Device-Free Fall Detection from LiDAR Point Clouds Using a Three-Keypoint Skeleton: A Subject-Independent Comparison of Simple and Graph-Convolutional Classifiers

Noel D'Avis[1 [0009−0008−8351−4352]] and Silvia Faquiri[1 [0000−0001−8700−3543]]

RheinMain University of Applied Sciences, Wiesbaden, Germany

Abstract. Fall detection for elderly and care-dependent individuals is a central function of ambient assisted living, yet camera-based approaches raise privacy concerns and degrade under poor illumination and occlusion. We present a device-free fall detection system that operates entirely on LiDAR point clouds and reduces each detected person to a compact three-keypoint skeleton, so that no visual appearance is retained for classification. A real-time pipeline performs background subtraction, voxel downsampling, ground-plane removal, density-based clustering, and multi-object tracking, after which a robust geometric estimator derives the three keypoints together with their velocities and accelerations. On a self-collected dataset of 294 clips from three subjects, we compare a range of classifiers on this representation, from a single-feature height threshold and a shallow classifier on the flattened coordinates to a compact two-stream adaptive graph convolutional network of 207,680 parameters, under both a conventional clip-level protocol and a subject-independent leave-one-subject-out protocol. Under the clip-level protocol all learned models perform comparably, reaching a fall F_1 of approximately 0.98. Under subject-independent evaluation the ranking changes: a random forest on absolute keypoint coordinates attains the highest macro-averaged fall F_1 (0.952 ± 0.003) and is the most consistent across the held-out subjects, outperforming the graph-convolutional network (0.876 ± 0.032). This macro ranking is not uniform across folds, being dominated by a single perfectly separated subject, with logistic regression surpassing the random forest on the other two, so we read these cross-subject results as indicative rather than conclusive. All models run in real time on a CPU, with per-clip inference well below the observation window. The pipeline runs online on the live sensor stream, but the results reported here are measured offline on pre-segmented single-person clips, so they are proof-of-concept and stream-level behavior under continuous operation remains to be quantified.

Keywords: Fall detection · LiDAR · Point cloud · Skeleton-based action recognition · Device-free sensing · Subject-independent evaluation · Ambient assisted living

1 Introduction

Falls are a leading cause of injury and loss of independence among older adults, and timely detection of a fall is a prerequisite for rapid intervention in home and care environments [11]. Automated fall detection has therefore become a central function of ambient assisted living systems. Camera-based recognition achieves strong performance under controlled conditions, but the continuous capture of identifiable visual information in private living spaces raises serious privacy concerns, and camera systems remain sensitive to illumination changes, shadows, and occlusion [19,5]. These limitations have motivated device-free sensing modalities that do not require the monitored person to wear or carry a device and that do not record identifiable imagery. LiDAR sensing is attractive in this context: it captures three-dimensional point clouds that are invariant to ambient lighting and that describe spatial structure directly, while not recording photometric appearance. At the same time, point clouds are sparse, irregularly sampled, and unordered, which complicates the direct application of standard convolutional architectures and increases the risk of overfitting on the limited datasets typical of healthcare settings [20,9].

In this work we pursue a representation that is both privacy-preserving and compact. Rather than classifying raw point clouds or dense voxel grids, we reduce each detected person to a three-keypoint skeleton consisting of the head, torso, and feet, and we classify the temporal evolution of this skeleton. The three keypoints retain the information most relevant to falls, namely the vertical configuration of the body and its dynamics, while discarding the underlying point cloud, so that no visual appearance is available downstream. Given this low-dimensional representation, it is not obvious how much model complexity fall classification actually requires. We therefore compare a range of classifiers on the same three-keypoint sequences, from a single-feature height threshold and a shallow classifier on the flattened coordinates to a compact two-stream adaptive graph convolutional network, an architecture family originally developed for skeleton-based human action recognition [24,21] and here adapted to a three-node graph and to LiDAR-derived keypoints. A recurring concern in human activity recognition is that clip-level evaluation, in which clips from all subjects appear in both training and test sets, can overstate the performance that a system achieves on people it has never seen. We therefore evaluate every classifier not only under a conventional clip-level split but also under a subject-independent leave-one-subject-out (LOSO) protocol, and we report the difference between the two. For the graph-convolutional model we additionally isolate two design choices, the presence of an explicit motion stream and the choice of coordinate frame, and we characterize the computational cost of each model for real-time deployment.

The contributions of this work are the following. We present a device-free LiDAR fall detection pipeline that reduces each person to a three-keypoint skeleton, retains no raw point cloud or visual appearance for classification, and operates as a continuous real-time system. On this representation we compare classifiers of widely differing complexity, from a single-feature height threshold

and a shallow classifier on the flattened coordinates to a compact two-stream adaptive graph convolutional network, under both a clip-level and a subject-independent protocol over 20 random seeds. Our main empirical finding is that, while all learned models perform comparably at the clip level, under subject-independent evaluation a random forest on absolute keypoint coordinates is the most consistent across subjects and a single height threshold already matches the graph-convolutional network, whose additional capacity does not improve subject-independent performance; we trace this to the absolute vertical coordinate, which encodes height above the ground directly. Because this cross-subject comparison rests on only three held-out subjects, with a macro ranking dominated by one perfectly separated fold, we present it as indicative rather than definitive. On this basis we recommend a shallow classifier such as a random forest on absolute coordinates as a simple and robust detector for this representation, and we show that a clip-level protocol alone would have concealed these differences. The remainder of this paper reviews related work, describes the system and method, presents the experimental comparison under both protocols together with the efficiency analysis, discusses the findings and limitations, and concludes.

Availability. The anonymized three-keypoint dataset (nine position coordinates per frame; no raw point clouds are retained) is available from the authors on reasonable request, subject to a signed data use agreement, in line with the consent terms of the recorded participants.

2 Related Work

Automated fall detection has been studied across three broad sensing paradigms: wearable inertial sensors, vision-based systems, and ambient device-free sensing [15]. Wearable accelerometers and gyroscopes, frequently integrated into smartphones or smartwatches, detect falls reliably but depend on the monitored person consistently carrying or wearing the device, which is often not the case for older adults [11]. Vision-based systems achieve high accuracy under controlled conditions but continuously capture identifiable imagery and remain sensitive to illumination, shadows, and occlusion, which limits their acceptance in private living spaces [5,8,19]. These constraints have motivated ambient, device-free modalities that require no worn device and that do not record photometric appearance.

LiDAR has been applied to indoor activity and fall detection in several forms. Two-dimensional LiDAR has been used for activity detection in healthcare and monitoring settings [3], including robot-mounted deployments [1] and multi-sensor configurations that combine several planar scanners to cover a room [2]. For fall detection specifically, low-cost LiDAR monitoring systems have employed rule- and state-machine-based decision logic [17]. Because such rule-based systems already achieve competitive results from simple geometric criteria, they raise the question of how much model complexity a learned skeleton-based detector actually needs, which motivates the comparison against simple baselines

reported here. These systems establish device-free LiDAR as a practical sensing modality for in-home fall detection, but they generally operate on geometric heuristics or on comparatively dense representations rather than on an explicit, low-dimensional skeleton.

