

DeepConvContext: A Multi-Scale Approach to Timeseries Classification in Human Activity Recognition

Marius Bock
bock@iai.uni-bonn.de

University of Bonn
Lamarr Institute for ML and AI
Bonn, Germany

Michael Moeller
michael.moeller@uni-siegen.de
University of Siegen
Siegen, Germany

Juergen Gall
gall@iai.uni-bonn.de

University of Bonn
Lamarr Institute for ML and AI
Bonn, Germany

Kristof Van Laerhoven
kvl@eti.uni-siegen.de
University of Siegen
Siegen, Germany

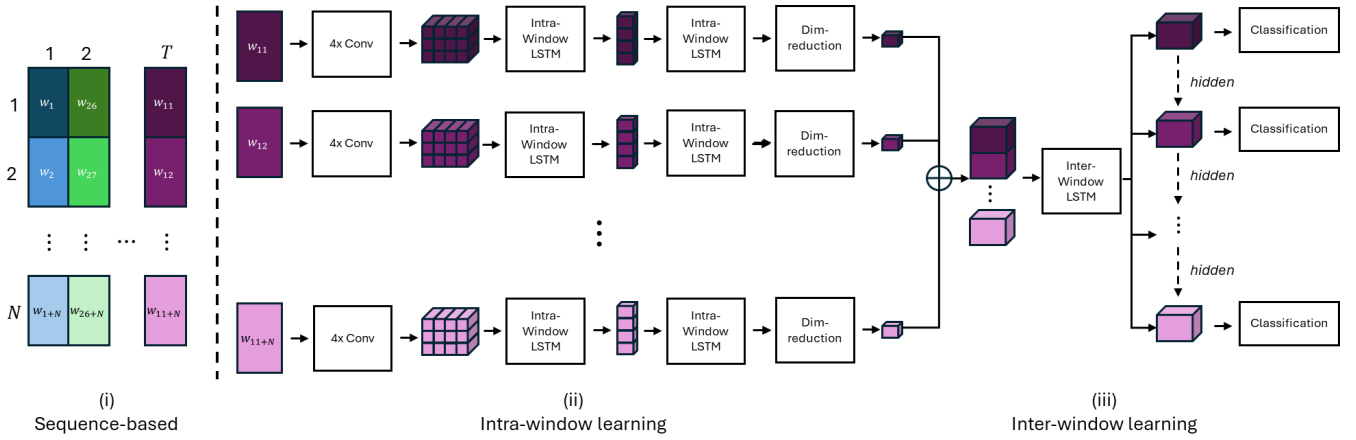


Figure 1: Overview of DeepConvContext. Our architecture uses a multi-scale approach for sliding window activity classification: Using a sequence-based shuffled batch loader during training (i), windows are individually processed by a *DeepConvLSTM*-like feature extraction (intra-window learning, ii). The resulting sequence of feature vectors of all windows (iii) are then passed to a second LSTM, to learn inter-window temporal features and are then individually classified. By preserving hidden states across windows, DeepConvContext generalises to arbitrary-length test sequences, enabling online prediction.

Abstract

Despite recognized limitations in modeling long-range temporal dependencies, Human Activity Recognition (HAR) has traditionally relied on a sliding window approach to segment labeled datasets. Deep learning models like the DeepConvLSTM typically classify each window independently, restricting learnable temporal context to within-window information and producing fragmented, temporally incoherent activity timelines. To address this constraint, we propose *DeepConvContext*, a multi-scale time series classification framework for HAR. Drawing inspiration from the vision-based Temporal Action Localization community, DeepConvContext models both intra- and inter-window temporal patterns separately by processing sequences of time-ordered windows. Across six widely-used HAR benchmarks, DeepConvContext achieves an average 5% improvement in F1-score and up to 18-point improvement in mAP over

related approaches, while achieving latency and throughput comparable to prior methods that extend temporal context through hidden state propagation across batches. Our quantitative and qualitative analysis underline the importance of inter-window learning and show how it produces more coherent activity segments even in online prediction scenarios. Code to reproduce our experiments is publicly available via www.github.com/mariusbock/deepconvcontext.

