Integrating Adaptive Sampling into Deep Learning-Based Human Activity Recognition

Pavankumar Chandankar[1], Marius Bock[2,3], Juergen Gall[2,3], Michael Moeller[1], and Kristof Van Laerhoven[1]

[1] University of Siegen, Siegen, Germany
Pavankumar.Chandankar@student.uni-siegen.de
[2] University of Bonn, Bonn, Germany
bock@iai.uni-bonn.de
[3] Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Abstract. Deep learning has substantially improved the accuracy of Human Activity Recognition (HAR) from wearable sensors, but existing pipelines mostly rely on fixed, high-frequency sampling strategies for the inertial sensors. Such designs mostly lead to unnecessary sensing, computation, and energy consumption, for instance during low-motion or repetitive activities. We show in this work that many activities in current benchmarks can be recognized accurately at much lower sampling rates, while others do benefit from higher temporal resolution. We then introduce a frequency-adaptive HAR framework that dynamically selects the sampling rate most appropriate for each activity. The framework is enabled by a single-checkpoint multirate ContinuousConvLSTM model, whose continuous convolution kernels adjust their effective temporal support as a function of sampling frequency. We evaluate ContinuousConvLSTM and quantify the achievable accuracy–efficiency trade-offs relative to fixed-frequency baselines.

Keywords: wearable activity recognition · inertial-based activity recognition · human activity recognition · adaptive sampling

1 Introduction and Motivation

Human Activity Recognition (HAR) has evolved considerably in the past decade with the advent of deep learning, enabling improved accuracy in classifying complex, real-world human behavior from wearable sensor data [9,1]. However, as the field matures, the focus is increasingly moving beyond raw performance metrics, towards efficiency, deployability, and real-world applicability [13,4]. The rise of always-on, battery-powered wearables, from smartwatches to clinical-grade health monitors, requires models that are both accurate and lightweight, as well as energy efficient and capable of operating directly on resource-constrained

Source code is available at GitHub: Integrating-Adaptive-Sampling-into-Deep-Learning-Based-HAR.

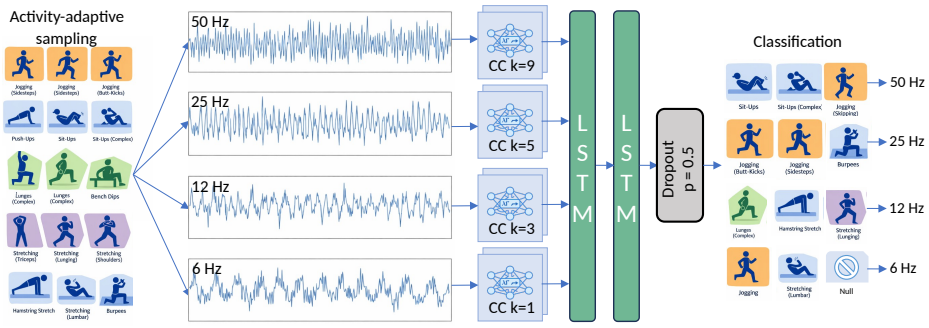

Fig. 1: Overview of the proposed multirate ContinuousConvLSTM. Instead of training separate models for each sampling rate, a single checkpoint supports different (50, 25, 12, and 6 Hz) inputs. ContinuousConv layers adapt their effective temporal kernel length (9, 5, 3, 1) according to the sampling rate, preserving temporal coverage across resolutions while sharing a common recurrent classifier.

devices [4]. This shift has spurred innovation towards compact deep learning architectures [13], quantization-aware training [5], and on-device optimization.

In many wearable HAR applications, systems must operate for extended durations on a single battery charge. Additionally, these deployments exhibit considerable device heterogeneity: smartphones, smartwatches, fitness trackers, and clinical-grade inertial units offer access to diverse sensors at varying default sampling rates. Operating system-induced sample-timing variability further complicates the use of fixed-interval models [18]. Therefore, an effective HAR pipeline should accommodate, and ideally exploit, sampling-rate variability.

Despite recent advances, a fundamental inefficiency remains: Most deep learning pipelines for human activity recognition assume fixed, high-frequency sensor sampling rates, which are typically designed for worst-case scenarios, rather than reflecting the dynamic and context-dependent nature of human activity. This approach leads to unnecessary computational and energy costs, especially when sensors collect redundant data during periods of inactivity or low-motion contexts [7,4]. A significant technical challenge for deep learning in accommodating variable sampling rates is that standard convolutional networks utilize discrete kernels. A kernel of length k covers different time periods at different sensor rates, breaking temporal alignment when switching between, for example, 50 Hz and 6 Hz inputs. To keep comparable temporal receptive fields across different rates, the effective kernel length must thus be adjusted to the sampling frequency.

Stationary activities can be reliably recognized at low sampling frequencies below 10 Hz, while highly dynamic motions need 25 to 50 Hz or higher [7]. Additionally, large-scale hyperparameter studies of deep HAR architectures indicate that the optimal temporal kernel size varies across datasets and tasks, and that no single discrete kernel generalizes effectively between recognition problems [17].

These two issues, rigid sampling rates and discrete kernel sizes, are often considered independent design choices. We argue that both relate to the fundamental question: *What is the most efficient temporal representation of the signal that enables the model to extract patterns across all activities?*

This paper addresses this challenge by parameterizing convolutional filters in continuous time. Instead of learning kernel values on a fixed grid, a coordinate-conditioned kernel function is learned over normalized temporal offsets, which enables the instantiation of discrete kernels for any target sampling rate. This approach was initially developed for non-grid data such as 3D point clouds [14] and has recently demonstrated effectiveness for general sequential data [11]. This papers investigates the use of parameterized convolutional filters and how they can be extended to wearable human activity recognition (HAR). This extension allows a single model to process windows from multiple sensor rates, while maintaining a consistent representation of physical time. When combined with rate-specific feature extractors that share a common recurrent and classification head, this method eliminates the need for both kernel size and per-rate retraining as manual hyperparameters. Our contributions are threefold:

1. We demonstrate that replacing fixed discrete convolutional kernels with deep parametric convolutions [14] allows inertial-based human activity recognition (HAR) architectures to dynamically adjust their effective kernel size based on the sampling rate. This method is shown to eliminate manual tuning of temporal kernel size across three HAR benchmarks: WEAR [2], RealWorld-HAR [12], and WISDM-watch [15].
2. We introduce *ContinuousConvLSTM*, an extension of DeepConvLSTM [9], which uses the parametric convolutions for feature extraction and inference across multiple sampling rates within a single checkpoint. The recurrent and classification components are shared for all rates, while dedicated convolutional branches address the different temporal resolutions.
3. We evaluate ContinuousConvLSTM on three benchmarks against per-rate DeepConvLSTM and pure-CNN baselines, showing that one checkpoint covers the full rate range without kernel-size tuning, and that the recurrent stage is what preserves accuracy at the lowest rates.

2 Related Work

Deep Learning Approaches for Inertial HAR. Deep learning has emerged as the dominant approach for inertial human activity recognition (HAR) due to its ability to learn hierarchical temporal features directly from raw sensor data. The DeepConvLSTM architecture [9] established a widely adopted baseline by integrating temporal convolutions with recurrent modeling, thereby capturing both local motion patterns and long-range temporal dependencies. Subsequent research has investigated enhanced sequence modeling and attention mechanisms [1], ensemble methods that improve performance stability across subjects and sensor placements [6,26], and systematic hyperparameter analyses that delineate the design space of deep, convolutional, and recurrent HAR models [17].

More compact variants, such as a DeepConvLSTM with a single-layer LSTM [19], the transformer-based TinyHAR [13], and the ultra-lightweight TinierHAR [24], have further reduced parameter count and computational cost while maintaining recognition accuracy on standard benchmarks. The present work adopts the DeepConvLSTM backbone as a fixed, well-understood foundation and focuses on rate-aware feature extraction. Instead of seeking improved architectures, this study examines how the convolutional stage should be structured when the sampling rate is not a fixed design parameter.

Sampling Rate and Adaptive Sensing. The selection of sampling rate significantly influences human activity recognition (HAR) performance, with several studies demonstrating that the optimal rate depends on the specific activity rather than the sensor or dataset [7,27]. Stationary and quasi-static behaviors can be reliably recognized at low frequencies, whereas highly dynamic motions require higher temporal resolution. This creates a trade-off that cannot be resolved by a single global rate. Recent work further shows that sampling irregularity itself can affect deployed HAR pipelines, linking rate selection to robustness under non-ideal sensing conditions [25]. Practical approaches to sampling-rate adaptivity have been explored in two main directions. Classical adaptive-sensing systems, such as AdaSense [10], adjust the sampling rate of body sensor networks to satisfy user-specified accuracy requirements while conserving sensing and communication power. More recent learning-based methods treat the sampling rate as a decision variable. For example, Datum-Wise Frequency Selection [3] learns a per-sample sampling policy using a Markov decision process; FreqSense [16] integrates a multi-resolution network with conditional early exit under a tunable compute budget; and CoSS [8] jointly optimizes sensor modality and sampling rate to avoid consistently sampling at the highest-rate configuration. These approaches all assume the availability of a separate model, or a model branch with a separately tuned kernel, for each operating rate. The present work extends this line of research by introducing a single architecture whose convolutional feature extractor adapts internally to the sampling rate. As a result, runtime rate selection-whether learned, oracle-guided, or rule-based-no longer requires maintaining a separate model for each rate.

Energy-Aware Human Activity Recognition. A complementary research direction enables human activity recognition (HAR) on resource-constrained devices by reducing computational, memory, and communication requirements. Recent surveys systematise this literature and contend that recognition performance and energy consumption should be optimised jointly rather than independently [4]. Approaches in this domain include model compression, pruning, network-level quantisation [5], adaptive inference [23], and efficient architecture design.

Continuous and Parametric Convolution. Standard convolutional networks utilize discrete kernels that are fixed to an input grid, which restricts their effectiveness when inputs are irregularly sampled or exhibit variable temporal resolution [21]. Wang et al. [14] introduced Parametric Continuous Convolution

(PCC), a learnable operator in which the kernel is formulated as a coordinate-conditioned function, approximated by a small multilayer perceptron (MLP) and evaluated at the relative offsets of neighboring points. Initially developed for 3D point-cloud segmentation, PCC eliminates the grid constraint and enables the operator to function at arbitrary support resolutions. Romero et al. [11] extended this approach to one-dimensional sequence modeling with CKConv, parameterizing the convolutional kernel as a continuous function so that the receptive field is defined in physical time, rather than in discrete taps. Related work on FlexConv learns differentiable kernel sizes within a continuous-kernel formulation [20]. Continuous convolution has also been explored for non-uniform event sequences, for example in COTIC [22].

To the best of our knowledge, continuous-kernel convolution operators have not previously been applied in wearable HAR. In this paper, we adapt parametric Continuous Convolution to wearable sensor time series, by constructing convolution kernels from a continuous temporal support. This ContinuousConv operator generates sampling-rate-specific kernels whose effective length is determined automatically from the input frequency, while keeping a consistent temporal receptive field. This decouples kernel support from the sampling rate, to reduce the need for rate-specific architectural tuning.

3 Single-Checkpoint Multirate HAR with Parametric Continuous Convolution

We argue in this paper for the need for a more flexible design: The sampling rate should be chosen at runtime according to the activity, instead of being fixed once during training. With a standard model such as the DeepConvLSTM, however, this is difficult to realize in practice. A discrete temporal kernel of length k_t covers k_t/f_s seconds, which means that its physical receptive field changes whenever the sampling rate changes. Supporting multiple rates would therefore require multiple separately tuned kernel sizes, multiple independently trained checkpoints, and an additional switching logic at inference time. This section introduces a model that removes both costs: a *single* checkpoint that operates at any target rate, with no per-rate kernel-size search.