A parallel line of work estimates human pose from LiDAR. Heterogeneous systems fuse LiDAR with inertial sensors to reconstruct three-dimensional pose in real time [16], while recent learning-based methods estimate skeletons directly from sparse point clouds, including transformer-based pose estimation for non-repetitive scanning LiDAR [12] and approaches that adapt motion-aware pose models to LiDAR input [26]; related work segments body parts from standalone 3D LiDAR for gesture identification [18]. These methods typically target dense skeletons with many joints, whereas the present work deliberately reduces each person to a three-keypoint representation in order to keep the downstream classifier compact and to discard appearance information.

Once a skeleton is available, recognizing actions from its temporal evolution is commonly addressed with graph convolutional networks, which represent the body as a graph of joints. Spatial-temporal graph convolutional networks combine graph convolution over a fixed skeleton topology with temporal convolution [24], and two-stream adaptive graph convolutional networks extend this design with a learnable adjacency component and a separate motion stream derived from joint velocities [21]. Multimodal variants additionally fuse skeleton and RGB streams [8]. These architectures were developed for dense skeletons of roughly twenty to twenty-five joints obtained from motion capture or RGB-D sensors; in this work we adapt the adaptive graph-convolutional and two-stream principles to a three-node graph derived from LiDAR, comprising the head, torso, and feet.

A methodological concern that cuts across these modalities is how generalization is measured. Many activity- and fall-recognition systems are evaluated on pre-segmented clips, and surveys of continuous activity recognition caution that accuracies obtained on segmented benchmarks can overstate the performance attainable under realistic conditions [23]. In the fall-detection literature, subject-independent evaluation has been adopted to assess transfer to unseen people; leave-one-subject-out validation has been used with UWB-radar fall detection to estimate cross-subject generalization [14,10]. Motivated by these observations, we report both a conventional clip-level protocol and a subject-independent leave-one-subject-out protocol so as to make the difference between the two explicit.

Finally, this work relates to our own prior study, in which we recognized human activities from LiDAR point clouds using a compact three-dimensional convolutional network operating on dense voxel grids, evaluated on the public HmPEAR dataset [4,13]; voxel-based action recognition has likewise been explored in other domains [25]. The present work differs in both representation and task: rather than classifying voxelized point clouds across many activity classes, it reduces each person to a three-keypoint skeleton and addresses binary fall detection on a self-collected dataset, replacing the voxel-based convolutional

network with a graph convolutional classifier and reducing the retained information per frame from a dense occupancy grid to nine coordinates.

3 System and Method

The system transforms a stream of LiDAR point clouds into per-person fall decisions through a sequence of stages: point cloud preprocessing and person detection, multi-object tracking, three-keypoint skeleton extraction, and classification of the resulting three-keypoint skeleton sequence. Figure 1 gives an overview of the complete pipeline, and Figure 2 shows the sensing setup. Data are acquired with a Velodyne VLP-16 LiDAR sensor, which provides a horizontal field of view of 360° and a vertical field of view of ±15° at a wavelength of 905 nm. The sensor is configured at a rotation speed of 600 revolutions per minute, corresponding to a nominal frame rate of 10 Hz, in dual-return mode. Because of the limited vertical field of view, the sensor is mounted at approximately half the body height of the monitored person: for a person of height H, the sensor height is $h_S = H/2$, and the recommended sensor-to-person distance follows from the vertical opening angle as $D = h_S / \tan(15°)$. As a concrete reference point for the target population, the average height of persons aged 75 and over in Germany is $H \approx 1.68$ m [22], which yields a sensor height of $h_S \approx 0.84$ m and a recommended distance of $D \approx 3.13$ m; at smaller distances a falling person may partially leave the vertical measurement range. Each point cloud frame stores per-point Cartesian coordinates and an associated range value together with a per-point timestamp, from which the duration of each frame is computed, so that the time difference between successive frames is derived from the data rather than assumed constant.

The processing pipeline transforms each raw frame into a set of per-person point clusters. A static background, recorded once from an empty scene, is subtracted by removing points within 0.1 m of any background point. The remaining points are downsampled with a voxel grid of edge length 0.05 m to reduce density variation. The dominant ground plane is removed using random sample consensus with a distance threshold of 0.06 m and a minimal sample size of three points [7]. The remaining points are clustered with density-based spatial clustering, using a neighborhood radius of 0.25 m and a minimum of 20 points per cluster [6], so that each sufficiently dense cluster corresponds to a candidate person. Candidate clusters are then associated across frames by a greedy nearest-neighbor tracker with a constant-velocity motion model; the tracker gates associations by distance, smooths velocity estimates with an exponential factor of 0.6, suppresses persistently static clusters as environmental objects, and re-identifies lost tracks within 2.5 m and 30 frames, the latter chosen to bridge the temporary loss of a track during a fall and a subsequent recovery. Each maintained track yields a temporally consistent sequence of point clusters for one person.

For each person and frame, three keypoints, head, torso, and feet, are estimated from the corresponding point cluster, as illustrated in Figure 3. The torso keypoint is obtained by a robust principal-component estimator: after percentile-

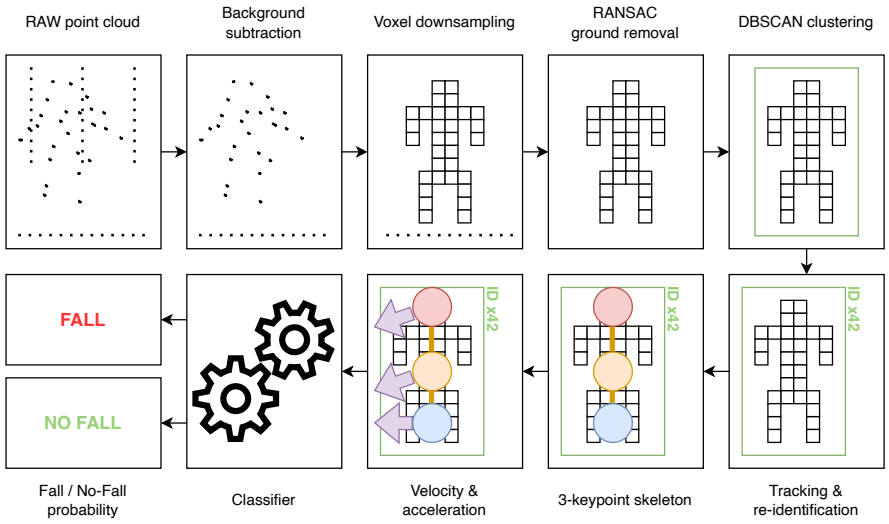


Fig. 1. Overview of the device-free fall detection pipeline. A raw point cloud from the VLP-16 sensor is reduced to per-person clusters through background subtraction, voxel downsampling, RANSAC ground-plane removal, and DBSCAN clustering. Each tracked person, kept consistent across frames by re-identification, is reduced to a three-keypoint skeleton (head, torso, feet) together with per-keypoint velocity and acceleration, which a classifier maps to a fall or no-fall decision. We compare several classifiers for this final stage (Section 4) and recommend a random forest on absolute coordinates.

