

CTX-FALLNET: A HYBRID CNN TRANSFORMER–XGBOOST FRAMEWORK FOR ELDERLY DAILY ACTIVITY CLASSIFICATION AND FALL DETECTION

Rohit Parmar¹, Dr. Samir Trapasiya^{2*}, Rakesh Soni³

¹Research Scholar, Gujarat Technological University (GTU), Chandkheda, Ahmedabad, India, rohitkumar.parmar@cvmu.edu.in

^{1,2}Assistant Professor, Electronics & Communication Department, G.H. Patel College of Engineering and Technology, The Charutar Vidya Mandal (CVM) University, Vallabh-Vidyanagar, Gujarat, 388120, India, samirkumar.trapasiya@cvmu.edu.in

³Assistant Professor, Government Engineering College, Godhra, Gujarat, India, soni_rakesh@gtu.edu.in

*Corresponding author: Dr. Samir Trapasiya, samirkumar.trapasiya@cvmu.edu.in

ABSTRACT

Wearable inertial sensors enable continuous monitoring of daily activities and falls, but separate deep models for human activity recognition (HAR) and fall detection duplicate feature learning and underutilize shared motion representations. This paper presents CTX-FallNet, a shared-representation framework coupling both tasks without end-to-end multi-task learning. A CNN–Transformer backbone is trained on activities-of-daily-living (ADL) windows and frozen to generate a 128-dimensional embedding for independently trained XGBoost classifiers performing multi-class HAR and binary fall detection. The contribution lies not in the individual components but in the shared-representation design and subject-independent evaluation. On SisFall and KFall, subject-independent StratifiedGroupKFold cross-validation prevents participant overlap between training and evaluation. Fall detection achieved accuracies of $99.63 \pm 0.21\%$ and $99.13 \pm 0.23\%$, ROC-AUC values of 0.998 ± 0.001 and 0.9995 ± 0.0003 , sensitivities of $95.41 \pm 1.75\%$ and $98.60 \pm 0.81\%$, and specificities of $99.62 \pm 0.23\%$ and $99.32 \pm 0.38\%$ on SisFall and KFall, respectively, with low Expected Calibration Error and Brier scores. HAR achieved accuracies of $84.92 \pm 0.63\%$ and $83.82 \pm 3.32\%$ and macro-F1 scores of $83.65 \pm 0.69\%$ and $83.61 \pm 2.90\%$ across 19 and 21 classes, respectively. Linear probing, PCA, and t-SNE support embedding discriminability. Cross-dataset, cross-device, and real-world generalization remain to be established.

KEYWORDS: Fall Detection; Human Activity Recognition; Wearable Sensors; CNN-Transformer; XGBoost; Elderly Monitoring.

1. INTRODUCTION

Fall related injury is a major public health concern, particularly among older adults. Worldwide, falls are the second most common cause of unintentional-injury death among adults aged 60 years and older [1]. Falls can also result in fractures, prolonged hospitalization, loss of independence, and reduced quality of life [2]. Delayed detection further increases the risk of complications, particularly when an individual remains on the ground for an extended period [3]. Therefore, reliable and unobtrusive systems for recognizing daily activities and detecting falls are important for continuous monitoring.

Among the sensing modalities examined for fall detection: including ambient/radio-frequency, vision-based, and wearable inertial sensing, wearable inertial sensing is particularly suitable for continuous monitoring because it is relatively low-cost, privacy-preserving, and applicable in indoor and outdoor settings. Publicly available benchmarks such as SisFall [6] and KFall [7] have further supported research in this area. Recent inertial-sensing methods have progressed from threshold-based and handcrafted approaches toward deep representation learning, with CNNs capturing local motion patterns and Transformer encoders modelling longer-range temporal dependencies; hybrid CNN–Transformer designs have consequently emerged as a prominent approach on inertial benchmarks [8], [9].

Despite this progress, several limitations remain. First, multi-class activity recognition and binary fall detection are commonly addressed using separate deep models, resulting in duplicated feature-learning components and limiting the reuse of a common motion representation [9]. Second, while CNN–XGBoost approaches have demonstrated the use of learned representations with boosted classifiers [10], the combination of a shared CNN–Transformer representation with task-specific XGBoost heads for both HAR and fall detection remains comparatively underexplored. Third, the class imbalance between fall and non-fall samples requires careful evaluation, as overall accuracy alone may not adequately reflect minority-class detection performance [11].

To address these limitations, we propose CTX-FallNet, a unified shared-representation framework that couples multi-class human activity recognition (HAR) and binary fall detection through a CNN–Transformer backbone and task-specific XGBoost classifiers. A CNN–Transformer backbone is trained on activities-of-daily-living (ADL) windows and subsequently frozen to generate a shared 128-dimensional embedding for ADL and fall windows. Two independently trained XGBoost classifiers then use this representation for HAR and fall detection. The framework is evaluated independently on SisFall [6] and KFall [7] using subject-independent StratifiedGroupKFold cross-validation, with all windows from each participant confined to a single fold. Accordingly, the evaluation assesses within-dataset subject-independent generalization rather than cross-dataset transfer.