3.1 Continuous Temporal Kernels

The coupling between kernel size and sampling rate happens because a discrete kernel stores a fixed number of weights on the temporal grid. We can break this coupling by representing each temporal kernel as a coordinate-conditioned function rather than a stored tensor, following Parametric Continuous Convolution [14]: A small multilayer perceptron (MLP) ϕ maps a normalized temporal offset $p \in [-1, 1]$ to a vector of kernel values, and the discrete kernel for a given rate is obtained by querying ϕ on a grid whose length is set by a temporal support fixed in seconds. For a target sampling rate f_s and with τ being the

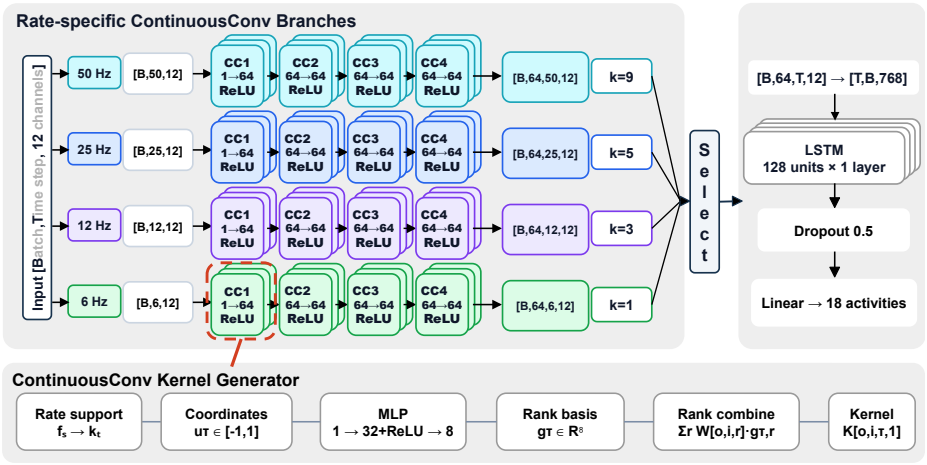

Fig. 2: The single-checkpoint multirate ContinuousConvLSTM, assuming 50 Hz data as an example: During training, windows are presented at all supported sampling rates (50,25,12,6 Hz). During inference, the input rate is snapped to the nearest supported rate and resampled accordingly. The resampled window is routed through the corresponding ContinuousConv branch comprising four stacked ContinuousConv–ReLU layers. The effective temporal kernel length is determined by Eq. (1). Thus, the sampling rate controls the temporal support of the convolution, while the ContinuousConv kernel-generation remains identical across rates. Features are flattened and processed by a shared LSTM classifier.

temporal support, the effective discrete kernel length k_t is obtained by:

$$k_t \;=\; \mathrm{odd}\big(\max\big(1,\; \mathrm{round}(\tau f_s)\big)\big)\,, \tag{1}$$

where $\mathrm{odd}(\cdot)$ rounds to the nearest odd integer so that symmetric "same" padding preserves the temporal length. We initialize $\tau = 0.18\,\mathrm{s}$ for 50 Hz inertial data, matching the standard $k_t = 9$ kernel at 50 Hz; The same τ then yields $k_t = 5, 3, 1$ at $25, 12, 6$ Hz. Since τ is fixed in physical time, the kernel spans approximately the same duration at every rate, and no separate kernel-size search is needed when the sampling rate changes.

3.2 A Single-Checkpoint ContinuousConvLSTM

We obtain the multirate baseline by replacing the four discrete temporal convolutions of the DeepConvLSTM model [9] with four continuous-kernel layers, leaving the recurrent and classification stages unchanged. To serve all target rates from one checkpoint, the model instantiates one convolutional branch per rate $f_s \in \mathcal{F}$, with $\mathcal{F} = \{50, 25, 12, 6\}$ Hz; every branch shares a single LSTM and linear classifier, so only the feature extractor is rate-specific (Fig. 2).

The discrete kernel for rate r, layer $l \in \{1, \ldots, 4\}$, output channel o, input channel i, and tap position t with coordinate $p_t \in [-1, 1]$, is given by:

$$K_{o,i}^{(r,l)}(t) \;=\; \sum_{a=1}^{R_r} A_{o,i,a}^{(r,l)}\, \phi_{r,l,a}(p_t), \tag{2}$$

where $\phi_{r,l} : [-1, 1] \to \mathbb{R}^{R_r}$ is the layer's coordinate MLP producing R_r basis functions, and $A^{(r,l)} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times R_r}$ is a block-specific channel-mixing tensor. Compared with the discrete DeepConvLSTM kernel, where each tap $W_{o,i}^{(l)}(t)$ is an independently learned parameter tied to a fixed sample index t, Eq. (2) replaces the free tap weights by values generated from a continuous coordinate p_t. The learnable parameters are therefore the temporal basis functions $\phi_{r,l,a}$ and the channel-mixing coefficients $A^{(r,l)}$, while the number of sampled taps k_t can change with the sampling rate. This way, the kernel shape is parameterized in normalized physical time rather than directly in discrete sample positions. The kernel $K^{(r,l)}$ is then applied as an ordinary temporal convolution with k_t taps from Eq. (1). During training, a rate is drawn uniformly from $\mathcal{F}$ per batch, the one-second window is resampled to the corresponding length and routed through the matching branch, so the shared head observes features from all rates. At inference, the input is snapped to the nearest supported rate, $\hat{f}_s = \arg\min_{f \in \mathcal{F}} |f - f_{\text{in}}|$. The result is *a single checkpoint* that operates across $\mathcal{F}$, in contrast to the per-rate checkpoints the discrete baseline would require.

4 Experiments and Results

We evaluate on three publicly available inertial HAR benchmarks spanning different body locations, sensor configurations, native sampling rates, and activity sets (including high-intensity activities as well as low-intensity ones).