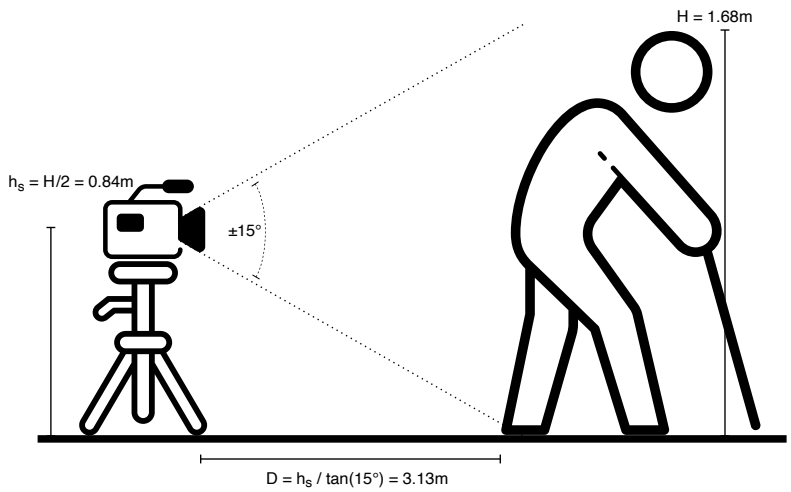


Fig. 2. Recommended sensing setup. The VLP-16 is mounted at approximately half the body height, $h_S = H/2$, where H is the height of the monitored person. Its vertical field of view of $\pm 15^\circ$ determines the recommended sensor-to-person distance $D = h_S / \tan(15^\circ)$. At smaller distances, a falling person may partially leave the vertical measurement range.

based outlier trimming, the body's longitudinal axis is identified as the principal axis of the trimmed points, the points are projected onto this axis, and the torso is taken as the median of the points lying in the central band of the projection; this procedure is insensitive to the asymmetric point density between the upper and lower body. The head and feet keypoints are determined as the two extremal endpoints of the body, with the choice of axis depending on posture: when the vertical extent of the cluster is at least 0.5 m, the person is treated as upright and the endpoints are computed along the vertical axis, with the head taken as the mean of the highest five percent of points and the feet as the mean of the lowest thirteen percent, the asymmetric fractions reflecting that the head occupies few points while the foot region is broader; when the vertical extent is below 0.5 m, the person is treated as non-upright and the endpoints are computed along the horizontal axis of greatest extent. The 0.5 m switch is not a sensitive parameter: a standing or walking person spans well over a metre vertically and a person lying on the floor spans only a few decimetres, so the threshold sits in the wide gap between the two regimes and its exact value has no effect on the axis selected for realistic upright or fallen poses. To prevent the head and feet labels from swapping between consecutive frames, the assignment that minimizes the total displacement with respect to the previous head and feet positions is selected, and a swap is accepted only when it is at least 0.05 m more consistent than retaining the current assignment. Coordinates are expressed in

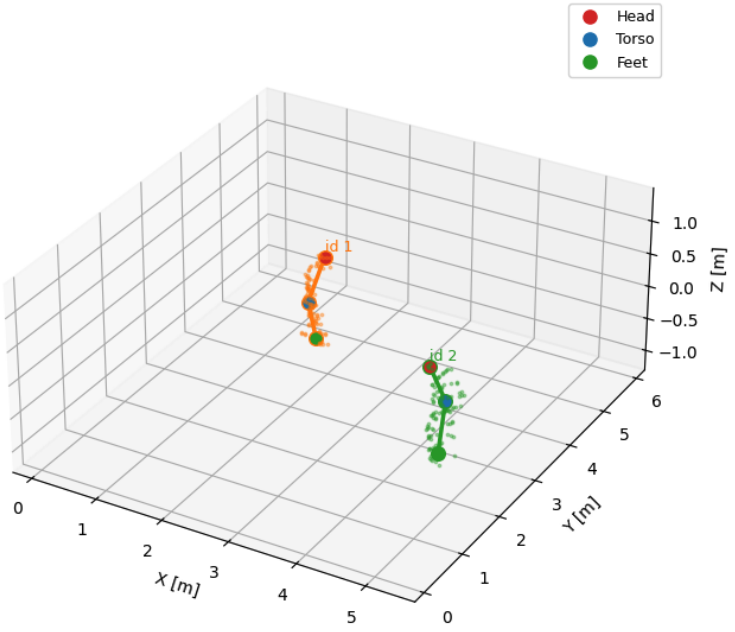


Fig. 3. Three-keypoint skeletons extracted from segmented point clouds for two simultaneously tracked persons (id 1 and id 2). For each tracked person, the foreground points remaining after background subtraction and RANSAC ground-plane removal are reduced to a head (red), torso (blue), and feet (green) keypoint by the robust geometric estimator of Section 3; the head–torso and torso–feet edges form the three-node skeleton graph used by the graph-convolutional classifier. Coordinates are given in metres in the sensor frame, with z measured relative to the sensor origin (negative below the sensor).

the sensor frame, with the vertical axis measured relative to the sensor origin, so that a lower absolute value corresponds to a position closer to the floor.

For each keypoint $\mathbf{p}(t)$ the velocity and acceleration are computed by finite differences, where Δt is the data-derived inter-frame interval,

$$\mathbf{v}(t) = \frac{\mathbf{p}(t) - \mathbf{p}(t-1)}{\Delta t}, \qquad \mathbf{a}(t) = \frac{\mathbf{v}(t) - \mathbf{v}(t-1)}{\Delta t}. \tag{1}$$

Velocities and accelerations are set to zero for the first frame of a track and after a tracking gap, and a previous state is reused only when the person was updated within the two preceding processing cycles, which prevents spurious velocity spikes after an interruption. Each frame therefore yields, per keypoint, three position, three velocity, and three acceleration components, so that the nine stored position coordinates (three keypoints in three dimensions) are the only data retained and released, while the derived kinematics bring the classifier input to 27 features per frame (459 per clip over the $T = 17$ frames).

The most complex of the classifiers we compare is a compact two-stream adaptive graph convolutional network, described here; the simpler baselines it is compared against, a height threshold and a shallow classifier on the same per-frame features, are specified in Section 4. The skeleton is modeled as an undirected graph with three nodes, head, torso, and feet, with head–torso and torso–feet edges and self-loops. Let A be its adjacency matrix including self-loops and D the degree matrix; graph convolution uses the symmetric normalization $\hat{A} = D^{-1/2} A D^{-1/2}$, and, following the adaptive design [21], each layer augments the fixed normalized adjacency with a learnable additive component P initialized near zero, so that a spatial graph-convolution layer with input features $\mathbf{X}^{(l)}$ and weights $W^{(l)}$ computes

$$\mathbf{X}^{(l+1)} = \sigma\Big((\hat{A} + P)\, \mathbf{X}^{(l)}\, W^{(l)} \Big) . \tag{2}$$

The network has two streams of identical structure on the same graph: the joint stream receives the three position components per node and the motion stream the six velocity and acceleration components. Each stream applies input normalization, four spatial-temporal graph-convolution blocks with 32, 32, 64, and 64 channels, each combining an adaptive graph convolution following Equation (2) with a temporal convolution, batch normalization, a residual connection, and dropout, and a global average pooling layer whose output a linear layer maps to two class logits. Denoting by $\mathbf{l}_{\text{joint}}$ and $\mathbf{l}_{\text{motion}}$ the two streams' logits, they are fused by averaging before a softmax produces the fall and no-fall probabilities,

$$\mathbf{z} = \operatorname{softmax}\big(\tfrac{1}{2} \left(\mathbf{l}_{\text{joint}} + \mathbf{l}_{\text{motion}} \right) \big) . \tag{3}$$

Removing the motion stream leaves the single-stream joint-only variant, and replacing torso-relative with absolute joint coordinates gives the absolute variant; these four combinations form the graph-convolutional configurations compared in Section 4. Each input sequence is resampled to a fixed length of $T = 17$ frames by centered cropping or end-padding. The system is implemented as a continuous real-time pipeline in which incoming frames are written to a thread-safe ring buffer and classification operates on a sliding window over the buffered frames once at least ten frames are available; the processing stage performs detection, tracking, skeleton extraction, and classification per person and emits per-person fall probabilities. This streaming pipeline was deployed and run online on the live VLP-16 sensor stream, ingesting point clouds in real time and emitting per-person fall probabilities to a live monitoring dashboard at the sensor frame rate. We report this online operation only qualitatively; the quantitative evaluation in this paper is conducted offline at the clip level on recorded sequences.