The contribution of CTX-FallNet does not lie in the individual CNN, Transformer, or XGBoost components, which have been investigated previously [9], [10], but in their integration through a shared representation and the subject-independent evaluation protocol. Rather than using separate deep feature extractors for the two tasks, the proposed framework reuses one ADL-trained CNN–Transformer representation through two task-specific XGBoost classifiers. Subject-independent evaluation is maintained throughout training and testing. Class imbalance is treated according to task requirements: class-weighted backbone training is used for the multi-class ADL task, whereas the binary fall-detection head is trained on the naturally imbalanced fall/non-fall distribution without resampling or class reweighting.

The main contributions of this work are summarized as follows:

Shared CNN–Transformer backbone: Three one-dimensional convolutional blocks and three Transformer encoder layers produce a pooled 128-dimensional embedding from inertial sensor windows.

Shared representation across both tasks: The backbone is trained once on ADL windows and frozen, and its embedding is reused by two independently trained, task-specific XGBoost classifiers for multi-class HAR and binary fall detection, rather than an end-to-end multi-task network.

Subject-independent evaluation: Five-fold StratifiedGroupKFold cross-validation keeps all windows from each participant within a single fold, supporting evaluation on previously unseen participants while preserving class distributions across folds.

Comprehensive evaluation: The framework is assessed using complementary predictive, calibration, representation, and model-comparison analyses, including accuracy, F1-score, Cohen's κ , ROC-AUC, PR-AUC, sensitivity, specificity, ECE, Brier score, confusion matrices, reliability diagrams, linear-probe and silhouette analyses, PCA, t-SNE, and ablation experiments.

The remainder of this paper is organized as follows. Section II reviews related work in vision-based, ambient/RF, and wearable-sensor fall detection, together with relevant deep-learning and boosted-classifier approaches. Section III presents the proposed CTX-FallNet methodology, including datasets, subject-independent partitioning, preprocessing, the shared CNN–Transformer backbone, task-specific XGBoost classifiers, training configuration, and evaluation metrics. Section IV presents the results. Section V discusses comparative, limitations, implications, and future directions. Section VI concludes the paper.

2. Related Work

Fall detection systems have been developed using ambient/ radio frequency, vision-based, and wearable sensing modalities. Ambient and Wi-Fi-based approaches enable device-free monitoring but can depend on environmental conditions and dedicated infrastructure [5]. Vision-based methods can provide rich motion information but may be affected by privacy, occlusion, and illumination conditions [4]. Wearable inertial sensing provides a practical alternative for unobtrusive monitoring, and datasets such as SisFall [6], KFall [7], MobiFall [12], and UMAFall [13] have enabled systematic evaluation of learning-based methods.

Early wearable approaches relied on handcrafted signal features, threshold rules, and conventional machine learning classifiers. Subsequent methods adopted deep learning to learn motion representations directly from inertial signals. CNN-based models can capture local temporal patterns and have also been developed in lightweight forms for resource constrained deployment. Recurrent architectures were subsequently combined with sensor fusion and attention mechanisms to model temporal evolution. For example, multimodal Bi-LSTM approaches have combined inertial and radar information, while CNN–BiLSTM and attention-based models have been used to enhance temporal feature learning [14].

More recent studies have explored hybrid recurrent, attention, and optimization-based architectures. Ahirwar et al. combined BiLSTM, BiGRU, and feature selection for fall detection [15], while Yadav and Gupta incorporated multiscale CNN features with additive attention [16]. Other work has explored spatial–temporal attention with BiLSTM for sensor-based fall classification [17], and weighted feature deep learning for activity recognition [18]. These studies demonstrate the continued use of temporal modelling and attention mechanisms for extracting discriminative motion characteristics, although their architectures and evaluation settings differ.

Transformer-based models have subsequently been investigated for inertial sensing because self-attention can model dependencies across a complete motion window [8]. Hybrid CNN–Transformer architectures combine local feature extraction with Transformer-based temporal modelling and have reported strong performance on inertial benchmarks [9]. In parallel, CNN–XGBoost approaches have explored the use of learned deep representations with gradient-boosted classifiers for tasks such as direction and severity-aware fall detection and activity recognition [10]. These studies provide relevant foundations for the two principal components of CTX-FallNet, but they do not establish the same shared representation and dual-task configuration adopted in the proposed framework.

Overall, the literature indicates two relevant directions: CNN–Transformer architectures for learning local and temporal motion representations, and CNN–XGBoost approaches for classification using learned features. However, the combination of a shared CNN–Transformer representation with independently trained XGBoost heads for both multi-class HAR and binary fall detection is less represented in the reviewed studies. In addition, differences in datasets, activity definitions, window configurations, and subject-partitioning protocols make direct performance comparisons difficult. These observations motivate the proposed CTX-FallNet framework and the subject-independent evaluation conducted on both SisFall and KFall.