WEAR [2] provides synchronized inertial data from four limb-worn devices for 22 participants performing 18 outdoor workouts. We use the 3-axis accelerometers from all four positions (12 channels) and treat the 18 exercises plus background segments, represented as a *null* class, as 19 classes. The streams are provided as uniformly resampled 50 Hz time series [2], which we use as the highest-rate reference.

RealWorld-HAR [12] contains 3-axis waist accelerometer recordings from 15 participants across eight locomotion and posture activities. We use the data at its native 50 Hz sampling rate.

WISDM-watch [15] provides 3-axis smartwatch accelerometer data from 51 subjects across 18 daily-living activities, ranging from locomotion to fine-grained hand motions, at a native sampling rate of 20 Hz.

Constructing lower-rate streams. To study sampling-rate behaviour under controlled conditions, lower-rate inputs are derived from each native stream rather than recorded separately, so the same motion is observed at every rate. We recursively retain every second sample, doubling the sampling interval at each

step: 50, 25, 12, and 6 Hz for WEAR and RealWorld-HAR, and 20, 10, and 5 Hz for WISDM-watch (Fig. 3). This captures the primary effect of reduced temporal resolution, although it may not fully reflect the filtering behaviour of a sensor running natively at the lower rate, since most datasets underwent prefiltering and none of the activities contain truyly high-frequency components. The window duration is fixed at one second across all rates, so the physical time span remains constant and only the number of samples changes: 50/25/12/6 for WEAR and RealWorld-HAR, and 20/10/5 for WISDM-watch.

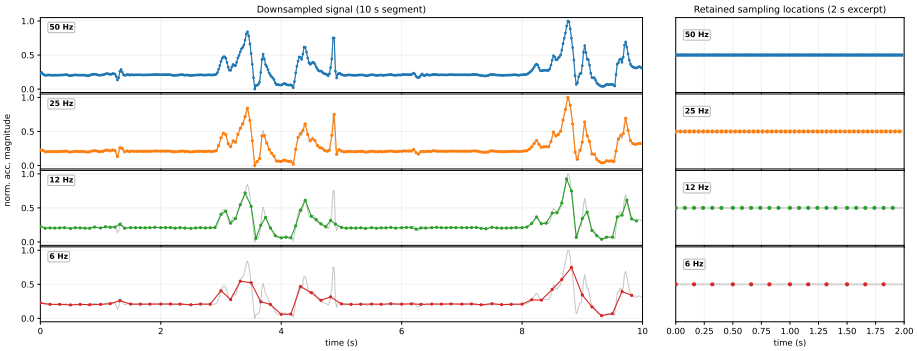

Fig. 3: Constructing lower-rate streams from a WEAR *burpees* segment recorded at the right arm. Left: the downsampled signal over a 10 s segment. Right: the retained sampling locations over a 2 s excerpt.

Windowing, normalization, and labels. Each stream is segmented into one-second windows with 50% overlap, and each window is assigned a single activity label according to the dataset annotations. Per-channel normalization uses statistics computed only from the training portion of each split, which are then reused for validation and testing to avoid data leakage.

Training and Evaluation Protocol. All experiments use leave-one-subject-out (LOSO) cross-fold evaluation, giving 22, 15, and 51 folds for WEAR, RealWorld-HAR, and WISDM-watch, respectively. Models are trained with Adam using a learning rate of 10^{-4}, weight decay of 10^{-6}, and a step schedule with decay factor 0.9 every 10 epochs. Training is performed for 30 epochs using class-weighted cross-entropy loss. We report mean per-class F_1 (macro-F_1) averaged across folds, with standard deviations computed across held-out subjects.

Sampling Rate's Recognition Impact. A single fixed sampling rate is rarely optimal across all activities (Table. 1). Activities with limited body movement can typically be recognized effectively at low sampling rates, others require higher temporal resolution for accurate recognition. To examine the potential benefits

Table 1: Per-activity performance along the sampling rates, on the WEAR dataset, using two standard models, a pure CNN and the DeepConvLSTM model. Values are mean, per-class F_1 scores from LOSO cross validation. For each model, the best scores are underlined, with close scores (within a 2 percent range) in green. Results illustrate that several activities (e.g., stretching and sit-ups) can still be detected accurately at lower sampling rates.

Model:	Pure CNN				DeepConvLSTM			
Activity	6	12	25	50	6	12	25	50
null	0.467	0.495	0.539	0.529	0.633	0.684	0.697	0.698
bench-dips	0.652	0.650	0.660	0.650	0.695	0.614	0.616	0.636
burpees	0.553	0.570	0.628	0.590	0.580	0.605	0.602	0.639
jogging	0.804	0.838	0.888	0.866	0.799	0.847	0.848	0.859
jogging (butt-kicks)	0.735	0.811	0.842	0.831	0.736	0.800	0.799	0.828
jogging (rotating arms)	0.666	0.737	0.795	0.786	0.700	0.774	0.790	0.797
jogging (sidesteps)	0.832	0.878	0.899	0.891	0.857	0.877	0.876	0.884
jogging (skipping)	0.747	0.834	0.890	0.874	0.800	0.874	0.862	0.882
lunges	0.439	0.451	0.518	0.489	0.501	0.537	0.544	0.541
lunges (complex)	0.588	0.597	0.594	0.603	0.628	0.634	0.641	0.638
push-ups	0.582	0.611	0.623	0.600	0.570	0.574	0.574	0.603
push-ups (complex)	0.596	0.619	0.641	0.606	0.602	0.630	0.635	0.619
sit-ups	0.625	0.624	0.636	0.638	0.670	0.658	0.648	0.643
sit-ups (complex)	0.680	0.664	0.683	0.671	0.681	0.682	0.692	0.687
stretching (hamstrings)	0.701	0.695	0.700	0.707	0.718	0.705	0.723	0.715
stretching (lumbar rotation)	0.884	0.898	0.906	0.897	0.877	0.876	0.878	0.867
stretching (lunging)	0.594	0.616	0.647	0.654	0.682	0.701	0.709	0.694
stretching (shoulders)	0.520	0.517	0.541	0.526	0.653	0.701	0.700	0.685
stretching (triceps)	0.757	0.752	0.746	0.748	0.765	0.768	0.749	0.759
Macro mean	0.654	0.677	0.704	0.692	0.692	0.713	0.715	0.720
Macro F_1 optimal per-activity	0.702				0.723			

