. Figure 6 presents results of our implemented two-stage TAL training, which extends the training process as described in the previous chapters with a prepended feature extraction using inertial models. In total, we assess two feature embeddings: LSTM-based features extracted from a DeepConvLSTM [38] and attention-based features from a TinyHAR architecture [70]. Though the two-stage training using LSTM-based features only yields better F1-scores for the Wetlab, WEAR, and RWHAR datasets, the training using attention-based features improves results across all datasets across all TAL models. Given the significant improvements on the RWHAR and WEAR datasets, we assume that the feature extraction via the prepended inertial network helps increase the discriminability of the window-level features, yet at the cost of making it harder for the TAL models to learn cross-window temporal relations, as evident by the decrease in mAP scores across (almost) all datasets.

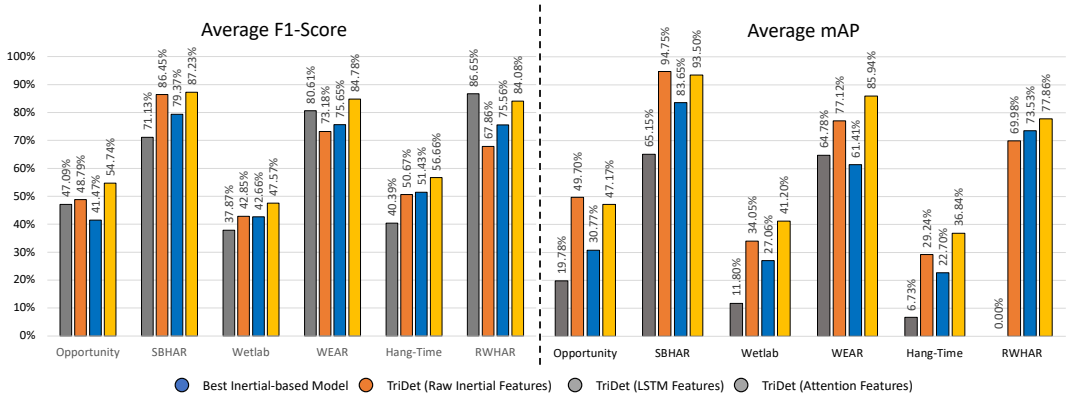

Fig. 6. *Offline Activity Recognition*: Average LOSO results of the two-stage TAL training of the TriDet model. We compare the LSTM and attention feature-based two-stage TAL training with the best inertial-based architecture and the single-stage TAL training on raw inertial data of all six benchmark datasets. One can see a clear improvement when using attention-based features across all datasets.

Table 3. Average LOSO results obtained using the evaluated TAL architectures [47, 52, 66] employing different degrees of chunking during validation. During validation each subject is split into equal-sized chunks with each chunk being individually predicted by the trained TAL models. The table provides per-sample F1-Score (F1), average mAP applied at different tIoU thresholds (0.3:0.1:07) calculated on each chunk individually (c-mAP) and on the reconstructed validation stream (mAP). Results are averaged across three runs employing different random seeds. Results which are underlined are not score thresholded.