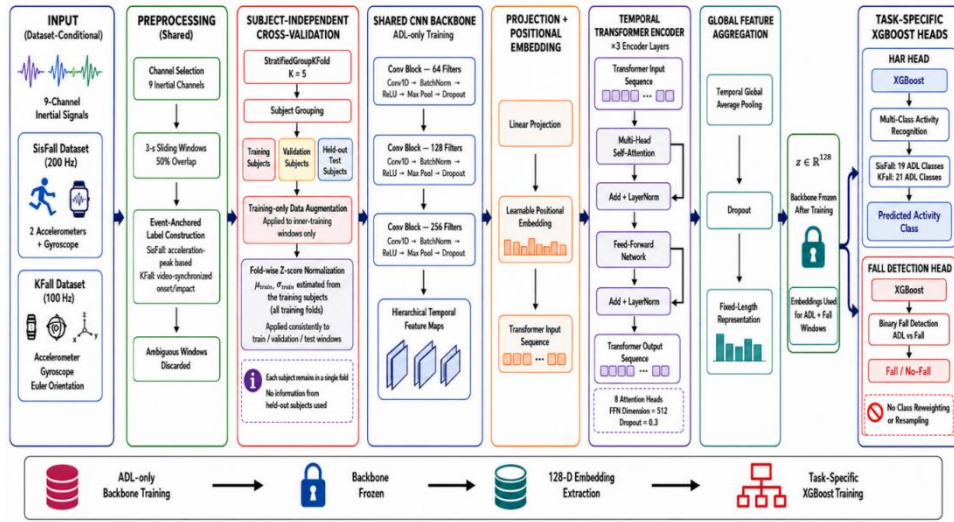


Figure 1. Detail Architectural diagram for the proposed CTX-FallNet framework

3. Proposed Work

Figure 1 illustrates the overall architecture of the proposed CTX-FallNet framework. The framework addresses two complementary recognition tasks from a single stream of wearable inertial sensor data: (i) multi-class Human Activity Recognition (HAR) and (ii) binary fall detection. As illustrated in Fig. 1, CTX-FallNet first trains a shared CNN–Transformer backbone using only ADL windows to learn a discriminative temporal representation. After training, the backbone is frozen and used as a fixed feature extractor to generate a 128-dimensional latent embedding for both ADL and fall windows. Two independently trained XGBoost classifiers subsequently operate on this shared embedding: one performs multi-class HAR, while the other performs binary fall detection. This shared feature-learning strategy enables both tasks to leverage a common representation while maintaining independent task-specific decision boundaries. The overall mapping of the proposed framework is expressed in Eq. (1) as:

$$(\hat{y}_{\text{HAR}}, \hat{y}_{\text{Fall}}) = (g_{\text{HAR}}(f_{\text{CTX}}(X)), g_{\text{Fall}}(f_{\text{CTX}}(X))) \quad (1)$$

where X denotes the input sensor window, $f_{\text{CTX}}(\cdot)$ represents the shared CNN–Transformer backbone, and $g_{\text{HAR}}(\cdot)$ and $g_{\text{Fall}}(\cdot)$ denote the task-specific XGBoost classifiers for multi-class HAR and binary fall detection, respectively.

The framework is evaluated independently on the SisFall and KFall datasets using the same architecture and training protocol. Since models are trained and evaluated separately on each dataset, the reported results reflect within-dataset subject-independent generalization rather than cross-dataset transfer. The complete subject-independent training and evaluation procedure of the proposed CTX-FallNet framework is summarized in Algorithm 1.

Algorithm 1: Subject-Independent Training and Evaluation of CTX-FallNet

Input:

- Raw sensor recordings R ; window length L ; stride S ;
- number of folds $K = 5$; random seed = 42;
- rotation range $\pm 15^\circ$; jitter standard deviation σ_j ;
- backbone configuration Θ_B ; xgboost configuration Θ_X

Output:

HAR and fall-detection metrics (mean $\pm$ SD over K folds)

- Select the nine inertial channels from R
- Segment each recording into windows $W_1 = X_{I_S}, X_{I_S+1}, \dots, X_{I_S+L-1}$
- Assign HAR and fall-detection labels to each W_i
- Discard windows with ambiguous fall-event labels
- Generate K subject-independent folds using stratifiedgroupfold
- For $k = 1$ to K do
- Assign four folds to training subjects and one fold to held-out subjects
- Split the training subjects into inner-training and validation subjects
- Apply training-only augmentation to the inner-training
- Windows using random three-axis rotation ($\pm 15^\circ$) and additive Gaussian jitter, according to the dataset-specific sensor configuration
- Compute M_{Train} And Σ_{Train} From the training subjects only
- Apply the M_{Train} And Σ_{Train} To training, validation, and held-out windows
- Compute class weights: $W_C = N/(C \cdot N_C)$
- Train F_{CTX} On ADL windows using weighted cross-entropy, with inner validation for early stopping
- And LR scheduling
- Freeze F_{CTX}

- Extract the 128-dimensional embedding: $z = F_{CTX}(X) \in \mathbb{R}^{128}$ For training, validation, and held-out test windows.
- Train G_{HAR} Using ADL embeddings from the training subjects
- Train G_{FALL} Using ADL and fall embeddings from the training subjects without class reweighting or Resampling
- Predict HAR and fall labels for the held-out subjects:
- $\hat{Y}_{HAR} = G_{HAR}(Z)$
- Predict fall-detection labels for the held-out test subjects:
- $\hat{Y}_{Fall} = G_{Fall}(Z)$
- Compute HAR and fall-detection performance metrics
- End for
- Report mean $\pm$ SD across the K held-out folds

3.1 Dataset Description

The proposed framework is evaluated independently on two publicly available wearable inertial datasets, sisfall and kfall, which differ in participant demographics, sensor configuration, sampling frequency, and activity taxonomy. Applying the same architecture to both datasets demonstrates the adaptability of CTX-fallnet across heterogeneous acquisition settings while maintaining a consistent learning framework.