of adaptive sampling rates, we perform first illustrative study using the WEAR dataset [2], known for its many, highly-dynamic fitness activities, and evaluate the recognition performance for individual activities and over the whole data. For the other datasets, results can be seen in Appendix A. For each sampling rate, separate models (a pure CNN and DeepConvLSTM [9]) are trained and evaluated using leave-one-subject-out (LOSO) cross validation. Averaged over all activities, the recognition quality decreases only slightly as the sampling rate is reduced, from 0.720 for the 50 Hz model to 0.692 for the 6 Hz one, a drop of just 0.028 absolute F_1 across an eight-fold reduction in rate. The drop is gradual rather than abrupt: halving the rate from 50 to 25 Hz costs only 0.005 absolute F_1 ($0.720 \rightarrow 0.715$), and a further halving to 12 Hz remains within 0.01 of the full-rate model. The recurrent LSTM stage shows to be most valuable where

the convolutional receptive field is smallest: at the lowest rates, the discrete kernel shrinks to a few taps, and the LSTM recovers temporal context that the convolutional front-end can no longer capture on its own.

Optimal sampling rates thus vary by activity, so matching each activity to its preferred rate improves recognition accuracy over any fixed-rate model. Assigning each window to its activity-optimal rate (see the "per-activity best" row in Table 1) achieves a mean per-class F_1 of 0.723. While this aggregate gain is modest, it is achieved while recognizing most activities below the full rate. There is also the argument for data efficiency: Since the sampling rate directly determines the number of samples acquired and processed per window, reducing this rate proportionally decreases sensing and computation costs. For example, operating an activity at 25 Hz instead of 50 Hz reduces the data volume by 50%, while 12 Hz and 6 Hz yield reductions of approximately 76% and 88%, respectively. For many WEAR activities, these reductions do not compromise accuracy: Among the 19 classes, 16 achieve their optimum at 25 Hz or lower, including *bench-dips*, both sit-up variants, and several stretching exercises. For these activities, reducing the sampling rate from 50 Hz saves at least half the data while maintaining or even improving accuracy. The effect is clearest for static and low-motion classes such as *bench-dips* and *sit-ups*, which peak at 6 Hz and therefore retain full accuracy at roughly one-eighth of the data rate. Even for activities that perform best at 50 Hz, the per-class trade-offs presented in Table 1 are often minimal, allowing for a modest reduction in accuracy, in exchange for significant energy savings.

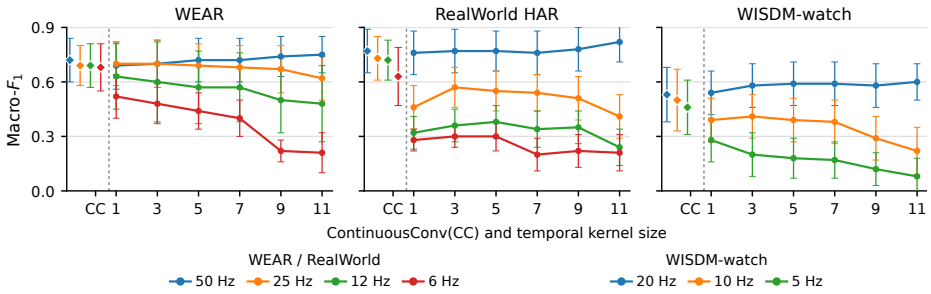

Fig. 4: Comparing the macro-F_1 scores for ContinuousConvLSTM (left of each plot) and the standard DeepConvLSTM (right of each plot) for the tree datasets. Shown is the performance to the discrete temporal kernel size k_t across fixed sampling rates. The ContinuousConvLSTM model uses a single temporal support τ and derives k_t automatically from Eq. (1). Across datasets and sampling rates, the optimal discrete kernel varies substantially, whereas ContinuousConvLSTM consistently achieves performance close to the best manually tuned configuration without rate-specific kernel selection.

Can ContinuousConvLSTM eliminate per-rate kernel tuning? The continuous-kernel design rests on the claim that a discrete kernel size is a sub-optimal way

to specify temporal support, since k_t taps cover a different physical duration at each sampling rate. We test this directly by sweeping the standard CNN over discrete temporal kernel sizes $k_t \in 1, 3, 5, 7, 9, 11$ at every sampling rate, and overlaying the ContinuousConvLSTM model performance, which fixes the support τ once and derives k_t from Eq. (1). Figure 4 reports these sweeps across all three datasets. Two patterns hold across all three datasets: First, there is no single discrete kernel size that is optimal everywhere: the best k_t shifts with the sampling rate and differs between datasets, so a fixed-kernel model must be retuned whenever the operating rate or the task changes. Second, the ContinuousConv model, which never selects k_t manually, sits at or near the best discrete configuration at every rate. The continuous formulation therefore removes kernel size as a hyperparameter at a negligible accuracy cost, making a single multirate checkpoint feasible.

5 Conclusions

We investigated in this paper how deep-learning-based human activity recognition can operate across multiple sampling rates without requiring separate models or rate-specific architectural tuning. We introduced ContinuousConvLSTM, a multirate extension of the DeepConvLSTM architecture, in which convolutional kernels are parameterized through a continuous temporal support measured in physical time. The resulting architecture adapts its effective kernel length automatically according to the input sampling frequency while maintaining a single deployable checkpoint.