4 Experimental Evaluation

We evaluate the system along three axes: the classification performance of all compared classifiers under two protocols, an ablation within the graph-convolutional family, and computational cost. All classification metrics refer to the fall class,

Table 1. Clip distribution of the recorded dataset (three subjects, two classes).

Subject	Fall	No-Fall	Total
P1	52	79	131
P2	13	34	47
P3	83	33	116
Total	148	146	294

and every configuration is retrained over 20 random seeds with mean and standard deviation reported across seeds. The dataset was recorded on separate days across two different environments (a laboratory room and a furnished private home) and comprises both non-fall and fall scenarios: the non-fall recordings contain a single person walking, a single person running, and a range of everyday non-fall movements such as sitting down, tying shoes, and pouring a glass of water, while the fall scenarios comprise a single person falling, a single person falling while using a walking aid (walker), and multi-person scenes in which one or more persons fall. Each recorded sequence is segmented into non-overlapping clips of varying length, later resampled to the fixed length $T = 17$ described in Section 3, and each clip is labeled at the clip level as fall or no-fall, where a clip is labeled as a fall when at least one of its frames carries a fall annotation; each clip is the skeleton sequence of a single tracked person, so that in a multi-person scene a fall clip corresponds to the person who falls, and Figure 4 illustrates a representative fall and non-fall sequence. The dataset contains 294 clips from three subjects, denoted P1–P3, whose per-subject and per-class distribution is reported in Table 1. The three participants were aged 25–29 years, 1.70–1.85 m tall, and 70–120 kg; all falls were instructed and performed onto mats. The dataset is unbalanced at the subject level: the number of clips per subject ranges from 47 to 131, and the fall to no-fall ratio differs between subjects, which is relevant to the subject-independent protocol below, where each held-out subject induces a different test-set class balance.

Ethical considerations and data handling. The dataset was collected within a supervised student project at the authors' institution; the recorded participants were the project members themselves, who consented to the recording and to the use of their data for this research, and no members of the public or otherwise external persons were recorded. All participants were healthy consenting adults performing instructed movements onto crash mats under mutual supervision. The raw LiDAR point clouds were used only for keypoint extraction and were deleted afterwards; the retained and released data consist solely of the nine-coordinate-per-frame keypoint sequences, from which visual appearance cannot be reconstructed.

We report results under two protocols. Under the clip-level protocol, clips are split into training, validation, and test sets by a 70/15/15 split without regard to subject identity, so that clips from all three subjects may appear in every split,

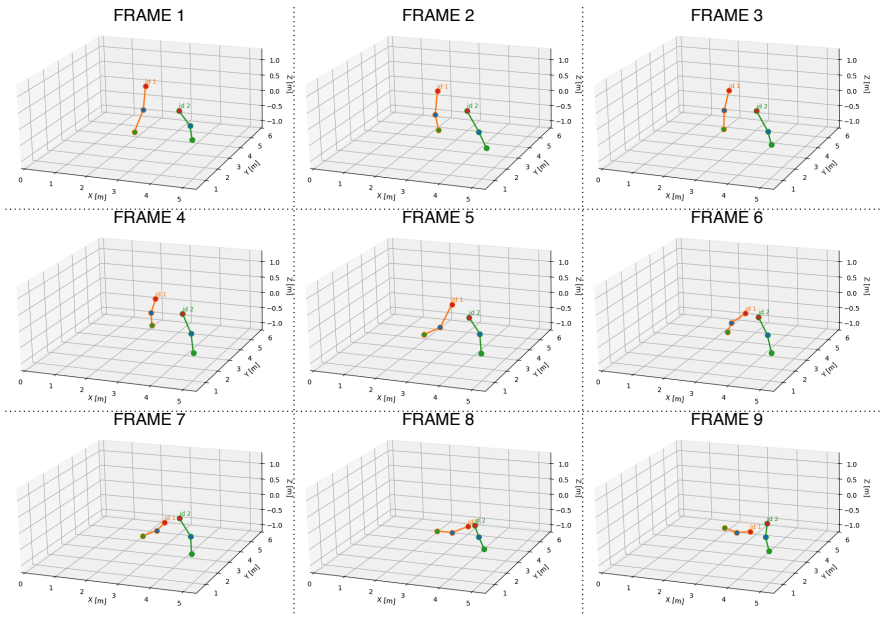


Fig. 4. Nine-frame sub-sequence extracted from a stream. One skeleton (orange) falls starting at frame 5, while the other (green) remains upright.

which reflects the conventional clip-level evaluation setting; no class weighting was applied during training, as the dataset is approximately balanced overall (148 fall, 146 no-fall clips). Under the cross-subject protocol, leave-one-subject-out cross-validation holds out all clips of one subject for testing and trains on the remaining two subjects, repeating the procedure for each subject and macro-averaging the per-fold metrics, which measures transfer to a previously unseen person; for model selection within each cross-subject fold a class-stratified validation set is drawn from the two training subjects, while the held-out subject is used only for testing. All models are trained with the AdamW optimizer at an initial learning rate of 10^{-3}, a weight decay of 10^{-4}, and a cosine decay schedule, with a batch size of 16 for up to 60 epochs; early stopping selects the epoch with the highest validation fall F_1 score with a patience of 10 epochs, normalization statistics are computed on the training split only, and each input sequence has length $T = 17$.

We compare seven classifiers on the same three-keypoint sequences, spanning a wide range of model complexity. The simplest is a height-threshold rule that flags a fall when the absolute vertical coordinate of the head remains below a threshold for a number of consecutive frames, its two parameters being fitted on the training clips of each fold. A shallow classifier, evaluated both as logistic regression and as a random forest, operates on the per-clip feature vector obtained

by flattening the resampled $T = 17$ sequence of 27 per-frame features in absolute coordinates. The remaining four configurations are graph-convolutional: the two-stream model with torso-relative and with absolute joint coordinates, and the two corresponding joint-only models that remove the motion stream. Table 2 reports all seven under both protocols. Under the clip-level protocol the learned models cluster near a fall F_1 of 0.98, differing by no more than seed-level variability (two-stream absolute 0.980 ± 0.025, logistic regression 0.978, random forest 0.977, joint-only absolute 0.956 ± 0.020), while the height threshold trails at 0.824; the height threshold and the shallow classifiers show no seed variance at this precision under the fixed clip-level split.