a) **Sisfall:** The sisfall dataset [6] comprises 19 activities of daily living (adls) and 15 fall types collected from 38 participants (23 young adults and 15 older adults). The sensing unit, mounted at the waist, contains two tri-axial accelerometers (ADXL345 and MMA8451Q) and one tri-axial gyroscope (ITG3200), providing nine inertial channels sampled at 200 Hz. The dataset contains 2,707 ADL and 1,798 fall recordings.

b) **Kfall:** The kfall dataset [7] contains 21 adls and 15 engineered fall types acquired from 32 participants using a lower-back-mounted nine-axis inertial sensor that records tri-axial acceleration, angular velocity, and Euler-angle signals at 100 Hz. The dataset consists of 2,729 ADL and 2,346 fall recordings. Unlike sisfall, kfall provides video-synchronized fall onset and impact annotations, which are utilized for event-aware fall window labelling during preprocessing.

- **Class imbalance:** In both datasets, fall windows are substantially fewer than ADL windows, while the number of windows per activity also varies because of differences in trial duration. The multi-class HAR task addresses this imbalance through class-weighted training of the shared backbone, whereas the binary fall detection task is evaluated under the naturally imbalanced distribution. No windows are discarded or artificially resampled.

3.2 Data Collection and Pre-processing

Both datasets are processed using a unified preprocessing pipeline to generate fixed-length labeled windows for the proposed CTX-fallnet framework. As preprocessing comprises channel selection, sliding-window segmentation, two-task label construction, training-only data augmentation, and fold-wise normalization. Dataset-specific parameters are introduced only where acquisition characteristics differ.

1) **Channel selection:** To maintain a common network architecture across both datasets, nine inertial channels are used as model inputs. For sisfall, the signals from two tri-axial accelerometers and one tri-axial gyroscope are retained. For kfall, the tri-axial acceleration, angular velocity, and Euler-angle signals are retained, while timestamp and frame-counter fields are discarded. Consequently, both datasets provide a nine-channel input despite differences in sensor configuration.

2) **Sliding-window Segmentation:** Each recording is segmented into fixed-length windows of 3.0 s with 50% overlap, using the native sampling frequency of each dataset to avoid resampling artefacts. This corresponds to 600 samples (step = 300) for sisfall (200 Hz) and 300 samples (step = 150) for kfall (100 Hz). The i -th segmented window is defined as Eq. (2):

$$W_i = \{X_{Is}, X_{Is+1}, \dots, X_{Is+L-1}\} \quad (2)$$

Where L is the window length, S is the stride between successive windows, and W_i Denotes the i -th segmented multichannel sensor window. Windowing begins from the first sample of each recording, and only complete windows are retained; incomplete trailing segments are discarded to avoid introducing artificial padding. Each segmented window inherits the participant identifier of its source recording, enabling the subject-independent partitioning described in Section III-C.

3) **Two-task Label Construction:** Each segmented window is assigned two independent labels corresponding to the two recognition tasks.

For binary fall detection, labels are generated at the event level rather than the trial level. In kfall, the provided video-synchronized fall onset and impact annotations are used directly. A window is labelled Fall only if it contains the annotated impact sample, whereas windows occurring entirely before the onset are labelled No-Fall. Windows spanning the onset-to-impact transition without containing the impact, as well as post-impact windows, are discarded to avoid ambiguous supervision. Since sisfall does not provide temporal fall annotations, the impact instant is estimated as the maximum resultant acceleration magnitude computed from the raw acceleration signals, and the onset is defined as 1s before the estimated impact. The same event-aware labelling strategy is then applied.

For multi-class HAR, only ADL windows are retained. The activity labels comprise 19 ADL classes (D01–D19) for sisfall and 21 ADL classes (D01–D21) for kfall. Fall windows are excluded from this task because they do not represent distinct activity categories and are recognized exclusively by the binary fall detection branch. All labels are integer encoded for model training.

4) **Training-only Data Augmentation:** To improve robustness to sensor orientation and mounting variability, data augmentation is applied only to the training windows of each cross-validation fold. Random three-axis rotation ($\pm 15^\circ$) and additive Gaussian jitter are applied while preserving the original window labels. For sisfall, rotation is performed on all inertial triads (two accelerometers and one gyroscope). For kfall, rotation is applied only to the accelerometer and gyroscope triads, whereas the Euler-angle channels receive additive jitter only, since Euler-angle measurements do not transform linearly under rigid-body rotation. Validation and test windows are not augmented.

5) **Fold-wise Z-score Normalization:** Following augmentation, each sensor channel is standardized using z-score normalization, where the mean and standard deviation are estimated exclusively from the training subjects within the current cross-validation fold. The normalized sensor sample is computed as Eq. (3):

$$X' = \frac{X - M_{\text{Train}}}{\Sigma_{\text{Train}} + \epsilon} \quad (3)$$

Where M_{Train} And Σ_{Train} Denote the mean and standard deviation computed exclusively from the training subjects within the current cross-validation fold, and ϵ is a small constant added to prevent division by zero.

The same statistics are subsequently applied to the corresponding validation and test windows, preventing distributional information from the held-out subjects from influencing model training. A small numerical constant is added to the denominator to avoid division by zero for constant-valued channels.