		1 sec.			5 sec.			30 sec.			60 sec.			Unchunked	
	Model	F1	c-mAP	mAP	F1	c-mAP	mAP	F1	c-mAP	mAP	F1	c-mAP	mAP	F1	mAP
Opp.	AF	2.63	2.45	0.36	37.73	18.10	25.51	49.96	41.74	45.84	50.50	45.45	47.84	51.93	51.24
	TM	2.22	2.09	0.31	38.00	19.08	25.99	45.32	39.24	42.83	45.27	42.15	44.22	44.74	46.31
	TD	1.48	2.26	0.37	9.96	10.51	13.6	12.15	12.73	13.37	1.99	4.15	4.27	48.79	49.70
SBHAR	AF	3.44	7.56	0.12	34.49	26.20	15.09	78.88	74.28	79.52	81.57	82.45	86.39	84.43	95.46
	TM	3.30	5.39	0.05	33.49	47.11	23.87	78.08	73.95	77.64	81.04	82.41	85.19	83.66	94.39
	TD	2.13	10.18	0.13	17.69	29.31	8.05	31.70	52.01	49.38	44.76	72.94	72.64	86.45	94.75
Wetlab	AF	5.13	2.37	0.02	14.55	3.12	0.77	35.35	25.65	20.15	35.98	30.46	28.06	40.71	33.53
	TM	4.82	2.23	0.03	11.33	15.81	1.51	34.37	25.87	20.16	35.35	30.24	26.90	37.09	35.72
	TD	1.77	4.08	0.03	2.20	4.40	0.63	4.03	6.10	7.38	7.78	10.98	9.11	42.85	34.05
WEAR	AF	5.70	3.47	0.00	15.54	10.44	0.06	53.57	40.02	18.72	69.21	56.89	52.11	72.43	73.80
	TM	5.30	2.65	0.00	38.37	53.39	0.19	58.29	45.09	18.94	67.91	56.60	47.89	69.52	69.18
	TD	1.25	5.41	0.00	3.72	15.12	0.09	34.67	49.73	21.57	43.96	49.33	43.79	73.18	77.12
Hang-T.	AF	6.41	3.19	0.70	42.41	17.18	14.67	49.32	26.47	26.82	50.69	27.64	27.65	51.23	29.26
	TM	6.39	2.63	0.50	41.13	17.54	15.17	48.15	25.72	25.38	48.67	26.63	26.33	47.17	27.86
	TD	19.52	0.00	3.56	19.33	3.67	3.48	26.02	12.42	13.01	35.72	18.80	19.15	50.67	29.24
RWHAR	AF	8.47	28.67	0.00	17.00	51.96	0.00	50.15	76.19	1.37	62.40	77.58	7.91	61.24	65.40
	TM	8.06	17.13	0.00	31.35	48.89	0.00	52.71	60.47	1.31	58.47	61.86	7.61	60.59	50.60
	TD	5.88	25.68	0.00	9.07	40.11	0.00	39.78	47.78	0.94	48.65	57.51	7.39	67.86	69.98

5.2 Online Activity Recognition

A major difference in how TAL models are intended to be used compared to inertial models is that they can only provide an offline prediction of previously recorded timeseries data. In previous experiments, we demonstrated that TAL models are capable of analyzing a participant's data stream as a whole and can even deal with arbitrary lengths of timeseries data and activity durations. However, to investigate how capable TAL models are at analyzing only small snippets of incoming data streams, we modified the previously employed Leave-One-Subject-Out validation loop so that TAL models are tasked to predict each validation subjects' data in a chunked manner. Specifically, given the data of a validation participant, we divided the data into equal-sized portions (e.g. 5 seconds worth of data) and had the TAL models, trained using unchunked data, predict each chunk individually. Table 3 compares the initial unchunked results of the tested TAL models (ActionFormer, TemporalMaxer and TriDet) on the six inertial benchmark datasets with results obtained when fragmenting each validation split into 1, 5, 30, or 60-second chunks. Across all datasets one can witness that with smaller-sized chunks classification and mAP results worsen across all algorithms and datasets. Overall, we can see that the smaller the chunk, the lower the overall confidence of the TAL models regarding each predicted segment. The lower confidence further leds to almost all segment predictions being removed even with a low score threshold such as 0.05, resulting in us not applying score thresholding in those cases, i.e., using all segments predicted by the respective TAL model. We assume that this drop in prediction confidence is likely caused by artificially splitting long segments of activities, contradicting what the models have seen during training, as datasets such as the Opportunity and Hang-Time dataset, which contain mostly short-lasting segments, are less affected by the smaller chunks as e.g. the WEAR dataset which has almost exclusively segments lasting longer than 30 seconds. While the TriDet model is not capable of dealing with the chunked validation, most likely due to it using a different style of regression head, the ActionFormer and the TemporalMaxer are capable of maintaining predictive performance even for small-sized chunks of 5 seconds across almost all datasets. As expected, mAP scores are more impacted by the chunked prediction as chances are increased that long lasting segments are split due to the predicted segments not spanning across chunks. Nevertheless, we expect this effect to dampen if one would use an additional majority filtering as used in our inertial-based experiments, as this would eliminate potential intermediate activity switches.

Table 4 provides a comparison in terms of learnable parameters, size on disk, average batch training and inference times of the inertial-based models compared with the TAL models. One can see that the TAL models are significantly larger in size and number of learnable parameters. Though all architectures part of our analysis are capable of predicting a window of one second as well as the complete prediction stream of a participant within less than half a second, inertial-based models are faster than TAL models in both inference scenarios being on average faster by around a magnitude of 10. Taking into consideration that all TAL models fail to predict 1 second chunks and are significantly larger than inertial-based models, it becomes aparent, that TAL models are not suited for inference on edge-devices. It becomes apparent that with the models being trained to predict activity occurrences as a whole, the models require enough surrounding context in order to spot activities and correctly identify what they have seen during training. Nevertheless, as soon as the TAL models are given enough surrounding context (e.g., a prior context of 60 seconds), the models are capable to spot activities within said activity stream in a reasonable run time.