CCS Concepts

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

Keywords

Human Activity Recognition, Deep Learning, CNN-LSTMs

1 Introduction

Inertial-based Human Activity Recognition (HAR) in wearable computing research has, over the past two decades, traditionally relied



This work is licensed under a Creative Commons Attribution 4.0 International License.

on a sliding window-based classification approach [6]. Enabling near real-time prediction, this method divides continuous sensor data into overlapping windows, each of which is independently classified by a machine learning model, such as the DeepConvLSTM [11]. While subsequent research has sought to enhance models' capabilities through the integration of mechanisms like attention [20] to better model temporal patterns, these models remain fundamentally constrained by their dependence on sliding windows, restricting broader temporal understanding across windows [6].

Knowing these temporal limitations of window-based classification, only limited research has aimed to address these with alterations to training procedures of HAR models [8, 13, 17]. Most prominently, Pellatt and Roggen [13] suggested to extend reasoning of LSTMs used within model architectures by making positions among batches temporally connected during training and passing along hidden states between batches. While construction of these so called *CausalBatches* showed to significantly improve classification results, giving models a larger temporal horizon while still using small-sized sliding windows, the LSTM used within these models are tasked to model both local and global temporal structure, yet are optimized on batch-wise gradients derived only from within-batch windows and hence gradients that capture only local patterns. We argue in this work that this mismatch between what the model is asked to learn and what the training signal actually encodes is a fundamental limitation of the CausalBatch approach.

Recently, vision-based Temporal Action Localization (TAL) models have been successfully applied to inertial data and shown to outperform established HAR architectures such as DeepConvLSTM across various datasets [5]. While these TAL models are designed for offline timeline reconstruction, we argue that their core design choice of separating local feature extraction from global temporal reasoning into distinct model components naturally maps onto the intra- and inter-window structure of sliding-window HAR as well as existing models. With human behaviour evolving naturally through local and global patterns, the TAL community has identified that temporal understanding of the past (as well as the future) leading up to a current timestamp that is sought to be classified, is crucial for reliably detecting different types of activities [18].

We thus introduce *DeepConvContext*, a novel inertial-based architecture inspired by TAL methodologies, which is specifically designed to model both short-term (intra-window) and long-term (inter-window) temporal dependencies across sequences of sliding windows of inertial data. Our contributions are three-fold:

- (1) We propose *DeepConvContext*, a novel architecture which effectively uses sequence-based batches for multi-scale recognition via an inter-window LSTM and achieves comparable latency and throughput to related benchmarks, despite its additional module.
- (2) An extensive evaluation on six widely-used HAR benchmark datasets shows that DeepConvContext outperforms the standard DeepConvLSTM by up to 21% and CausalBatch by up to 10% in F1-score, while producing the highest mAP scores across all experiments, improving over related baselines by up to 18 points in an online prediction scenario.
- (3) During ablation experiments we demonstrate that LSTMs consistently outperform attention- and transformer-based

inter-window mechanisms by up to 5% in F1-score across all benchmarks, suggesting that the sequential, locally-ordered inductive bias of LSTMs is better suited to HAR than the global, order-agnostic structure of self-attention.

2 Related Work

With the rise of deep learning, researchers have adapted convolutional neural networks (CNNs) for use on time series sensor data, eliminating the need for manual feature extraction that previously relied on domain expert knowledge. Known as the *DeepConvLSTM*, Ordóñez and Roggen [11] extended this approach by incorporating recurrent layers into inertial-based architectures, enabling them to capture temporal relationships between convolutional features. Despite significant progress in improving methods for extracting temporal information, e.g. via attention mechanisms [1, 10, 20], these architectures rely on a sliding window approach, where each window of sensor data is processed independently to predict its label. As detailed by Bulling et al. [6], this fixed window size limits the model's ability to capture inter-window temporal dependencies, making the choice of window length a critical factor in HAR performance.

Research has therefore emphasized the need for models that can capture both local and global temporal patterns to overcome the limitations of the sliding window approach [7, 8, 17]. Hiremath and Ploetz [8] explored the impact of larger context lengths on HAR performance, aggregating convolutional features using larger kernels before passing them through an LSTM. *CausalBatch* [13] extends the LSTM's temporal horizon by making windows within a batch causally dependent on those in neighbouring batches, allowing LSTM states to persist across batch boundaries without increasing the window size. The Shallow DeepConvLSTM [3], even though it was primarily proposed as a more lightweight alternative to the original DeepConvLSTM, inherently introduces an inter-window learning effect by applying the LSTM across the batch sequence rather than within individual windows. DeepConvContext builds on the insights of both approaches by introducing a dedicated second LSTM explicitly designed for inter-window context modeling, keeping local and global temporal learning structurally separate, making inter-window reasoning an explicit, optimized objective.