3.3 Subject-independent Data partitioning

Model selection and evaluation used subject-independent stratified 5-fold group cross-validation (stratifiedgroupkfold). Each window retained the subject identifier of its source recording, and all windows from a participant across recordings and trials were assigned to the same fold, preventing participant overlap between training and test sets and avoiding leakage from overlapping windows. Stratification maintained class distributions across folds, particularly for the minority fall class, while a fixed seed (42) ensured reproducibility. In each iteration, four folds were used to train the shared CNN–Transformer backbone, and the resulting embeddings were used to train task-specific xgboost classifiers for multi-class HAR and binary fall detection. The remaining fold, containing unseen participants, was reserved exclusively for evaluation. Metrics were computed for each test fold and reported as mean $\pm$ standard deviation across five folds. The same protocol was applied to both tasks, ensuring subject-independent generalization.

3.4 Shared CNN Backbone

The shared CNN–Transformer backbone is identical for both datasets and serves as the common feature extractor for both recognition tasks. The backbone extracts temporal features from the nine-channel sensor windows and produces 128-dimensional latent embedding. It consists of the following components:

- **Three 1-D convolutional blocks:** Three 1-D convolutional blocks with 64, 128, and 256 filters, respectively. Each block comprises Conv1D, Batch Normalization, relu activation, Max Pooling, and Dropout, progressively learning hierarchical temporal representations from the multichannel sensor sequence.
- **Feature projection and positional embedding:** The pooled feature map is linearly projected into the transformer embedding dimension and combined with learnable positional embeddings to preserve temporal information.
- **Temporal Transformer encoding:** Three Transformer encoder layers model temporal dependencies using multi-head self-attention and position-wise feed-forward networks with residual connections and layer normalization.
- **Global feature aggregation:** Global Average Pooling is applied along the temporal dimension, followed by Dropout, to obtain a fixed-length 128-dimensional latent embedding, as defined in Eq. (4). This shared embedding is subsequently provided to separate xgboost classifiers for HAR and fall detection.

$$Z = \frac{1}{T} \sum_{T=1}^T Z_T = F_{\text{CTX}}(X) \in \mathbb{R}^{128} \quad (4)$$

The backbone is trained independently within each cross-validation training fold using only ADL windows and a class-weighted multi-class cross-entropy objective. Class weights are computed from the training-fold class frequencies. After training, the backbone is frozen and used to extract 128-dimensional embeddings for both ADL and fall windows. These embeddings are then classified using separate xgboost heads for HAR and fall detection:

3.5 Task-Specific xgboost Classification Heads

The shared 128-dimensional embedding from the frozen CNN–Transformer backbone feeds two independently trained xgboost classifiers, one for each task, as defined in Eqs. (5)–(6):

$$\hat{Y}_{\text{HAR}} = G_{\text{HAR}}(Z) \quad (5)$$

$$\hat{Y}_{\text{Fall}} = G_{\text{Fall}}(Z) \quad (6)$$

Where z denotes the latent embedding, extracted by the shared CNN–Transformer backbone, while $G_{\text{HAR}}(\cdot)$ And $G_{\text{Fall}}(\cdot)$ Represent the xgboost classifiers for multi-class human activity recognition and binary fall detection, respectively. The HAR head is trained only on ADL windows (19 classes for sisfall and 21 for kfall), whereas the fall-detection head uses the complete ADL and fall window set for binary classification. As described in Section III-D, the fall detector is trained on the naturally imbalanced distribution without reweighting or resampling; its impact is evaluated in Section IV using sensitivity, specificity, precision, recall, ROC-AUC, and calibration analyses.

Both heads use the same subject-independent CV folds described in Section III-C. For each fold, the backbone is trained on the training subjects, frozen, and used to extract embeddings for the corresponding training, validation, and test subjects, thereby preserving subject independence throughout training and evaluation.

3.6 Training Configuration

The proposed CTX-fallnet framework is trained using a two-stage procedure corresponding to the architecture. In the first stage, the shared CNN–Transformer backbone is trained using only the training subjects within each subject-independent cross-validation fold. An inner subject-grouped validation split is used for model selection, early stopping, and learning-rate scheduling. After convergence, the backbone parameters are frozen, and the trained network is used to extract a 128-dimensional latent embedding for each segmented window.

In the second stage, the extracted latent embeddings are used to train two independent xgboost classifiers for multi-class HAR and binary fall detection. A fixed empirical hyperparameter configuration is applied consistently across all cross-validation folds and both datasets, ensuring a uniform training protocol without per fold hyperparameter tuning. Throughout the training pipeline, the subject-independent partitioning is strictly preserved. Consequently, the CNN–Transformer backbone, feature extraction stage, and both xgboost classifiers are trained exclusively using the training subjects of each fold, while the held-out subjects remain completely unseen until the final evaluation, thereby preventing information leakage and ensuring a fair assessment of subject-independent model generalization.

3.7 Evaluation metrics

CTX-fallnet is evaluated separately for binary fall detection and multi-class HAR using the subject-independent 5-fold CV protocol (Section III-C), with results reported as mean $\pm$ SD across folds.