6 DISCUSSION & CONCLUSION

This article demonstrated the applicability of vision-based, single-stage Temporal Action Localization for inertial-based Human Activity Recognition. We showed that three state-of-the-art TAL models [47, 52, 66] can be applied in a plain fashion to raw inertial data and achieve competitive results on six popular inertial HAR datasets [7, 21, 43, 45, 46, 50], outperforming in most cases popular models from the inertial community in an offline prediction scenario by a significant margin. Using a combination of an inertial network as a feature extractor and

Table 4. Comparison in terms of learnable parameters (in million (M)), size on disk (in MB), average batch training and inference times (in milliseconds) of the seven benchmark algorithms. Training and inference speeds are the average across the first 5 epochs of the first LOSO validation split of the algorithms being applied on the WEAR dataset. Benchmarking was performed on a single NVIDIA GeForce RTX 4090 with an AMD Ryzen 7800 X3D. We assess both inference speeds of the architectures being tasked to predict one 1-second window as well as one complete subject. ** 1 batch equals 100 1-second windows ** 1 batch equals 1 participant.

	Model	Total Params	Size	Avg. Time per Batch		
				Train	Test (1 sec.)	Test (1 participant)
Baseline	DeepConvLSTM	0.71 M	2.69MB	4ms*	2ms	27ms
	Shallow D.	0.57 M	2.19MB	3ms*	2ms	29ms
	A-and-D	0.57 M	2.19MB	31ms*	10ms	41ms
	TinyHAR	0.04 M	0.14MB	7ms*	3ms	28ms
	ActionFormer	27.02 M	103.37MB	131ms**	39ms	346ms
	TemporalMaxer	4.89 M	18.94MB	69ms**	25ms	380ms
	TriDet	13.75 M	52.73MB	127ms**	37ms	347ms

a TAL model showed to significantly enhance classification results for all TAL models – especially on datasets where classic inertial models had previously outperformed TAL models. Our two-stage experiments further suggest potential improvements that could be achieved by investigating improved methods for feature extraction using TAL models and inertial data, e.g. via a fully-differentiable combined version of both type of architectures (see Figure 7).

A previously unexplored metric in inertial-based HAR, mean Average Precision (mAP), reveals that TAL models predict less fragmented timelines compared to inertial models and overall achieve larger degrees of overlap with ground truth segments. Furthermore, TAL models show to predict even non-periodic and complex activities more reliably than inertial architectures, providing consistent results across all types of classes across all datasets. Being one of the key challenges in HAR [8], the TAL architectures are further shown to be less affected by the unbalanced nature of HAR datasets due to a large NULL-class. Across the five benchmark datasets which offered a NULL-class, the TAL architectures showed to deliver the highest NULL-class accuracy. Additional experiments, which involved applying the TAL models on artificially created chunked sequences of data, showed that TAL models, though intended for the offline analysis of prerecorded activity timelines, have the capability of being applied in a near-online fashion (see Figure 7). Although the TAL models highlighted in this paper overall size and complexity would not allow them to be run on edge devices (yet), their reasonable performance and inference time on large enough chunks suggest that they could function e.g. as a server-side prediction model as seen in [4], which analyzes chunks of data in regular intervals.

The research community for inertial-based activity recognition has contributed methods to better model temporal relationships in the past years, yet most such architectures are limited to learning context within a fixed-sized sliding window. To this date, the length of the sliding window remains a crucial parameter in inertial-based HAR, which may result in a significant performance drops in recognition performance when set incorrectly. Specifically, if dealing with both long and short lasting activities, a small sliding window might end up being too small to fully capture an activity, while a too large window size might cause sliding windows containing mixed activity types. With studies such as Pellatt and Roggen [39] and Guan and Plötz [19] contributing strategies to improve training, the TAL community offers an interesting new perspective to inertial-based HAR – alleviating the need for fixed size windows and making algorithms capable of dealing with activities of arbitrary length. Although models from these two independent communities share similarities, the TAL community offers many unexplored design choices and training concepts for a multitude of application scenarios, which we argue should be considered for investigation by the inertial-based HAR community.