3 Methodology

3.1 DeepConvContext

Figure 1 illustrates the proposed *DeepConvContext* architecture, which combines three key design choices.

- (1) **Sequence-based batch loading.** Following Shao et al. [17], we employ a batch loader that, at each iteration, randomly selects a participant and a starting point within their input time series, and extracts a contiguous sequence of B windows from that point. Unlike *CausalBatch* [13], which connects hidden states across otherwise unordered batches, this preserves temporal consistency among batch elements directly, enabling the model to learn inter-window patterns from more informative gradient updates. To avoid overfitting, the sampling start point is randomized per batch, and all LSTM hidden states are reset between batches.

- (2) **Intra-window feature extraction.** Each window in the input sequence is processed independently by a feature extractor that is identical to the original DeepConvLSTM [11]. LSTM hidden states are reset between windows, ensuring that this module captures only local, within-window temporal structure. The output of each window’s extractor is summarized into a fixed-size feature vector of dimension $1 \times LSTM_h$, where $LSTM_h$ denotes the number of hidden units in the final intra-window LSTM layer.
- (3) **Inter-window context modeling.** Lastly, the sequence of B window-level feature vectors is passed to a second, context-based LSTM that models temporal dependencies across windows. Its output sequence is passed through a dropout layer and a classification layer to produce per-window predictions. Because each window is processed by both the intra- and inter-window LSTMs, the final output vectors encode both local temporal structure and each window’s temporal relationship to earlier windows in the sequence.

3.2 Datasets

We base our experiments on six widely used HAR datasets: the Wetlab [16], WEAR [4], SBHAR [14], RWHAR [19], Opportunity [15], and Hang-Time datasets [9]. This selection provides a diverse set of prediction scenarios, each posing distinct challenges for activity recognition models to address. Among these challenges are variations in the number of participants and activity classes, the presence of a NULL-class [4, 9, 15, 16], and the inclusion of complex, short-duration, transitional, and locomotion activities [4, 9, 14–16, 19]. Additionally, the datasets vary in sensor configurations, with several collected in multi-sensor environments [4, 15, 19].

3.3 Baselines & Architecture Variants

We compare DeepConvContext against three baselines: (1) a classic DeepConvLSTM trained on randomly sampled windows, (2) a DeepConvLSTM trained using *CausalBatch* [13], and (3) the Shallow DeepConvLSTM [3]. Specifically, each of these baselines models different temporal structures: shuffling restricts (1) to intra-window context only, while (2) and (3) extend temporal context during training by propagating the LSTM hidden state between batches, resetting every $l = 100$ batches in (2) and across batch elements in (3). During testing, all baselines and the proposed DeepConvContext are applied online, predicting participant timelines batch by batch with hidden states carried across the full test timeline, ensuring minimal latency.

LSTMs, though effective at modeling local temporal dependencies, are known to struggle with long-range relationships as the temporal distance between relevant elements grows [2, 12]. Self-attention mechanisms can capture dependencies between any two sequence elements regardless of temporal distance, and have driven notable performance gains in inertial sensor-based architectures such as Attend-and-Discriminate [1], making them a natural candidate for inter-window context modeling. However, unlike recurrent architectures, attention-based models lack a natural sequential state that can be updated incrementally, making per-window online inference computationally expensive. This limits their suitability for

streaming deployment compared to LSTMs, which process inputs one step at a time with constant-cost state updates. We therefore include both multi-head attention and transformer layers as controlled drop-in replacements for the inter-window LSTM in our ablation study, rather than as primary design choices. To ensure comparability to unidirectional LSTMs, we restrict both attention-based variants to their causal formulations. Nevertheless, since activity recognition can benefit from future context at prediction time, we also report results for bidirectional variants of all three inter-window mechanisms (LSTM, attention, and transformer). Note that bidirectional LSTMs, same as attention-based variants, require access to a full batch input sequence before inference, making them less suited for online activity recognition purposes.