Consistent with recent studies on wearable-based fall detection and elderly activity recognition, [20]–[25], Fall detection is assessed using accuracy, precision, recall (sensitivity), specificity, F1-score, Cohen's κ , and ROC-AUC, computed from the confusion matrix as in Eqs. (7)-(11):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (10)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

Since missed falls carry greater clinical risk than false alarms, recall is treated as the primary indicator. Cohen's κ , measuring agreement beyond chance, is given by Eq. (12):

$$K = \frac{P_O - P_E}{1 - P_E} \quad (12)$$

The ROC-AUC measures ranking quality, i.e., the probability that a fall window receives a higher predicted score than a non-fall window.

Probability calibration is assessed using the Brier score (Eq. 13) and Expected Calibration Error (ECE) (Eq. 14):

$$BS = \frac{1}{N} \sum_{N=1}^N (P_N - Y_N)^2 \quad (13)$$

$$ECE = \sum_{M=1}^B \frac{|B_M|}{N} |\text{Acc}(B_M) - \text{conf}(B_M)| \quad (14)$$

With lower BS and ECE indicating better calibration. Reliability diagrams are additionally used to visualize agreement between predicted probabilities and observed event frequencies.

HAR performance is assessed using accuracy, macro- and weighted-F1 scores, Cohen's κ , and macro one-vs-rest ROC-AUC, with per-class F1 and AUC additionally reported for activity-wise analysis [19] [25].

4. RESULTS

4.1 Experimental Configuration

All experiments were conducted on a Linux workstation equipped with two NVIDIA Tesla T4 GPUs (16 GB GDDR6). The shared CNN-Transformer backbone was implemented in Python 3.11 using TensorFlow/Keras 2.19.0 and trained with mixed-precision arithmetic (mixed_float16) to improve computational efficiency. The two task-specific classification heads were implemented using XGBoost 3.2.0, while scikit-learn 1.6.1 was used for data preprocessing, model evaluation, and performance analysis. GPU acceleration was provided through CUDA 12.x and cuDNN 9.1.

Training was performed using the Adam optimizer with a batch size of 128. Early stopping with best-weight restoration was employed to prevent overfitting, causing training to terminate before the maximum number of training epochs for both datasets. After training, the shared CNN-Transformer backbone was frozen, and a 128-dimensional latent embedding was extracted for each input window. These embeddings were subsequently used to train two independently optimized XGBoost classifiers for multi-class HAR and binary fall detection. During inference, each input window is processed once by the shared backbone, and the resulting embedding is evaluated independently by the two task-specific classification heads.

Training was performed using the Adam optimizer with a batch size of 128. Early stopping with best-weight restoration was employed to prevent overfitting, causing training to terminate before the maximum number of training epochs for both datasets. After training, the shared CNN-Transformer backbone was frozen, and a 128-dimensional latent embedding was extracted for each input window. These embeddings were subsequently used to train two independently optimized XGBoost classifiers for multi-class HAR and binary fall detection. During inference, each input window is processed once by the shared backbone, and the resulting embedding is evaluated independently by the two task-specific XGBoost classifiers.

4.2 Multi-class ADL Recognition Performance

Table 1 reports the subject-independent five-fold performance of CTX-FallNet for multi-class ADL recognition. On SisFall, the model achieved $84.92 \pm 0.63\%$ accuracy, $83.65 \pm 0.69\%$ Macro-F1, 0.8386 ± 0.0042 Cohen's κ , and 0.9667 ± 0.0075 Macro-AUC, with a narrow accuracy range (84.11-86.03%) indicating stable performance across held-out subjects. KFall achieved $83.82 \pm 3.32\%$ accuracy, $83.61 \pm 2.90\%$ Macro-F1, 0.8289 ± 0.0352

Table 1. Subject-independent stratified 5-fold cross-validation performance for multi-class ADL recognition on the SisFall (19 ADL classes) and KFall (21 ADL classes) datasets

Dataset	Fold	Accuracy (%)	Macro-F1 (%)	Cohen's κ	Macro-AUC
SisFall	1	84.27	83.27	0.8363	0.9732
	2	85.35	84.74	0.8336	0.9587
	3	84.11	83.62	0.8374	0.9587
	4	84.84	86.03	0.8418	0.9697
	5	86.03	83.96	0.8438	0.9732
	Mean $\pm$ SD	84.92 ± 0.63	83.65 ± 0.69	0.8386 ± 0.0042	0.9667 ± 0.0075
KFall	1	80.31	81.29	0.7918	0.9610
	2	89.16	88.54	0.8854	0.9864
	3	84.34	83.50	0.8344	0.9722
	4	82.24	81.73	0.8122	0.9751
	5	83.03	83.00	0.8207	0.9767
	Mean $\pm$ SD	83.82 ± 3.32	83.61 ± 2.90	0.8289 ± 0.0352	0.9743 ± 0.0091

κ , and 0.9743 ± 0.0091 Macro-AUC; its wider range (80.31–89.16%) and larger deviations indicate greater sensitivity to fold composition, potentially reflecting dataset-specific movement and activity differences that were not experimentally isolated. Nevertheless, KFall Macro-AUC remained high across folds (0.9610-0.9864). Mean accuracy differed by 1.10 percentage points and Macro-F1 by only 0.04 points, indicating comparable class-wise recognition. KFall achieved higher Macro-AUC, while Cohen's κ values above 0.82 indicate substantial agreement. Small accuracy-Macro-F1 gaps suggest that performance was not strongly dominated by high-frequency classes.