The shallow classifiers use scikit-learn implementations with the following settings. The random forest uses 300 trees with the Gini criterion and unbounded depth, drawing $\sqrt{d}$ features per split (default), bootstrap sampling, and no class weighting; the only non-default setting is the number of trees. Logistic regression uses the default L2-regularized `lbfgs` solver with inverse regularization strength $C = 1$, no class weighting, and up to 5000 iterations. Both operate on the 459-dimensional per-clip vector formed by flattening the $T = 17$ sequence of 27 absolute-coordinate features, standardized with statistics computed on the training split only. The height threshold has two parameters, the head-coordinate threshold τ and the minimum number of consecutive frames k, selected by grid search over $k \in \{1, \ldots, 10\}$ and 41 quantile-spaced values of τ to maximize training fall F_1; all three baselines are fitted per fold on the training subjects only.

Under the subject-independent protocol the ordering changes markedly. The random forest attains the highest fall F_1 at 0.952 ± 0.003, followed by logistic regression at 0.915 ± 0.010; the absolute-coordinate graph-convolutional models (two-stream 0.876 ± 0.032, joint-only 0.882 ± 0.044) and the height threshold (0.896 ± 0.009) form a lower cluster, and the torso-relative graph-convolutional models trail (two-stream 0.831 ± 0.043, joint-only 0.804 ± 0.039). Throughout, the combined standard deviation of two configurations denotes the root of the sum of their squared seed-level standard deviations, $\sigma_{\text{comb}} = \sqrt{\sigma_1^2 + \sigma_2^2}$, a deliberately conservative estimate of the standard deviation between two single-seed observations rather than the substantially smaller standard error of the mean, reflecting the run-to-run variability observed on this hardware and software platform. We emphasize that σ_{comb} quantifies only reproducibility under reseeding on a fixed subject split and does not measure whether a ranking transfers to an unseen person; the uncertainty relevant to cross-subject transfer is the between-subject dispersion, which is far larger. The recommended random forest, for instance, varies by only ± 0.003 across seeds yet ranges from 0.919 to 1.000 in fall F_1 across the three held-out subjects (Table 3). We therefore report the seed-level standard deviations as a reproducibility measure only and read the cross-subject comparisons below as descriptive observations on three subjects rather than as tests of statistical significance. On this three-subject sample the random forest exceeds the strongest graph-convolutional model by 0.076 and logistic regression by 0.039, both larger than the corresponding run-to-run stan-

Table 2. Fall-detection performance of all compared classifiers under two evaluation protocols: clip-level (in-subject) and leave-one-subject-out (LOSO, macro-averaged over three subjects). Values are mean ± std over 20 random seeds; all metrics refer to the fall class. The height threshold and the logistic-regression classifier are deterministic under the fixed clip-level split; the seed-dependent random forest varies under LOSO (±.003) but rounds to zero at the clip level at three-decimal precision. Best fall F_1 per protocol in bold.

Model	Acc.	Prec.	Rec.	F1
Clip-level				
Height threshold	.795 ± .000	.724 ± .000	.955 ± .000	.824 ± .000
Shallow, LogReg	.977 ± .000	.957 ± .000	1.000 ± .000	.978 ± .000
Shallow, Random Forest	.977 ± .000	1.000 ± .000	.955 ± .000	.977 ± .000
GCN, Two-Stream (rel.)	.908 ± .048	.900 ± .059	.925 ± .098	.907 ± .058
GCN, Joint-only (rel.)	.856 ± .063	.837 ± .092	.904 ± .093	.863 ± .060
GCN, Two-Stream (abs.)	.980 ± .027	.969 ± .047	.993 ± .016	**.980 ± .025**
GCN, Joint-only (abs.)	.955 ± .022	.943 ± .044	.970 ± .022	.956 ± .020
LOSO (cross-subject)				
Height threshold	.884 ± .006	.846 ± .005	.961 ± .020	.896 ± .009
Shallow, LogReg	.934 ± .008	.927 ± .008	.907 ± .015	.915 ± .010
Shallow, Random Forest	.950 ± .003	.970 ± .000	.936 ± .005	**.952 ± .003**
GCN, Two-Stream (rel.)	.835 ± .045	.845 ± .052	.864 ± .055	.831 ± .043
GCN, Joint-only (rel.)	.809 ± .030	.853 ± .034	.814 ± .081	.804 ± .039
GCN, Two-Stream (abs.)	.905 ± .019	.932 ± .034	.841 ± .055	.876 ± .032
GCN, Joint-only (abs.)	.913 ± .025	.921 ± .040	.861 ± .061	.882 ± .044

dard deviations ($\sigma_{\text{comb}} = 0.032$ and 0.033) and hence stable under reseeding, whereas the height threshold's margin of 0.020 ($\sigma_{\text{comb}} = 0.033$) lies within run-to-run variability, so on these three subjects a single height feature is on par with the graph-convolutional network. Within the graph-convolutional family the motion stream changes fall F_1 by less than σ_{comb} under both coordinate frames, so any motion-stream effect is smaller than run-to-run noise, while absolute coordinates exceed torso-relative ones by more than run-to-run noise for the joint-only model (0.078 vs. $\sigma_{\text{comb}} = 0.059$) and favorably but within it for the two-stream model (0.045 vs. $\sigma_{\text{comb}} = 0.054$).

Table 3 reports the per-subject breakdown of the recommended detector under the cross-subject protocol. In contrast to the graph-convolutional models, the random forest is stable across folds: the held-out subject with the fewest clips, P2 with 47 clips, is separated perfectly, while the two larger held-out subjects yield fall F_1 scores of 0.919 and 0.936 with small seed variance. Figure 5 compares the aggregated confusion matrices of the strongest graph-convolutional configuration and the recommended random forest, each pooled over all held-out subjects and all 20 seeds. The random forest reduces the false-positive rate from 0.069 to 0.041 while also raising fall recall from 0.871 to 0.916, so it improves on the graph-convolutional model in both error directions simultaneously. Because

Table 3. Per-subject LOSO results for the recommended detector, a random forest on absolute keypoint coordinates. Each row holds out one subject for testing; mean ± std over 20 seeds. Subject identifiers are anonymized.

Held-out subject	Acc.	Prec.	Rec.	F_1
P1	$.939 \pm .006$	$.958 \pm .001$	$.884 \pm .014$	$.919 \pm .008$
P2	$1.000 \pm .000$	$1.000 \pm .000$	$1.000 \pm .000$	$1.000 \pm .000$
P3	$.910 \pm .006$	$.950 \pm .000$	$.923 \pm .008$	$.936 \pm .004$
Macro avg.	$.950 \pm .003$	$.970 \pm .000$	$.936 \pm .005$	$.952 \pm .003$

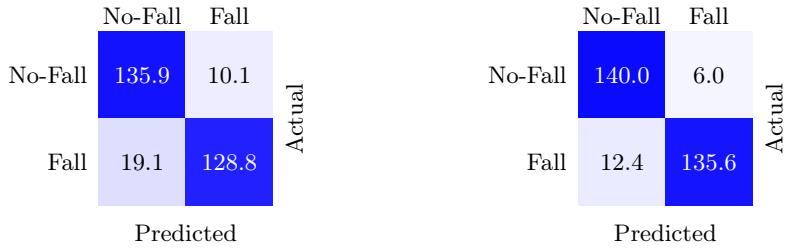
Table 4. Per-fold LOSO fall F_1 for all classifiers (mean ± std over 20 seeds). Each column holds out one subject. The macro average is the mean over the three folds.