3.4 Training

For each experiment, we perform Leave-One-Subject-Out (LOSO) cross-validation, where each participant in the dataset is used as the validation set once, while all other participants are used for training. Each experiment is repeated three times with varying random seeds. Averaging these three runs, we report two key metrics for each experiment: the class-averaged macro F1-score and the mean Average Precision (mAP), both averaged across all validation splits (i.e., participants). The mAP, which evaluates the overlap between predicted activity segments and ground truth annotations, is computed across five temporal Intersection-over-Union (IoU) thresholds (0.3, 0.4, 0.5, 0.6, and 0.7), following the evaluation conventions established in the TAL literature [18]. We report the average across these five thresholds as the final mAP metric for each experiment. To ensure that both metrics (F1 and mAP) are computed on the same temporal resolution, we convert the windowed predictions of each model back to per-sample predictions by unwinding them. The final reported values per dataset are the averages of the IoU-averaged mAP and class-averaged F1-score across all validation subjects. All model architectures are trained using a weighted cross-entropy loss and the Adam optimizer, with a learning rate of $1e^{-4}$, weight decay of $1e^{-6}$. Each model is trained for 30 epochs, using a fixed learning rate schedule that multiplies the learning rate by a factor of 0.9 every 10 epochs. All architectures evaluated in this benchmark use four convolutional layers with 64 filters, each of size 9×1 , except for the Opportunity dataset, where a smaller filter size of 5×1 is used due to the lower sampling rate of the input signal. As suggested by Bock et al. [3], LSTMs use a single layer with a hidden size of 128 (see appendix for comparison to two layered variants). In ablation experiments where LSTMs are replaced with transformers or multi-head self-attention mechanisms, we use four attention heads with a hidden dimension of 768 and no dropout. The transformer-based variants use two layers, 4 heads and a smaller hidden dimension of 256. Same as Ordóñez and Roggen [11], embeddings get passed through a dropout layer with a probability of 0.5 before being passed to the classifier.

4 Results

4.1 Benchmark analysis

Figure 2 presents results for the standard DeepConvLSTM [11], a CausalBatch-trained DeepConvLSTM [13], the Shallow DeepConvLSTM [3], and the proposed DeepConvContext, all using a

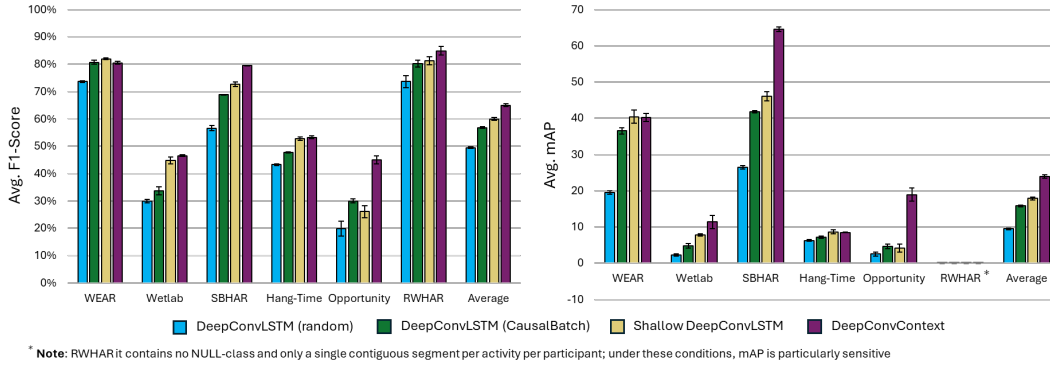


Figure 2: Average F1-score and mAP results of the DeepConvLSTM [11], Shallow DeepConvLSTM [3] and proposed DeepConvContext applied to the investigated datasets. One can see that the DeepConvContext combines strengths of both architectures and improves upon results across all datasets, providing the highest F1-score and mAP.

single LSTM layer per module. On average, the multi-scale DeepConvContext achieves the highest prediction performance, with an average F1-score of 64.96%, compared to 49.46% for the standard, 56.84% for the CausalBatch-trained and 59.92% for the Shallow DeepConvLSTM, across the six evaluated datasets. Furthermore, as DeepConvContext leverages both intra- and inter-window context, it produces the highest mAP scores across all experiments, outperforming the Shallow DeepConvLSTM by as much as 18 score points for the SBHAR and 14 score points for the Opportunity dataset.

Although all models propagate LSTM hidden states across predictions at test time, the standard DeepConvLSTM, trained without inter-window supervision, consistently produces the least coherent segments, as reflected in its substantially lower mAP across all datasets. While Pellatt and Roggen [13] claim to address the limited temporal reasoning of the DeepConvLSTM, both mAP results and qualitative segment visualizations in Figure 4 show that sequence-based learning, as used in the Shallow DeepConvLSTM and DeepConvContext, outperforms CausalBatch in both prediction accuracy and timeline reconstruction quality. As also seen in Figure 4, DeepConvContext shows less frequent misclassifications of NULL windows into activity classes, which we hypothesize is due to its dedicated inter-window LSTM that helps the model leverage surrounding temporal context to disambiguate locally activity-like but isolated windows. We attribute this to the stronger inter-window training signal produced by gradients computed over temporally ordered sequences, as opposed to the within-batch gradients of CausalBatch which capture only local structure. All in all, the DeepConvContext effectively combines the strengths of both intra- and inter-window learning, maintaining high predictive performance on all activity type, as shown in the confusion matrices in Figure 3.