Table 2. Subject-independent stratified 5-fold cross-validation performance for binary fall detection on the SisFall and KFall datasets.

Dataset	Fold	Accuracy (%)	Sensitivity (%)	Specificity (%)	ROC-AUC
SisFall	1	99.71	94.00	99.84	0.9992
	2	99.29	94.66	99.32	0.9977
	3	99.61	96.17	99.50	0.9988

	4	99.70	94.09	99.87	0.9973
	5	99.86	98.13	99.59	0.9989
	Mean $\pm$ SD	99.63 $\pm$ 0.21	95.41 $\pm$ 1.75	99.62 $\pm$ 0.23	0.998 $\pm$ 0.001
KFall	1	98.82	99.18	98.69	0.9993
	2	99.47	99.51	99.45	0.9998
	3	99.18	98.18	99.53	0.9996
	4	99.06	97.49	99.65	0.9992
	5	99.10	98.64	99.27	0.9996
	Mean $\pm$ SD	99.13 $\pm$ 0.23	98.60 $\pm$ 0.81	99.32 $\pm$ 0.38	0.9995 $\pm$ 0.0003

4.3 Binary Fall Detection Performance

Table 2 summarizes CTX-FallNet's subject-independent five-fold fall-detection performance. On SisFall, the model achieved $99.63 \pm 0.21\%$ accuracy, $95.41 \pm 1.75\%$ sensitivity, $99.62 \pm 0.23\%$ specificity, and 0.9980 ± 0.0010 ROC-AUC; KFall achieved $99.13 \pm 0.23\%$, $98.60 \pm 0.81\%$, $99.32 \pm 0.38\%$, and 0.9995 ± 0.0003 , respectively. KFall showed 3.19 pp higher sensitivity and 0.0015 higher ROC-AUC, with smaller deviations in these two measures, whereas SisFall showed 0.50 pp higher accuracy and 0.30 pp higher specificity. Comparable accuracy variability (0.21% vs. 0.23%) indicates similarly stable overall classification. These differences may reflect sensor, activity, and fall-event characteristics, although these factors were not experimentally isolated. Figs. 2–3 corroborate these results: confusion matrices show strong diagonal dominance, with a smaller false-negative proportion for KFall (1.4% vs. 4.6%), while ROC curves approach the upper-left corner. Table 3 shows well-calibrated estimates (ECE: 0.0017/0.0029; Brier: 0.0061/0.0070), indicating slightly better calibration for SisFall. Event-level sensitivity was high (98.56%/99.96%), with 1.08 vs. 0.03 missed falls per participant.

Probability Calibration and Operating-Point Analysis: At the default 0.50 threshold, CTX-FallNet produced 9.03 and 16.68 false-positive windows/h on SisFall and KFall, respectively. ROC-derived operating points (Table 3) at 90%, 95%, and 99% sensitivity demonstrate a controllable trade-off: increasing the target from 95% to 99% raised false positives from 8.42 to 53.63 windows/h for SisFall and 3.56 to 24.18 windows/h for KFall. Reliability diagrams (Fig. 4) show that predicted probabilities closely track observed frequencies, with slightly better calibration for SisFall, consistent with its lower ECE (0.0017

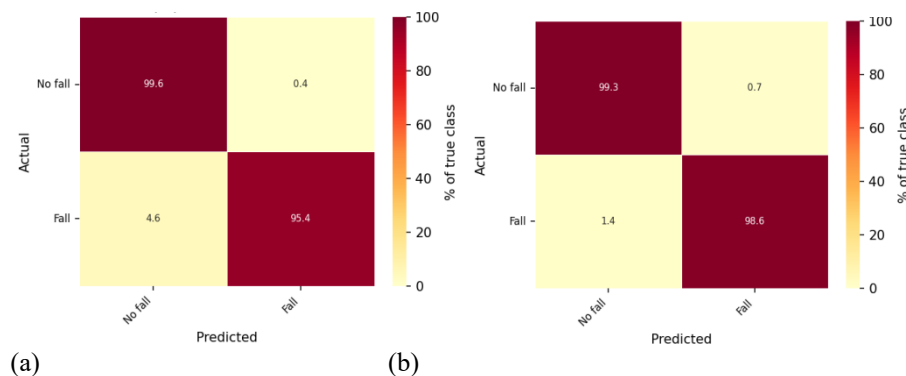


Figure 2. Normalized confusion matrices for binary fall detection on (a) SisFall and (b) KFall.