Model	P1	P2	P3	Macro
Height threshold	$.790 \pm .028$	$1.000 \pm .000$	$.899 \pm .005$	$.896 \pm .009$
Shallow, LogReg	$.927 \pm .012$	$.868 \pm .020$	$.951 \pm .014$	$.915 \pm .010$
Shallow, Random Forest	$.919 \pm .008$	$1.000 \pm .000$	$.936 \pm .004$	$.952 \pm .003$
GCN, Two-Stream (rel.)	$.816 \pm .078$	$.881 \pm .059$	$.796 \pm .112$	$.831 \pm .043$
GCN, Joint-only (rel.)	$.745 \pm .046$	$.886 \pm .078$	$.782 \pm .092$	$.804 \pm .039$
GCN, Two-Stream (abs.)	$.874 \pm .031$	$.831 \pm .093$	$.922 \pm .044$	$.876 \pm .032$
GCN, Joint-only (abs.)	$.873 \pm .038$	$.826 \pm .128$	$.949 \pm .026$	$.882 \pm .044$

Table 3 macro-averages the three held-out folds with equal weight while the confusion matrix pools all held-out clips across folds and seeds, the fall recall implied by the matrix ($135.6/148.0 \approx 0.916$) differs from the macro-averaged recall of 0.936, the difference reflecting the equal weight the macro-average gives to the small, perfectly separated fold P2.

Table 4 reports the per-fold fall F_1 of all classifiers. The ranking is not uniform across folds: logistic regression exceeds the random forest on P1 and P3, and the random forest's macro advantage stems largely from the perfectly separated P2 fold, while the height threshold, competitive on macro average, is the weakest model on P1 (.790, below every graph-convolutional variant). The random forest is nonetheless the most consistent classifier, never falling below .919, whereas the other models vary more strongly between folds. With only three folds these macro averages are sensitive to the small P2 fold, which reinforces the limitation noted below; the robust observation across folds is that no graph-convolutional configuration is the single best on any fold, so the simple models are at least competitive with the graph-convolutional network throughout.

Table 5 reports model complexity and single-clip inference cost. The two-stream model contains 207,680 parameters and occupies 1.07 MB on disk, while the joint-only model contains 103,726 parameters and occupies 0.54 MB; inference latency was measured on a CPU only (Apple Silicon M2, eight cores, TensorFlow 2.21) with batch size one, averaged over 300 runs after warm-up, and the two-stream model requires 0.76 ± 0.07 ms per clip while the joint-only



(a) GCN, Two-Stream (abs.) (b) Random forest (abs.)

Fig. 5. Aggregated LOSO confusion matrices, averaged over twenty seeds and pooled across all three held-out subjects; numbers give the mean clip count per run and cell shading the row-normalized rate. The recommended random forest (b) lowers both false positives and false negatives relative to the strongest graph-convolutional configuration (a).

Table 5. Model complexity and single-clip inference cost. Latency measured on CPU only (Apple Silicon M2, 8 cores), batch size 1, mean ± std over 300 runs after warm-up. The Two-Stream row covers both the relative and absolute variants (identical architecture); the random forest and logistic regression operate on the flattened absolute coordinates. The random-forest latency is for the 300-tree scikit-learn implementation and is dominated by per-call overhead at batch size one. A height threshold on the head coordinate classifies a clip in about 1.6 µs, negligible relative to the entries above, and is therefore omitted from the table.

Model	Params	Size (MB)	MFLOPs	Latency (ms)	Clips/s
GCN, Two-Stream	207 680	1.07	13.2	0.76 ± 0.07	1316
GCN, Joint-only	103 726	0.54	6.6	0.45 ± 0.03	2237
Random forest (abs.)	–	0.57	–	12.69 ± 2.21	79
Shallow, LogReg (abs.)	460	0.004	≈ 0.001	0.09 ± 0.02	11480

model requires 0.45 ± 0.03 ms. At the nominal frame rate of 10 Hz the input length of $T = 17$ frames corresponds to an observation window of approximately 1.7 s, whereas the compared models classify a clip in between about 0.1 and 13 ms on a CPU (Table 5), so that even the most expensive model leaves the classification stage far from the limiting factor for real-time operation and the end-to-end latency is governed by the observation window rather than by the model. Both baselines avoid gradient-based training and run on a CPU without an accelerator, but their per-clip inference costs differ markedly. The linear model reduces classification to a single dot product and is the lightest of all compared models, whereas the random-forest ensemble is the most expensive per clip, though still far below the real-time budget set by the observation window; all compared models therefore sustain real-time operation.

5 Discussion

The principal empirical finding of this evaluation is a reversal between the two protocols. Under the clip-level protocol the learned models cluster near a fall F_1 of 0.98, differing by no more than seed-level variability, so this protocol alone would suggest that the choice of classifier is immaterial. Under the subject-independent protocol the ordering changes: the random forest on absolute coordinates attains the highest fall F_1 (0.952 ± 0.003) and exceeds the strongest graph-convolutional configuration (0.876 ± 0.032) by 0.076; logistic regression likewise exceeds it, while a single height threshold is on par with it (0.896 vs. 0.876). These gaps are larger than the run-to-run standard deviations and hence stable under reseeding, but with only three held-out subjects and the large between-fold spread reported in Section 4 we read the ordering as an observation on this sample rather than as a cross-subject significance result. The additional capacity of the graph-convolutional model therefore does not translate into improved cross-subject performance on this dataset, and reporting only clip-level accuracy would have concealed this entirely. This is the central role of the dual-protocol evaluation: it does not merely lower the reported numbers but changes which model appears best.

A plausible explanation is that the discriminative signal is dominated by a single subject-invariant cue, the absolute vertical coordinate of the keypoints, which encodes height above the ground directly. Two observations support this. Within the graph-convolutional family, absolute coordinates exceed torso-relative ones by more than run-to-run noise for the joint-only model (0.078 vs. $\sigma_{\text{comb}} = 0.059$) and favorably but within it for the two-stream model, because subtracting the torso removes exactly this height information; and a height threshold on the head coordinate alone already matches the graph-convolutional network. The cost of the bare threshold is a high false-positive rate of 0.247: it flags nearly every fall (pooled recall 0.951, equivalently the macro-averaged 0.961 of Table 2 up to the same fold-weighting effect noted above for the random forest) but also fires on low-posture non-fall movements such as sitting down, bending, or tying shoes. The random forest, operating on the full flattened feature vector, suppresses these false alarms to a rate of 0.041 while retaining high recall (0.916), which is why it leads. The graph-convolutional model lies between the two, with a false-positive rate of 0.069 but a lower recall of 0.871; its global average pooling over time and its graph structure, designed for distributed patterns across many joints, do not preferentially capture the height extremum that dominates this task, and with only two training subjects per fold its capacity generalizes less well, as reflected in its clip-level-to-LOSO drop of 0.104 against the random forest's 0.025.