Experiments on three different benchmark datasets, WEAR, RealWorld-HAR, and WISDM-watch, demonstrated three main findings: (1) continuous kernels consistently matched or approached the performance of the best manually tuned discrete kernels across sampling rates, eliminating kernel size as a rate-dependent hyperparameter. (2) Activity-dependent sampling-rate sensitivity was observed across all datasets: many activities remained within a small performance margin of their optimum at substantially reduced sampling rates, with mostly highly dynamic activities benefitting from the original higher temporal resolution. (3) A single multirate ContinuousConvLSTM checkpoint supports operation from 6 to 50 Hz, while maintaining competitive recognition performance and substantially reducing deployment complexity, compared with maintaining separate models for each rate.

These results suggest that sampling rate can be adapted during runtime, rather than being a fixed, design-time constraint: One checkpoint can replace multiple rate-specific models, while preserving a competitive recognition accuracy. Our further investigations will now investigate integrating the proposed multirate architecture with learned rate-selection policies and evaluating the resulting energy savings on resource-constrained wearable devices.

All experiment scripts and code for reproducing figures and results of this paper can be found in the following Github repository: `https://github.com/pavan1609/ContinuousConvLSTM`.

Acknowledgments. This work was funded by the DFG Project WASEDO (grant number 506589320) and with the help of the University of Siegen's OMNI cluster.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abedin, A., Ehsanpour, M., Shi, Q., Rezatofighi, H., Ranasinghe, D.C.: Attend and Discriminate: Beyond the State-of-the-Art for Human Activity Recognition Using Wearable Sensors. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(1), Article 1 (2021). https://doi.org/10.1145/3448083
2. Bock, M., Kuehne, H., Van Laerhoven, K., Moeller, M.: WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **8**(4), Article 175 (2024). https://doi.org/10.1145/3699776
3. Cheng, W., Erfani, S., Zhang, R., Kotagiri, R.: Learning Datum-Wise Sampling Frequency for Energy-Efficient Human Activity Recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, pp. 2143–2150 (2018). https://doi.org/10.1609/aaai.v32i1.11862
4. Contoli, C., Freschi, V., Lattanzi, E.: Energy-aware human activity recognition for wearable devices: A comprehensive review. Pervasive Mob. Comput. **104**, 101976 (2024). https://doi.org/10.1016/j.pmcj.2024.101976
5. Fatima, M., Agnihotri, S., Bock, M., Gandikota, K.V., Van Laerhoven, K., Moeller, M., Keuper, M.: γ-Quant: Towards Learnable Quantization for Low-bit Pattern Recognition. In: Pattern Recognition. DAGM GCPR 2025. LNCS, vol. 16125, pp. 57–72. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-12840-9_5
6. Guan, Y., Plötz, T.: Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(2), Article 11 (2017). https://doi.org/10.1145/3090076
7. Khan, A., Hammerla, N., Mellor, S., Plötz, T.: Optimising sampling rates for accelerometer-based human activity recognition. Pattern Recognit. Lett. **73**, 33–40 (2016). https://doi.org/10.1016/j.patrec.2016.01.001
8. Liu, M., Zhao, Z., Geißler, D., Zhou, B., Suh, S., Lukowicz, P.: CoSS: Co-optimizing Sensor and Sampling Rate for Data-Efficient AI in Human Activity Recognition. arXiv preprint arXiv:2401.05426 (2024). https://arxiv.org/abs/2401.05426
9. Ordóñez, F.J., Roggen, D.: Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors **16**(1), 115 (2016). https://doi.org/10.3390/s16010115
10. Qi, X., Keally, M., Zhou, G., Li, Y., Ren, Z.: AdaSense: Adapting Sampling Rates for Activity Recognition in Body Sensor Networks. In: 2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 163–172. IEEE (2013). https://doi.org/10.1109/RTAS.2013.6531089
11. Romero, D.W., Kuzina, A., Bekkers, E.J., Tomczak, J.M., Hoogendoorn, M.: CKConv: Continuous Kernel Convolution For Sequential Data. In: 10th International Conference on Learning Representations (ICLR) (2022). https://openreview.net/forum?id=8FhxBtXSl0

12. Sztyler, T., Stuckenschmidt, H.: On-body localization of wearable devices: An investigation of position-aware activity recognition. In: 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1–9. IEEE (2016). https://doi.org/10.1109/PERCOM.2016.7456521
13. Zhou, Y., Zhao, H., Huang, Y., Riedel, T., Hefenbrock, M., Beigl, M.: TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In: Proceedings of the 2022 ACM International Symposium on Wearable Computers (ISWC), pp. 89–93. ACM (2022). https://doi.org/10.1145/3544794.3558467
14. Wang, S., Suo, S., Ma, W.-C., Pokrovsky, A., Urtasun, R.: Deep Parametric Continuous Convolutional Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2589–2597. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00274
15. Weiss, G.M., Yoneda, K., Hayajneh, T.: Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. IEEE Access **7**, 133190–133202 (2019). https://doi.org/10.1109/ACCESS.2019.2940729
16. Yang, G., Zhang, L., Bu, C., Wang, S., Wu, H., Song, A.: FreqSense: Adaptive Sampling Rates for Sensor-Based Human Activity Recognition Under Tunable Computational Budgets. IEEE J. Biomed. Health Inform. **27**(12), 5791–5802 (2023). https://doi.org/10.1109/JBHI.2023.3321639
17. Hammerla, N.Y., Halloran, S., Plötz, T.: Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1533–1540. AAAI Press (2016) https://doi.org/https://doi.org/10.48550/arXiv.1604.08880
18. Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T.S., Kjærgaard, M.B., Dey, A., Sonne, T., Jensen, M.M.: Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 127–140. ACM (2015). https://doi.org/10.1145/2809695.2809718
19. Bock, M., Hölzemann, A., Moeller, M., Van Laerhoven, K.: Improving Deep Learning for HAR with Shallow LSTMs. In: Proceedings of the 2021 ACM International Symposium on Wearable Computers (ISWC), pp. 7–12. ACM (2021). https://doi.org/10.1145/3460421.3480419
20. Romero, D.W., Bruintjes, R.-J., Tomczak, J.M., Bekkers, E.J., Hoogendoorn, M., van Gemert, J.C.: FlexConv: Continuous Kernel Convolutions with Differentiable Kernel Sizes. In: 10th International Conference on Learning Representations (ICLR) (2022). https://openreview.net/forum?id=3jooF27-0Wy
21. Kosma, C., Nikolentzos, G., Vazirgiannis, M.: Time-Parameterized Convolutional Neural Networks for Irregularly Sampled Time Series. arXiv preprint arXiv:2308.03210 (2023). https://arxiv.org/abs/2308.03210
22. Zhuzhel, V., Grabar, V., Boeva, G., Stepikin, A., Zabolotnyi, A., Zholobov, V., Ivanova, M., Orlov, M., Kireev, I.A., Burnaev, E., Rivera-Castro, R., Zaytsev, A.: COTIC: Embracing Non-Uniformity in Event Sequence Data via Multilayer Continuous Convolution. IEEE Access 13, 210741–210757 (2025). https://doi.org/10.1109/ACCESS.2025.3641785
23. Rashid, N., Demirel, B.U., Al Faruque, M.A.: AHAR: Adaptive CNN for Energy-Efficient Human Activity Recognition in Low-Power Edge Devices. IEEE Internet of Things Journal 9(15), 13041–13051 (2022). https://doi.org/10.1109/JIOT.2022.3140465