4.2 Ablation experiments

Impact of Context Length. With the DeepConvContext architecture using temporal sequences as batches, applied batch sizes during training become an important tuneable hyperparameter as they influence the learnable inter-window context. To measure sensitivity of our proposed approach to the applied training batch size, Table 1 shows results of applying smaller (25 and 50) as well as a larger

Table 1: Average F1-score and mAP results of DeepConvContext using varying train sequence batch sizes (25, 50 and 200). We report results for the WEAR [4], Wetlab [16], Hang-Time [9], RWHAR [19], Opportunity [15] and SBHAR [14]. Results show that DeepConvContext remains largely stable across context lengths, with longer sequences yielding modest gains on datasets with complex or long-duration activities.

D	25		50		200	
	F1	mAP	F1	mAP	F1	mAP
[4]	79.14%	36.43	80.78%	39.38	80.22%	41.07
[16]	42.04%	9.50	45.39%	10.87	46.63%	10.83
[9]	53.97%	9.13	52.84%	8.60	52.67%	8.54
[19]	80.89%	0.0	85.28%	0.01	88.62%	0.01
[15]	39.07%	12.01	43.84%	17.47	46.71%	20.88
[14]	74.82%	53.47	77.36%	56.63	76.17%	62.96
Avg	61.65%	20.09	64.25%	22.16	65.17%	24.05

training batch size (200). Results show that DeepConvContext remains largely stable across all tested context lengths. Notably, even with as few as 25 windows (corresponding to 12.5 seconds of data) the model retains competitive performance across most datasets, suggesting that inter-window learning is effective even under the short-buffer constraints typical of wearable deployment. Nevertheless, datasets with complex or long-duration activities, such as RWHAR and Opportunity, show consistent gains with longer context, indicating that the optimal batch size is activity-type dependent and represents a tunable deployment parameter.

LSTMs vs. Attention. As detailed in Section 3.3, we replace the newly introduced second context-based LSTM in the DeepConvContext architecture with either a multi-head attention block or a transformer module. Results in Table 2 show that LSTMs outperform attention-based mechanisms consistently across the selected benchmark datasets by as much as 5% in F1-score. With LSTMs being sequential by design, LSTM-based results further produce higher mAP scores, as they allow passing of hidden states among batches and are thus not dependent on applying the same batch size during training and testing. Lastly, while attention mechanisms allow features to interact across the entire input sequence, they typically exhibit a reduced sensitivity to the natural order of events in