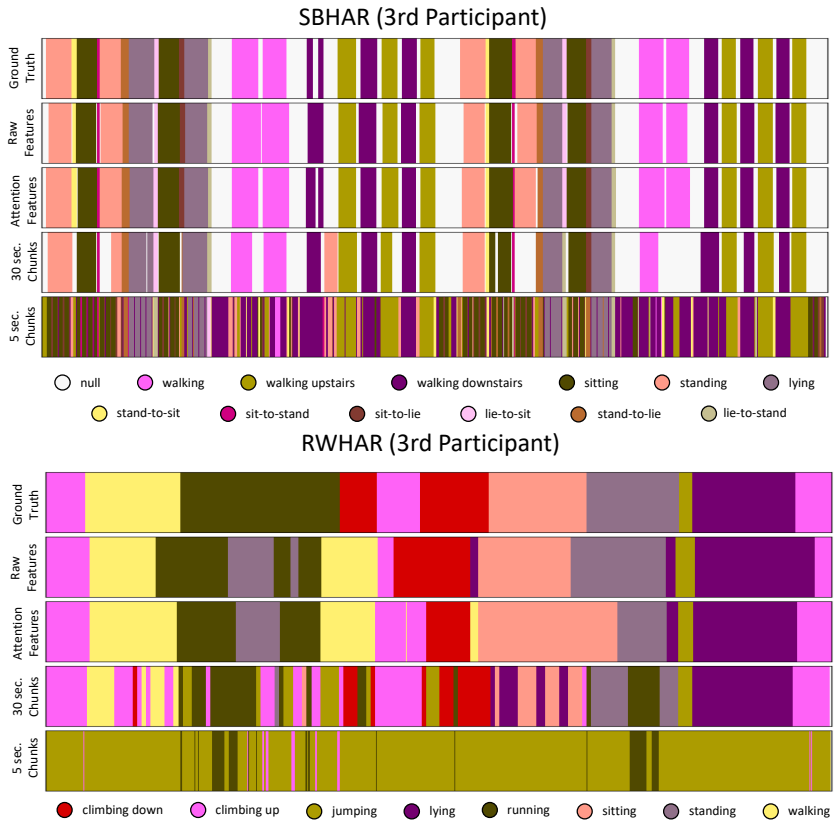

Fig. 7. Color-coded visualization of the effect of the *Offline* and *Online Activity Recognition* experiments using the ActionFormer model [66]. Results are compared with the ground truth of a sample participant as well as offline single-stage training for both the RWHAR [50] and SBHAR [43] dataset. One can see that training the TAL models using attention based features clearly improves results with an overall better recognition of all activities. Contrarily, once predicting a chunked output, performance can be somewhat maintained for larger chunks, yet drops near zero once applied on only 5-second long windows.

ACKNOWLEDGMENTS

We gratefully acknowledge the DFG Project WASEDO (grant number 506589320) and the University of Siegen's OMNI cluster.

REFERENCES

[1] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C. Ranasinghe. 2021. Attend and Discriminate: Beyond the State-Of-The-Art for Human Activity Recognition Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–22. https://doi.org/10.1145/3448083

[2] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. 2018. Diagnosing Error in Temporal Action Detectors. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-030-01219-9_16

[3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. 2020. Boundary Content Graph Neural Network for Temporal Action Proposal Generation. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-030-58604-1_8

[4] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R. Schmidtke, and Michael Beigl. 2010. ActiServ: Activity Recognition Service for mobile phones. In *IEEE International Symposium on Wearable Computers*. https://doi.org/10.1109/ISWC.2010.5665868

[5] Ulf Blanke and Bernt Schiele. 2009. Daily Routine Recognition through Activity Spotting. In *Location and Context Awareness*. Vol. 5561. https://doi.org/10.1007/978-3-642-01721-6_12

[6] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. Improving Deep Learning for HAR With Shallow Lstms. In *ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3460421.3480419

[7] Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. 2023. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. *CoRR* abs/2304.05088 (2023). https://arxiv.org/abs/2304.05088

[8] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *Comput. Surveys* 46, 3 (2014), 1–33. https://doi.org/10.1145/2499621

[9] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2017.502

[10] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. 2022. DCAN: Improving Temporal Action Detection via Dual Context Aggregation. In *AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v36i1.19900

[11] Ling Chen, Rong Hu, Menghan Wu, and Xin Zhou. 2023. HMGAN: A Hierarchical Multi-Modal Generative Adversarial Network Model for Wearable Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (2023). https://doi.org/10.1145/3610909