The within-family ablation is consistent with this account. Under the subject-independent protocol the addition of the motion stream changes fall F_1 by less than σ_{comb} under both coordinate frames, so any motion-stream effect is below run-to-run noise on this sample, and the advantage of absolute over relative coordinates is the largest single effect within the graph-convolutional family. On this basis we recommend a random forest on absolute keypoint coordinates as

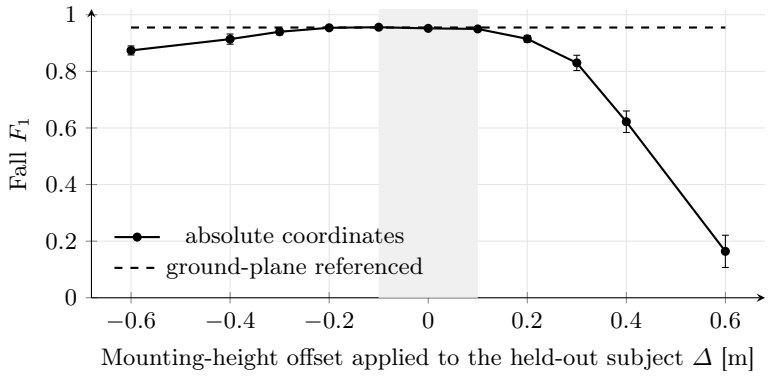


Fig. 6. Robustness of the recommended random-forest detector to a change in sensor mounting height, modelled as a vertical offset Δ applied only to the held-out (deployment) subject under the leave-one-subject-out protocol (mean ± standard deviation over 20 seeds). On absolute coordinates the fall F_1 is stable only within roughly ±0.1 m (shaded) and degrades sharply for larger upward offsets, where fallen keypoints rise into the height range associated with standing and falls are missed. Referencing the vertical coordinate to a floor estimate recovered from the data removes the dependence entirely, matching the unshifted score across the full range.

a simple and robust detector for this representation; it requires no accelerator and trains in seconds, and although its per-clip inference is the most expensive of the compared models (Table 5), it remains well within the real-time budget set by the observation window. Where inference cost is critical, the linear model retains most of the cross-subject robustness at a small fraction of that cost, so best accuracy and lowest cost point to different baselines. These comparisons are drawn on three subjects with large between-fold variability, most pronounced for the smallest held-out subject, which bounds the strength of any conclusion; whether the additional capacity of the graph-convolutional model would pay off on a larger and more diverse dataset remains open.

To test how strongly the recommended detector depends on the sensor mounting height without recording new data at different heights, we re-evaluate it on the existing clips under a controlled, synthetic change of that height. Because a uniform vertical offset applied to all recordings is absorbed by the per-feature standardisation and leaves every model unchanged, we instead model a sensor remounted at a different height for the deployment subject as a vertical translation applied only to the held-out subject, added to the three position coordinates while the offset-invariant velocities and accelerations are left unchanged. Under this shift the random forest on absolute coordinates tolerates only about ±0.1 m before its fall F_1 drops beyond seed-level variability, and it degrades sharply for larger upward shifts (Figure 6; from 0.952 at no shift to 0.830 at +0.3 m and 0.164 at +0.6 m); the degradation is asymmetric and recall-driven, as raising the mounting lifts the fallen keypoints into the height range the detector asso-

ciates with standing so that falls are missed (recall $0.935 \rightarrow 0.472$ at $+0.4$ m at near-perfect precision), whereas lowering it instead produces false alarms at preserved recall. Expressing the vertical coordinate relative to a floor estimate recovered from the data (a robust low percentile of the feet coordinate, a proxy for the ground plane that the pipeline estimates and then discards) removes this dependence entirely: the same detector is invariant across the full ± 0.6 m range at 0.955 ± 0.003, matching its unshifted score, and the same shift degrades the logistic-regression and height-threshold baselines and is likewise removed by floor referencing. This confirms that the discriminative cue is the absolute height above the ground and shows that the recommended detector requires either refitting or, more simply, ground-plane-referenced coordinates when the mounting height changes. Because it is applied as an offset to the existing coordinates rather than recorded anew, this experiment isolates the effect of the mounting height on the learned decision boundary but does not reproduce the other physical consequences of a real remounting, such as field-of-view truncation, changes in point density, or keypoint-extraction accuracy; it models a level remount as a pure vertical translation and so does not capture sensor tilt, and the floor proxy is estimated per subject rather than per recording. A full treatment on data recorded at genuinely different mounting heights remains future work.

There are several limitations that bound the scope of these results. First, the dataset comprises three young, healthy adults (aged 25–29) performing instructed falls onto mats, whereas the target population is older adults, whose fall kinematics are known to differ; the cross-subject metrics are moreover based on only three held-out subjects and exhibit large between-fold variability, so they should be read as indicative rather than conclusive. Second, although the system was deployed and operated online as described in Section 3, its online behavior is reported only qualitatively; a quantitative evaluation on uninterrupted streams, including detection latency, false alarms per unit time, and missed events under continuous operation, remains future work. Third, the limited vertical field of view of the sensor has two consequences. At short sensor-to-person distances, a falling person may partially leave the measurement volume, which affects detection independently of the classifier; and because full-body coverage forces a sensor-to-person distance of several metres at roughly half the body height (Section 3), the unit must stand freely within the living space, where it occupies floor area and can itself become an obstacle. Fourth, although some recordings contain more than one person, each clip is the sequence of a single tracked and labelled person, and the behavior of the tracking and classification stages under sustained multi-person occlusion is not separately quantified here. Fifth, two further practical constraints arise from the sensor rather than from the classifier: the rotating sensor emits audible noise during continuous operation, which may affect acceptance in living and sleeping areas, and the background-subtraction stage requires an initial recording of the empty scene, which is harder to maintain in a furnished home where the static layout changes over time. Finally, as quantified above, the recommended detector's reliance on absolute coordinates ties it to the sensor mounting height, a dependence that a ground-plane-referenced

coordinate frame removes; and the advantage of simple models over the graph-convolutional network is established here on three subjects, and it need not persist on larger and more diverse datasets, on which higher-capacity models may exploit structure that simple classifiers cannot.

6 Conclusion

We presented a device-free fall detection system that operates on LiDAR point clouds through a three-keypoint skeleton representation, retaining no raw point cloud or visual appearance for classification, and implemented as a continuous real-time pipeline. On this representation we compared classifiers of widely differing complexity, from a single-feature height threshold and a shallow classifier on the flattened coordinates to a compact two-stream adaptive graph convolutional network, under both a clip-level and a subject-independent protocol over 20 random seeds. While all learned models performed comparably at the clip level (fall $F_1 \approx 0.98$), under subject-independent evaluation a random forest on absolute keypoint coordinates was the most consistent across our three subjects (0.952 ± 0.003) and a single height threshold already matched the graph-convolutional network (0.876 ± 0.032), whose additional capacity did not improve cross-subject performance; we traced this to the absolute vertical coordinate, which encodes height above the ground directly. This cross-subject comparison rests on only three held-out subjects and a macro ranking dominated by one perfectly separated fold, so we read it as indicative rather than conclusive. We therefore recommend a shallow classifier such as a random forest on absolute coordinates as a simple and robust detector for this representation, and we note that a clip-level protocol alone would have concealed these differences. The pipeline runs online on the live sensor stream, but the results reported here are measured offline on pre-segmented single-person clips and should be read as proof-of-concept; stream-level behavior such as detection latency, false alarms per unit time, and track loss under continuous operation is not quantified here. All models run in real time on a CPU, with per-clip inference far below the observation window. Future work includes quantitative evaluation on continuous, uninterrupted streams, validation across a larger and more diverse set of subjects, on which the trade-off between simple and higher-capacity models may shift, and analysis of multi-person scenarios.