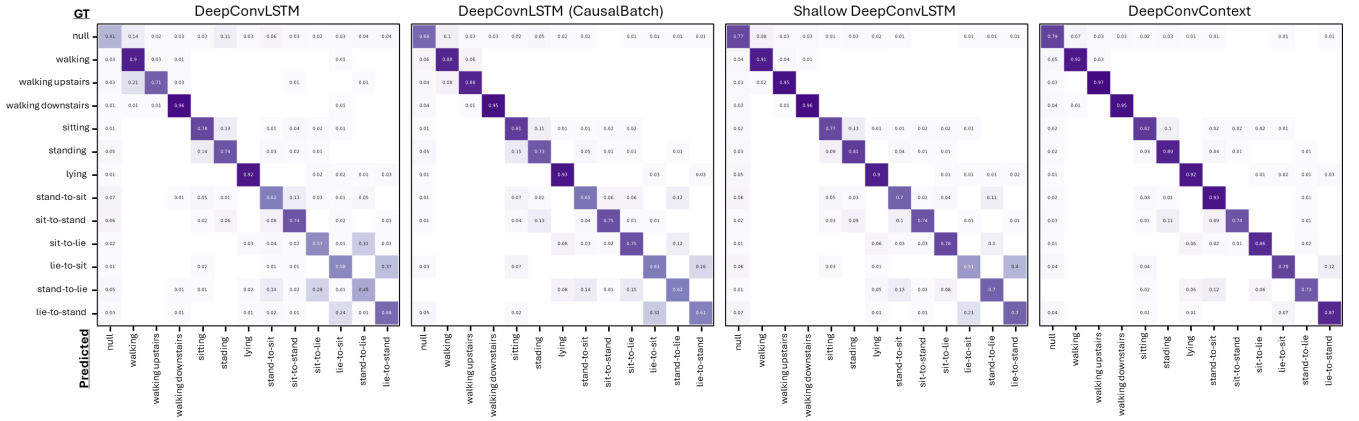


Figure 3: Per-class confusion matrices of the DeepConvLSTM, Shallow DeepConvLSTM and DeepConvContext being applied to the SBHAR dataset using LOSO cross-validation. DeepConvContext improves upon these benchmarks, with specifically transition classes such as *sit-to-stand* being more reliably detected.

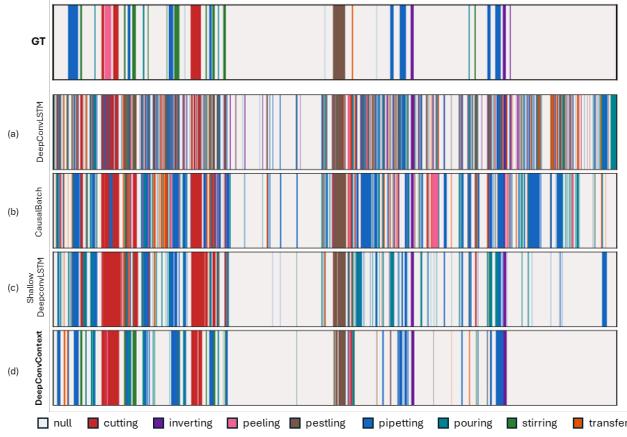


Figure 4: Color-coded visualization of a sample participant from the Wetlab dataset reconstructed by (a) classic DeepConvLSTM, (b) CausalBatch, (c) Shallow DeepConvLSTM and (d) our proposed DeepConvContext. One can see that DeepConvContext most effectively recognizes and reconstructs ground truth (GT) segments.

time and rely on positional encodings to provide temporal structure. Although we included such encodings in our models, the relatively lower performance of the attention-based variants suggests that LSTMs may be more effective at capturing temporal relationships among windows.

Complexity and Inference Efficiency. Table 3 compares the model complexity and inference efficiency of the proposed DeepConvContext with its variants as well as predecessors. Compared to CausalBatch, which maintains B separate hidden states across batch boundaries during training, DeepConvContext processes a single temporally ordered sequence per batch. By passing only a compressed, fixed-size feature vector per window to the inter-window LSTM DeepConvContext achieves a memory footprint comparable to the Shallow DeepConvLSTM (123.0 MB vs. 112.5 MB) and is comparable

Table 2: Average F1-score results of DeepConvContext using (unidirectional or bidirectional) LSTM, attention or a Transformer to learn inter-window context. We report results for WEAR [4], Wetlab [16], Hang-Time [9], RWHR [19], Opportunity [15] and SBHAR [14]. LSTMs significantly outperform both attention and transformers. Best unidirectional and bidirectional results per dataset are underlined.

D	Unidirectional			Bidirectional		
	LSTM	Att.	Tran.	LSTM	Att.	Tran.
[16]	<u>46.52%</u>	36.45%	35.94%	<u>53.63%</u>	44.21%	39.79%
[4]	<u>80.50%</u>	79.21%	78.53%	<u>83.24%</u>	82.24%	80.83%
[9]	<u>53.27%</u>	48.81%	48.82%	<u>59.12%</u>	53.84%	50.55%
[19]	<u>84.92%</u>	83.54%	83.10%	<u>88.91%</u>	88.31%	<u>88.93%</u>
[15]	<u>45.06%</u>	39.18%	41.42%	<u>50.58%</u>	35.52%	37.90%
[14]	<u>79.48%</u>	69.41%	71.34%	<u>80.22%</u>	73.71%	75.87%
Avg	<u>64.96%</u>	59.43%	59.86%	<u>69.28%</u>	62.97%	62.31%

Table 3: Model complexity and inference efficiency of CausalBatch, Shallow DeepConvLSTM (Shallow D.), and DeepConvContext (DCC) variants, all with 1-layer LSTMs, on a single NVIDIA GeForce RTX 3090. Attention-based variants refer to their non-causal variants. We report parameters (ρ), GPU memory, latency (mean $\pm$ std, batch size = 100), and throughput (sps). Though having almost three times as many parameters, DCC remains comparable in memory and latency to CausalBatch.