[12] Ling Chen, Yi Zhang, Shenghuan Miao, Sirou Zhu, Rong Hu, Liangying Peng, and Mingqi Lv. 2023. SALIENCE: An Unsupervised User Adaptation Model for Multiple Wearable Sensors Based Human Activity Recognition. *IEEE Transactions on Mobile Computing* 22, 9 (2023). https://doi.org/10.1109/TMC.2022.3171312

[13] Ling Chen, Yi Zhang, and Liangying Peng. 2020. METIER: A Deep Multi-Task Learning Based Activity and User Recognition Model Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020). https://doi.org/10.1145/3381012

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2022. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision* (2022). https://doi.org/10.1007/s11263-021-01531-2

[15] Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer. 2016. Monitoring Activities of Daily Living in Smart Homes: Understanding human behavior. *IEEE Signal Processing Magazine* 33, 2 (2016), 81–94. https://doi.org/10.1109/MSP.2015.2503881

[16] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *International Conference on Artificial Intelligence and Statistics*. http://proceedings.mlr.press/v9/glorot10a

[17] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. 2020. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *IEEE International Conference on Multimedia and Expo*. https://doi.org/10.1109/ICME46284.2020.9102850

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, and Xingyu Liu. 2022. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR52688.2022.01842

[19] Yu Guan and Thomas Plötz. 2017. Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–28. https://doi.org/10.1145/3090076

[20] Fabian Caba Heilbron, Juan Carlos Niebles, Victor Escorcia, and Bernard Ghanem. 2015. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2015.7298698

[21] Alexander Hoelzemann, Julia L. Romero, Marius Bock, Kristof Van Laerhoven, and Qin Lv. 2023. Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors. *MDPI Sensors* 23, 13 (2023). https://doi.org/10.3390/s23135879

[22] Sozo Inoue, Paula Lago, Tahera Hossain, Tittaya Mairittha, and Nattaya Mairittha. 2019. Integrating Activity Recognition and Nursing Care Records: The System, Deployment, and a Verification Study. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019). https://doi.org/10.1145/3351244

[23] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS challenge: Action Recognition With a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/

[24] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and Understanding Recurrent Networks. *CoRR* abs/1506.02078 (2015). http://arxiv.org/abs/1506.02078

[25] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017). http://arxiv.org/abs/1705.06950

[26] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 2020. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper_files/paper/2020/file/f0bda020d2470f2e74990a07a607ebd9-Paper.pdf

[27] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. 2020. Fast Learning of Temporal Action Proposal via Dense Boundary Generator. In *IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1609/aaai.v34i07.6815

[28] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. 2021. Learning Salient Boundary Feature for Anchor-Free Temporal Action Localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr46437.2021.00333

[29] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *IEEE/CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2019.00399

[30] Qinying Liu and Zilei Wang. 2020. Progressive Boundary Refinement Network for Temporal Action Detection. In *AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v34i07.6829

[31] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. GiobalFusion: A Global Attentional Deep Learning Framework for Multisensor Information Fusion. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020). https://doi.org/10.1145/3380999

[32] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip H. S. Torr. 2021. Multi-Shot Temporal Event Localization: A Benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr46437.2021.01241

[33] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. 2022. End-To-End Temporal Action Detection With Transformer. *IEEE Transactions on Image Processing* 31 (2022). https://doi.org/10.1109/TIP.2022.3195321

[34] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. 2019. Gaussian Temporal Awareness Networks for Action Localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2019.00043

[35] Shenghuan Miao, Ling Chen, Rong Hu, and Yingsong Luo. 2022. Towards a Dynamic Inter-Sensor Correlations Learning Framework for Multi-Sensor-Based Wearable Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022). https://doi.org/10.1145/3550331

[36] Vishvak S. Murahari and Thomas Plötz. 2018. On Attention Models for Human Activity Recognition. In *ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3267242.3267287

[37] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. 2022. Proposal-Free Temporal Action Detection via Global Segmentation Mask Learning. In *European Conferencee on Computer Vision*. https://doi.org/10.1007/978-3-031-20062-5_37

[38] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *MDPI Sensors* 16, 1 (2016). https://doi.org/10.3390/s16010115

[39] Lloyd Pellatt and Daniel Roggen. 2020. CausalBatch: Solving Complexity/Performance Tradeoffs for Deep Convolutional and Lstm Networks for Wearable Activity Recognition. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3410530.3414365

[40] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018). https://doi.org/10.1145/3214277

[41] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. 2021. Temporal Context Aggregation Network for Temporal Action Proposal Refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr46437.2021.00055

[42] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. 2005. Activity Recognition From Accelerometer Data. In *17th Conference on Innovative Applications of Artificial Intelligence*. https://dl.acm.org/doi/abs/10.5555/1620092.1620107