24. Bian, S., Liu, M., Rey, V.F., Geißler, D., Lukowicz, P.: TinierHAR: Towards Ultra-Lightweight Deep Learning Models for Efficient Human Activity Recognition on Edge Devices. In: Proceedings of the 2025 ACM International Symposium on Wearable Computers (ISWC '25), pp. 163–169. ACM, New York (2025). https://doi.org/10.1145/3715071.3750410
25. Liu, M., Geißler, D., Bian, S., Zhou, B., Lukowicz, P.: Assessing the Impact of Sampling Irregularity in Time Series Data: Human Activity Recognition as a Case Study. In: 2025 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), pp. 311–316 (2025). https://doi.org/10.1109/PerComWorkshops65533.2025.00083
26. Khaked, A.A., Oishi, N., Roggen, D., Lago, P.: In Shift and In Variance: Assessing the Robustness of HAR Deep Learning Models Against Variability. Sensors 25(2), 430 (2025). https://doi.org/10.3390/s25020430
27. Yamane, T., Kimura, M., Morita, M.: Effects of Sampling Frequency on Human Activity Recognition with Machine Learning Aiming at Clinical Applications. Sensors 25(12), 3780 (2025). https://doi.org/10.3390/s25123780

Appendix

A Recurrent vs. Convolutional Front-End per Activity

Table 2: Per-activity sampling-rate sensitivity on **RealWorld-HAR**, pure CNN vs. DeepConvLSTM (mean per-class F_1, LOSO over 15 subjects).

	Pure CNN				DeepConvLSTM			
Activity	6	12	25	50	6	12	25	50
Climbing Down	0.453	0.611	0.682	0.693	0.594	0.643	0.666	0.684
Climbing Up	0.598	0.673	0.708	0.727	0.675	0.714	0.721	0.722
Jumping	0.427	0.778	0.840	0.847	0.718	0.857	0.850	0.846
Lying	0.947	0.976	0.975	0.950	0.972	0.976	0.976	0.974
Running	0.599	0.800	0.810	0.811	0.728	0.802	0.790	0.799
Sitting	0.716	0.695	0.660	0.665	0.654	0.669	0.716	0.700
Standing	0.580	0.626	0.595	0.604	0.545	0.639	0.654	0.641
Walking	0.430	0.642	0.721	0.755	0.649	0.695	0.774	0.787
Macro mean	0.594	0.725	0.749	0.756	0.692	0.749	0.768	0.769
Macro F_1 optimal per-activity			0.762				0.767	

Table 3: Per-activity sampling-rate sensitivity on **WISDM-watch**, pure CNN vs. DeepConvLSTM (mean per-class F_1, LOSO over 51 subjects).

	Pure CNN			DeepConvLSTM		
Activity	5	10	20	5	10	20
Walking	0.601	0.581	0.599	0.631	0.566	0.648
Jogging	0.781	0.806	0.854	0.820	0.898	0.894
Stairs	0.368	0.395	0.468	0.404	0.453	0.491
Sitting	0.583	0.556	0.572	0.569	0.450	0.508
Standing	0.601	0.645	0.705	0.630	0.639	0.675
Typing	0.401	0.416	0.497	0.523	0.535	0.546
Brushing Teeth	0.240	0.366	0.389	0.563	0.622	0.671
Eating Soup	0.137	0.224	0.315	0.377	0.393	0.422
Eating Chips	0.162	0.195	0.220	0.213	0.248	0.267
Eating Pasta	0.194	0.234	0.275	0.350	0.354	0.359
Drinking	0.245	0.349	0.414	0.354	0.376	0.403
Eating Sandwich	0.064	0.106	0.119	0.115	0.128	0.161
Kicking	0.494	0.506	0.557	0.582	0.574	0.562
Catch	0.442	0.449	0.594	0.467	0.531	0.559
Dribbling	0.438	0.516	0.524	0.580	0.649	0.685
Writing	0.504	0.482	0.493	0.570	0.606	0.637
Clapping	0.493	0.541	0.599	0.588	0.620	0.638
Folding Clothes	0.268	0.335	0.365	0.414	0.479	0.516
Macro mean	0.390	0.428	0.476	0.486	0.507	0.536
Macro F_1 optimal per-activity		0.476			0.536	

These tables expand the main-text comparison of the DeepConvLSTM against a pure CNN (LSTM replaced by temporal average pooling) to the activity level. The recurrent stage helps most at the lowest rate, where the discrete kernel shrinks to a single tap: the macro-F_1 gap at the lowest rate is +0.038 on WEAR , +0.086 on WISDM-Watch, and +0.098 on RealWorld-HAR.