Architecture	ρ	Memory	Latency (ms)↓	(sps)↑
CausalBatch	278k	114.4 MB	0.53 ± 0.03	188k
Shallow D.	278k	112.5 MB	4.18 ± 0.11	24k
DCC (LSTM)	704k	123.0 MB	0.96 ± 0.10	105k
DCC (Bi-LSTM)	969k	126.0 MB	1.39 ± 0.16	72k
DCC (Att.)	639k	121.1 MB	1.08 ± 0.02	93k
DCC (Tran.)	837k	124.1 MB	1.15 ± 0.05	87k

in latency (0.96 ms vs. 0.53 ms for CausalBatch) and throughput (105k vs. 188k samples per second) to CausalBatch while having significantly more learnable parameters.

Table 4: Average F1-score results of DeepConvLSTM (DCL), a CausalBatch-trained DeepConvLSTM, Shallow DeepConvLSTM (Shallow D.) and proposed DeepConvContext (DCC) using 1-layered (1-L) or 2-layered LSTMs (2-L) within the architectures. We report results for WEAR [4], Wetlab [16], Hang-Time [9], RWHAR [19], Opportunity [15] and SBHAR [14]. On average one-layered LSTMs outperform 2-layered LSTMs across all architectures.

<i>D</i>		DCL	CausalBatch	Shallow D.	DCC
[4]	1-L	73.64%	80.73%	81.98%	80.50%
	2-L	73.37%	75.15%	76.70%	78.46%
[16]	1-L	29.89%	33.63%	44.79%	46.52%
	2-L	26.42%	24.22%	36.00%	42.25%
[9]	1-L	43.22%	47.79%	52.76%	53.27%
	2-L	40.80%	47.97%	49.58%	52.21%
[19]	1-L	73.64%	80.14%	81.27%	84.92%
	2-L	70.53%	68.66%	69.48%	89.50%
[15]	1-L	19.84%	29.92%	26.08%	45.06%
	2-L	29.41%	37.78%	38.05%	40.41%
[14]	1-L	56.55%	68.81%	72.66%	79.48%
	2-L	55.73%	66.56%	59.93%	71.29%
Avg	1-L	49.46%	56.84%	59.92%	64.96%
	2-L	49.37%	53.39%	54.96%	62.35%

Revisiting shallow LSTMs. Using a similar approach as [3], we compare investigated baselines, and DeepConvContext using 1-layered and 2-layered LSTMs. Results (see Table 4) show that on average single-layer LSTMs outperform their two-layer counterparts across all tested datasets and models, with the exception being our tested baselines applied to the Opportunity dataset as well as the DeepConvContext applied to the RWHAR dataset.

5 Limitations

Although not specifically tested within this work, architectural design choices of the DeepConvContext can be readily extended to other inertial-based models such as TinyHAR [20] and Attend-and-Discriminate [1]. While our analysis evaluates only modified versions of the original DeepConvLSTM, we expect the observed performance improvements to generalize to other architectures, as they follow a common design pattern of combining convolutional feature extraction with some form of temporal modeling. Compared to the original DeepConvLSTM and the Shallow DeepConvLSTM, our architecture extends the original model design with additional modules and hence increases learnable parameters, yet, as shown in Table 3, has comparable throughput rate and model inference speed through the use of a dimensionality reduction layer. With HAR research and their applications using edge devices focus mostly on online activity recognition approaches, we did not expand on the offline activity recognition performances of DeepConvContext. Yet, as ablation experiments using bidirectional LSTMs within the DeepConvContext architecture show increased performance on HAR datasets, we expect DeepConvContext to be also suited for these purposes.

6 Discussion & Conclusions

In this paper we introduced *DeepConvContext*, a novel architecture for inertial-based activity recognition. Inspired by design choices

in the vision-based Temporal Action Localization community, the architecture adopts a multi-scale approach that learns both intra- and inter-window temporal dependencies through separate model components to predict the label of a given sliding window. While our approach challenges the conventional definition of a batch by leaving temporal consistency among sliding windows in batches intact, our analysis revealed that these sequence-based batches are inherently more suitable for learning inter-window temporal features. Our benchmark analysis shows that DeepConvContext outperforms the standard DeepConvLSTM by up to 21% and the Shallow DeepconvLSTM by on average 5% in F1-score, while achieving the highest mAP across all six datasets – improving of up to 18 points over the Shallow DeepConvLSTM, reflecting more coherent and less fragmented activity segment predictions. Unlike common post-hoc smoothing or filtering approaches (e.g., majority voting), which act only on already-made per-window decisions and can erode short but genuine transition activities, DeepConvContext learns coherent segments directly from the sensor data by shaping representations before a decision is made. Ablation experiments further reveal that LSTMs consistently outperform attention and transformer-based inter-window mechanisms across all benchmarks, suggesting that the sequential, locally-ordered inductive bias of LSTMs is better suited to HAR than the global, order-agnostic structure of self-attention. Context length ablations further reveal that DeepConvContext is robust to the choice of batch size, retaining competitive performance with as few as 25 windows (12.5 seconds of data).