[43] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171 (2016). https://doi.org/10.1016/j.neucom.2015.07.085

[44] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2019.00075

[45] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting Complex Activity Datasets in Highly Rich Networked Sensor Environments. In *IEEE Seventh International Conference on Networked Sensing Systems*. https://doi.org/10.1109/INSS.2010.5573462

[46] Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. 2015. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. https://doi.org/10.1145/2750858.2807547

[47] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. TriDet: Temporal Action Detection With Relative Boundary Modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr52729.2023.01808

[48] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. 2022. ReAct: Temporal Action Detection With Relational Queries. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-031-20080-9_7

[49] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. 2021. Class Semantics-Based Attention for Action Detection. In *IEEE/ CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv48922.2021.01348

[50] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-Body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *IEEE International Conference on Pervasive Computing and Communications*. https://doi.org/10.1109/PERCOM.2016.7456521

[51] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. 2021. Relaxed Transformer Decoders for Direct Action Proposal Generation. In *IEEE/CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv48922.2021.01327

[52] Tuan N. Tang, Kwonyoung Kim, and Kwanghoon Sohn. 2023. TemporalMaxer: Maximize Temporal Context With Only Max Pooling for Temporal Action Localization. *CoRR* abs/2303.09055 (2023). https://arxiv.org/abs/2303.09055

[53] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. 2018. Context Recognition In-The-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018). https://doi.org/10.1145/3161192

[54] Kristof Van Laerhoven, David Kilian, and Bernt Schiele. 2008. Using rhythm awareness in long-term activity recognition. In *12th IEEE International Symposium on Wearable Computers*. https://doi.org/10.1109/ISWC.2008.4911586

[55] 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*. https://arxiv.org/abs/1706.03762

[56] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. 2006. Evaluating Performance in Continuous Context Recognition Using Event-Driven Error Characterisation. In *Location- and Context-Awareness*. https://doi.org/10.1007/11752967_16

[57] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A Discriminative Feature Learning Approach for Deep Face Recognition. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-319-46478-7_31

[58] Christoph Wieland and Victor Pankratius. 2023. TinyGraphHAR: Enhancing Human Activity Recognition With Graph Neural Networks. In *IEEE World AI IoT Congress*. https://doi.org/10.1109/AIIoT58121.2023.10174597

[59] Rui Xi, Mengshu Hou, Mingsheng Fu, Hong Qu, and Daibo Liu. 2018. Deep Dilated Convolution on Multimodality Time Series for Human Activity Recognition. In *IEEE International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2018.8489540

[60] Kun Xia, Jianguang Huang, and Hanyu Wang. 2020. LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access* 8 (2020). https://doi.org/10.1109/ACCESS.2020.2982225

[61] Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. 2019. InnoHAR: A Deep Neural Network for Complex Human Activity Recognition. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2018.2890675

[62] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr42600.2020.01017

[63] Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. 2023. BasicTAD: An Astounding RGB-Only Baseline for Temporal Action Detection. *Computer Vision and Image Understanding* 232 (2023). https://doi.org/10.1016/j.cviu.2023.103692

[64] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. 2019. Graph Convolutional Networks for Temporal Action Localization. In *IEEE/CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2019.00719

[65] Bing Zhai, Yu Guan, Michael Catt, and Thomas Plötz. 2021. Ubi-SleepNet: Advanced Multimodal Fusion Techniques for Three-stage Sleep Classification Using Ubiquitous Sensing. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–33. https://doi.org/10.1145/3494961

[66] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing Moments of Actions With Transformers. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-031-19772-7_29

[67] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A Framework for Human Activity Recognition Using Imu Fusion and Convtransformer. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022). https://doi.org/10.1145/3534584

[68] Chen Zhao, Ali K. Thabet, and Bernard Ghanem. 2021. Video Self-Stitching Graph Network for Temporal Action Localization. In *IEEE/CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv48922.2021.01340

[69] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Bottom-up Temporal Action Localization With Mutual Regularization. In *European Conference on Computer Vision*. https://doi.org/10.1007/978-3-030-58598-3_32

[70] Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. 2022. TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In *ACM International Symposium on Wearable Computers*. https://doi.org/10.1145/3544794.3558467

[71] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. 2021. Enriching Local and Global Contexts for Temporal Action Localization. In *IEEE/CVF International Conference on Computer Vision*. https://doi.org/10.1109/iccv48922.2021.01326