B Single-Checkpoint ContinuousConvLSTM per Activity

The single multirate checkpoint preserves the activity-dependent rate structure seen with the per-rate baselines. Aggregate macro-F_1 is

0.715/0.689/0.689/0.678	for WEAR at 50/25/12/6 Hz,
0.767/0.733/0.723/0.626	for RealWorld-HAR at 50/25/12/6 Hz,
0.536/0.503/0.467	for WISDM-watch at 20/10/5 Hz.

This shows that the shared checkpoint does not simply collapse to a single globally preferred rate. Instead, several activities remain close to their best score at lower sampling rates, while others still benefit from the highest available rate. The per-activity results below therefore provide a more detailed view of where the single-checkpoint model preserves low-rate robustness and where higher temporal resolution is still useful.

Table 4: Per-activity sampling-rate sensitivity for the single-checkpoint ContinuousConvLSTM across WEAR, RealWorld-HAR, and WISDM-watch. Values are mean per-class F_1 across LOSO. For each activity, the best values are underlined, with close scores within 0.02 absolute F_1 of the best performance in green. The activity-optimal macro F_1 is computed by selecting the lowest sampling rate whose score lies within 0.02 absolute F_1 of the best score for each activity.

RealWorld-HAR

Activity	6 Hz	12 Hz	25 Hz	50 Hz	Opt.
Walking	0.505	0.615	0.638	0.751	50 Hz
Running	0.635	0.782	0.773	0.783	12 Hz
Sitting	0.676	0.641	0.649	0.675	6 Hz
Standing	0.578	0.629	0.630	0.621	12 Hz
Lying	0.967	0.973	0.968	0.978	6 Hz
Jumping	0.575	0.844	0.865	0.881	25 Hz
Climbing Up	0.638	0.692	0.711	0.732	50 Hz
Climbing Down	0.482	0.605	0.633	0.714	50 Hz
Macro mean	0.632	0.723	0.733	0.767	
Macro F_1 at activity-optimal rates: 0.764					

WEAR

Activity	6 Hz	12 Hz	25 Hz	50 Hz	Opt.
null	0.647	0.680	0.690	0.683	12 Hz
bench-dips	0.615	0.581	0.565	0.641	50 Hz
burpees	0.551	0.591	0.603	0.584	12 Hz
jogging	0.815	0.834	0.831	0.863	50 Hz
jogging (butt-kicks)	0.740	0.785	0.776	0.802	12 Hz
jogging (rotating arms)	0.697	0.772	0.774	0.792	12 Hz
jogging (sidesteps)	0.858	0.868	0.871	0.885	12 Hz
jogging (skipping)	0.767	0.831	0.828	0.852	50 Hz
lunges	0.496	0.515	0.536	0.542	25 Hz
lunges (complex)	0.622	0.621	0.628	0.641	6 Hz
push-ups	0.510	0.506	0.466	0.576	50 Hz
push-ups (complex)	0.579	0.578	0.572	0.597	6 Hz
sit-ups	0.660	0.635	0.631	0.662	6 Hz
sit-ups (complex)	0.675	0.676	0.675	0.662	6 Hz
stretching (hamstrings)	0.691	0.677	0.674	0.728	50 Hz
stretching (lumbar rotation)	0.851	0.845	0.853	0.871	6 Hz
stretching (lunging)	0.677	0.694	0.701	0.700	12 Hz
stretching (shoulders)	0.650	0.647	0.660	0.669	6 Hz
stretching (triceps)	0.771	0.759	0.760	0.827	50 Hz
Macro mean	0.677	0.689	0.689	0.715	
Macro F_1 at activity-optimal rates: 0.708					

WISDM-watch

Activity	5 Hz	10 Hz	20 Hz	Opt.
Walking	0.511	0.566	0.648	20 Hz
Jogging	0.820	0.881	0.894	10 Hz
Stairs	0.404	0.453	0.491	20 Hz
Sitting	0.469	0.450	0.508	20 Hz
Standing	0.630	0.639	0.675	20 Hz
Typing	0.523	0.535	0.546	10 Hz
Brushing Teeth	0.563	0.622	0.671	20 Hz
Eating Soup	0.377	0.393	0.422	20 Hz
Eating Chips	0.213	0.245	0.267	20 Hz
Eating Pasta	0.350	0.354	0.359	5 Hz
Drinking	0.354	0.376	0.403	20 Hz
Eating Sandwich	0.115	0.128	0.161	20 Hz
Kicking	0.462	0.534	0.562	20 Hz
Catch	0.467	0.531	0.559	20 Hz
Dribbling	0.580	0.649	0.685	20 Hz
Writing	0.570	0.606	0.637	20 Hz
Clapping	0.588	0.620	0.638	10 Hz
Folding Clothes	0.414	0.479	0.516	20 Hz
Macro mean	0.467	0.503	0.536	
Macro F_1 at activity-optimal rates: 0.533				

C Role of Per-Rate Branches

Table 5: Design space of the evaluated models, with macro-F_1 at each model's best sampling rate on the three benchmarks. $\mathcal{F}$ is the set of supported rates; "kernel tuning" indicates whether a discrete kernel size must be chosen manually per rate. Standard DeepConvLSTM reports the best score over separately trained per-rate checkpoints. Continuous single-branch is trained at a single rate only and does not transfer to other rates (Appendix C). ContinuousConvLSTM (ours) is the only single-checkpoint, tuning-free model, matching the per-rate DeepConvLSTM within 0.005 macro-F_1.

Model	Branches	Kernel Rate	Tuning	Single Ckpt.	Macro-F_1		
					WEAR	RWHAR	WISDM
Standard DeepConvLSTM	1	fixed k_t	per rate	no	0.720	0.769	0.536
Standard multi-branch	$\lvert\mathcal{F}\rvert$	fixed k_t per branch	per rate	yes	0.665	0.701	0.443
Continuous (ours)	$\lvert\mathcal{F}\rvert$	$\tau \cdot f_s$	none	yes	**0.708**	**0.764**	**0.533**
Continuous single-branch	1	$\tau \cdot f_s$	none	no	0.731	0.772	0.541