With DeepConvContext, we present the HAR community with a novel method for processing windowed data and predicting activity labels. We directly address the well-known sliding window problem [6] and align with the community’s ongoing efforts to overcome it [8, 13]. By compressing inter-window communication to a fixed-size feature vector, our suggested changes do not come with sacrifices in efficiency and are easily extendable to other architectures. We thus hope that our proposed multi-scale approach will become an essential component within future works.

Acknowledgments

This project was supported by the ERC Consolidator Grant FORHUE (101044724), the DFG Project WASEDO (506589320) and the University of Siegen’s OMNI cluster.

References

- [1] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezaatofghi, 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). doi:10.1145/3448083
- [2] Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent Is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166. doi:10.1109/72.279181
- [3] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. Improving Deep Learning for HAR with Shallow LSTMs. In *International Symposium on Wearable Computers*. ACM. doi:10.1145/3460421.3480419
- [4] Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. 2024. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024), 1–21. doi:10.1145/3699776
- [5] Marius Bock, Michael Moeller, and Kristof Van Laerhoven. 2024. Temporal Action Localization for Inertial-based Human Activity Recognition. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024). doi:10.1145/3699770

- [6] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *Comput. Surveys* 46, 3 (2014). doi:10.1145/2499621
- [7] Nils Y. Hammerla and Thomas Plötz. 2015. Let's (Not) Stick Together: Pairwise Similarity Biases Cross-Validation in Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. doi:10.1145/2750858.2807551
- [8] Shruthi Kashinath Hiremath and Thomas Ploetz. 2021. On the Role of Context Length for Feature Extraction and Sequence Modeling in Human Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. doi:10.1145/3460421.3478825
- [9] Alexander Hoelzemann, Julia Lee 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. *Sensors* 23, 13 (2023). doi:10.3390/s23135879
- [10] Vishvak S. Murahari and Thomas Plötz. 2018. On Attention Models for Human Activity Recognition. In *International Symposium on Wearable Computers*. ACM. doi:10.1145/3267242.3267287
- [11] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *MDPI Sensors* 16, 1 (2016). doi:10.3390/s16010115
- [12] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the Difficulty of Training Recurrent Neural Networks. In *International Conference on Machine Learning (Proceedings of Machine Learning Research)*. PMLR. <https://proceedings.mlr.press/v28/pascanu13.html>
- [13] Lloyd Pellatt and Daniel Roggen. 2020. CausalBatch: Solving Complexity/Performance Tradeoffs for Deep Convolutional and Lstm Networks for Wearable Activity Recognition. In *International Joint Conference on Pervasive and Ubiquitous Computing and International Symposium on Wearable Computers*. ACM. doi:10.1145/3410530.3414365
- [14] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171 (2016). doi:10.1016/j.neucom.2015.07.085
- [15] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkel, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millán. 2010. Collecting Complex Activity Datasets in Highly Rich Networked Sensor Environments. In *International Conference on Networked Sensing Systems*. IEEE. doi:10.1109/INSS.2010.5573462
- [16] Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. 2015. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. doi:10.1145/2750858.2807547
- [17] Shuai Shao, Yu Guan, and Victor Sanchez. 2024. Beyond Isolated Frames: Enhancing Sensor-Based Human Activity Recognition through Intra- and Inter-Frame Attention. *CoRR* abs/2405.19349 (2024). arXiv:2405.19349 [eess] doi:10.48550/arXiv.2405.19349
- [18] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. TriDet: Temporal Action Detection With Relative Boundary Modeling. In *Conference on Computer Vision and Pattern Recognition*. IEEE. <https://doi.org/10.1109/cvpr52729.2023.01808>
- [19] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-Body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *International Conference on Pervasive Computing and Communications*. IEEE. doi:10.1109/PERCOM.2016.7456521
- [20] 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 *International Symposium on Wearable Computers*. ACM. doi:10.1145/3544794.3558467