References

1. Bouazizi, M., Lorite Mora, A., Ohtsuki, T.: A 2d-lidar-equipped unmanned robot-based approach for indoor human activity detection. Sensors **23**(5) (2023). https://doi.org/10.3390/s23052534, https://www.mdpi.com/1424-8220/23/5/2534
2. Bouazizi, M., Mora, A.L., Feghoul, K., Ohtsuki, T.: Activity detection in indoor environments using multiple 2d lidars. Sensors **24**(2) (2024). https://doi.org/10.3390/s24020626, https://www.mdpi.com/1424-8220/24/2/626

3. Bouazizi, M., Ye, C., Ohtsuki, T.: Activity detection using 2d lidar for healthcare and monitoring. In: 2021 IEEE Global Communications Conference (GLOBECOM). pp. 01–06 (2021). https://doi.org/10.1109/GLOBECOM46510.2021.9685470
4. D'Avis, N., Hartmann, H., Grothe, L., Faquiri, S.: Densevoxelnet3d: A compact 3d cnn architecture for human activity recognition using lidar point clouds. In: Durmaz Incel, Ö., Qin, J., Bieber, G., Kuijper, A. (eds.) Sensor-Based Activity Recognition and Artificial Intelligence. pp. 116–134. Springer Nature Switzerland, Cham (2026)
5. Dhiman, C., Vishwakarma, D.K.: A review of state-of-the-art techniques for abnormal human activity recognition. Engineering Applications of Artificial Intelligence **77**, 21–45 (2019). https://doi.org/https://doi.org/10.1016/j.engappai.2018.08.014, https://www.sciencedirect.com/science/article/pii/S0952197618301775
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (Jun 1981). https://doi.org/10.1145/358669.358692, https://doi.org/10.1145/358669.358692
8. Franco, A., Magnani, A., Maio, D.: A multimodal approach for human activity recognition based on skeleton and rgb data. Pattern Recognition Letters **131**, 293–299 (2020). https://doi.org/https://doi.org/10.1016/j.patrec.2020.01.010, https://www.sciencedirect.com/science/article/pii/S0167865520300106
9. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: A survey (2020), https://arxiv.org/abs/1912.12033
10. Imbeault-Nepton, T., Maitre, J., Bouchard, K., Gaboury, S.: Fall detection from uwb radars: A comparative analysis of deep learning and classical machine learning techniques (09 2023). https://doi.org/10.1145/3582515.3609535
11. Karar, M.E., Shehata, H.I., Reyad, O.: A survey of iot-based fall detection for aiding elderly care: Sensors, methods, challenges and future trends. Applied Sciences **12**(7) (2022). https://doi.org/10.3390/app12073276, https://www.mdpi.com/2076-3417/12/7/3276
12. Kovács, L., Bódis, B.M., Benedek, C.: Lidpose: Real-time 3d human pose estimation in sparse lidar point clouds with non-repetitive circular scanning pattern. Sensors **24**(11) (2024). https://doi.org/10.3390/s24113427, https://www.mdpi.com/1424-8220/24/11/3427
13. Lin, Y., Wei, Z., Zhang, W., Lin, X., Dai, Y., Wen, C., Shen, S., Xu, L., Wang, C.: Hmpear: A dataset for human pose estimation and action recognition. In: Proceedings of the 32nd ACM International Conference on Multimedia. p. 2069–2078. MM '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3664647.3681055, https://doi.org/10.1145/3664647.3681055
14. Maitre, J., Bouchard, K., Gaboury, S.: Fall detection with uwb radars and cnn-lstm architecture. IEEE Journal of Biomedical and Health Informatics **25**(4), 1273–1283 (2021). https://doi.org/10.1109/JBHI.2020.3027967

15. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. Neurocomputing **100**, 144–152 (2013). https://doi.org/https://doi.org/10.1016/j.neucom.2011.09.037, https://www.sciencedirect.com/science/article/pii/S0925231212003153, special issue: Behaviours in video
16. Patil, A.K., Balasubramanyam, A., Ryu, J.Y., Chakravarthi, B., Chai, Y.H.: An open-source platform for human pose estimation and tracking using a heterogeneous multi-sensor system. Sensors **21**(7) (2021). https://doi.org/10.3390/s21072340, https://www.mdpi.com/1424-8220/21/7/2340
17. Piñeiro, M., Araya, D., Ruete, D., Taramasco, C.: Low-cost lidar-based monitoring system for fall detection. IEEE Access **12**, 72051–72061 (2024). https://doi.org/10.1109/ACCESS.2024.3401651
18. Rinchi, O., Nisbett, N., Alsharoa, A.: Patients arms segmentation and gesture identification using standalone 3-d lidar sensors. IEEE Sensors Letters **7**(9), 1–4 (2023). https://doi.org/10.1109/LSENS.2023.3303081
19. Roche, J., De-Silva, V., Hook, J., Moencks, M., Kondoz, A.: A multimodal data processing system for lidar-based human activity recognition. IEEE Transactions on Cybernetics **52**(10), 10027–10040 (2022). https://doi.org/10.1109/TCYB.2021.3085489
20. Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., Sattarvand, J.: A comprehensive overview of deep learning techniques for 3d point cloud classification and semantic segmentation. Machine Vision and Applications **35**(4) (May 2024). https://doi.org/10.1007/s00138-024-01543-1, http://dx.doi.org/10.1007/s00138-024-01543-1
21. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition (2019), https://arxiv.org/abs/1805.07694
22. Statistisches Bundesamt (Destatis): Körpermaße der bevölkerung nach altersgruppen und geschlecht (mikrozensus, erstergebnisse 2025). https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Gesundheit/Gesundheitszustand-Relevantes-Verhalten/Tabellen/liste-koerpermasse.html (2026), accessed: 2026-06-23
23. Ullmann, I., Guendel, R.G., Kruse, N.C., Fioranelli, F., Yarovoy, A.: A survey on radar-based continuous human activity recognition. IEEE Journal of Microwaves **3**(3), 938–950 (2023). https://doi.org/10.1109/JMW.2023.3264494
24. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition (2018), https://arxiv.org/abs/1801.07455
25. Zhang, J., Wang, D., An, X., Lv, M., Chen, D., Sun, A.: A voxel-based 3d reconstruction and action recognition method for construction workers. Advanced Engineering Informatics **65**, 103203 (2025). https://doi.org/https://doi.org/10.1016/j.aei.2025.103203, https://www.sciencedirect.com/science/article/pii/S1474034625000965
26. Zhao, Z., Zhuang, C., Li, J., Sun, H.: Lidar-based human pose estimation with motionbert. In: 2024 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 1849–1854 (2024). https://doi.org/10.1109/ICMA61710.2024.10632929