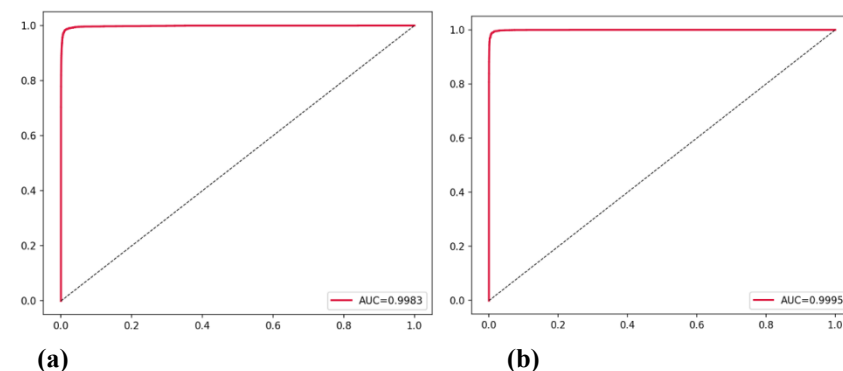
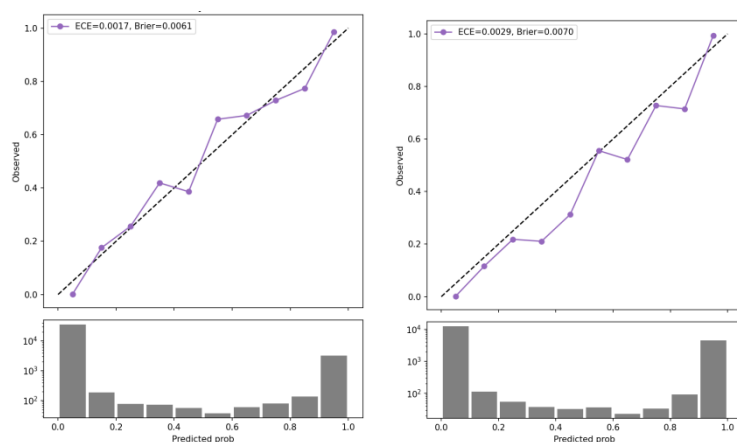


Figure 3. ROC curves for binary fall detection on (a) SisFall and (b) KFall.

vs. 0.0029) and Brier score (0.0061 vs. 0.0070). At the default threshold, event sensitivity reached 98.56% and 99.96%, respectively. These are threshold-dependent operating characteristics rather than absolute acceptability measures, demonstrating calibrated and

Table 3 Probability calibration and clinical operating characteristics of the proposed CTX-FallNet on the SisFall and KFall datasets.

Metric	SisFall	KFall
PR-AUC	0.990	0.999
ECE	0.0017	0.0029
Brier score	0.0061	0.0070
Event sensitivity (%)	98.56	99.96
False-positive windows/h (threshold = 0.50) (operating point)	9.03	16.68
Missed falls / participant	1.08	0.03
Specificity @ 90% window-level sensitivity (%) / FP windows/h	99.83 / 3.98	99.93 / 1.69
Specificity @ 95% window-level sensitivity (%) / FP windows/h	99.65 / 8.42	99.85 / 3.56
Specificity @ 99% window-level sensitivity (%) / FP windows/h	97.77 / 53.63	98.99 / 24.18



(a) (b)
Figure 4. Reliability diagrams for binary fall detection on (a) SisFall and (b) KFall, showing the relationship between predicted probabilities and observed event frequencies.

adjustable performance

4.4 Multi-class Activity Recognition Analysis

Table 4 Per-class F1-score and one-vs-rest AUC for the proposed model on SisFall (D01–D19) and KFall (D01–D21).

Activity	SisFall F1 (%)	SisFall AUC	KFall F1 (%)	KFall AUC
D01	95.63	0.9606	69.76	0.970
D02	96.61	0.9835	86.79	0.993
D03	95.57	0.9187	82.38	0.965
D04	96.68	0.9554	97.12	0.997
D05	93.47	0.9910	95.24	0.997
D06	92.70	0.9547	86.09	0.973
D07	68.31	0.9613	84.24	0.959
D08	77.13	0.9670	81.30	0.963
D09	64.64	0.9508	76.49	0.917
D10	75.66	0.9644	87.61	0.989
D11	86.13	0.9430	66.49	0.942
D12	82.93	0.9929	93.53	0.988
D13	77.01	0.9844	81.13	0.985
D14	85.47	0.9983	77.09	0.945
D15	70.16	0.9576	86.97	0.991
D16	81.03	0.9708	91.50	0.988
D17	88.63	0.9591	97.92	0.999
D18	91.54	0.9502	78.45	0.978
D19	87.98	0.9905	70.32	0.966
D20	—	—	86.93	0.983
D21	—	—	85.00	0.967

Macro avg.	84.59	0.9660	83.92	0.9742
Weighted avg. / Micro-AUC	84.28	0.9668	83.82	0.9733

Table 4 reports the per-class F1-scores and one-vs-rest (OvR) ROC-AUC for both datasets, with normalized confusion matrices and OvR ROC curves shown in Figs. 5 and 6, respectively. CTX-FallNet achieved comparable macro-level performance on SisFall (Macro-F1 = 84.59%, mean Macro-AUC = 0.9660) and KFall (Macro-F1 = 83.92%, mean Macro-AUC = 0.9742), with Macro-AUC computed as the mean across the five subject-independent folds. However, class-level performance varied substantially, with F1-scores ranging from 64.64–96.68% on SisFall and 66.49–97.92% on KFall. Fig. 5 shows strong diagonal dominance, with errors concentrated mainly among activities having similar motion characteristics, postures, or transitional patterns. The lowest-performing classes were D07, D09, and D15 for SisFall and D01, D11, and D19 for KFall. Their greater off-diagonal contributions indicate that these challenging classes disproportionately reduce Macro-F1, while most other activities remain consistently recognized. Differences in confusion patterns also suggest that confusion patterns vary with

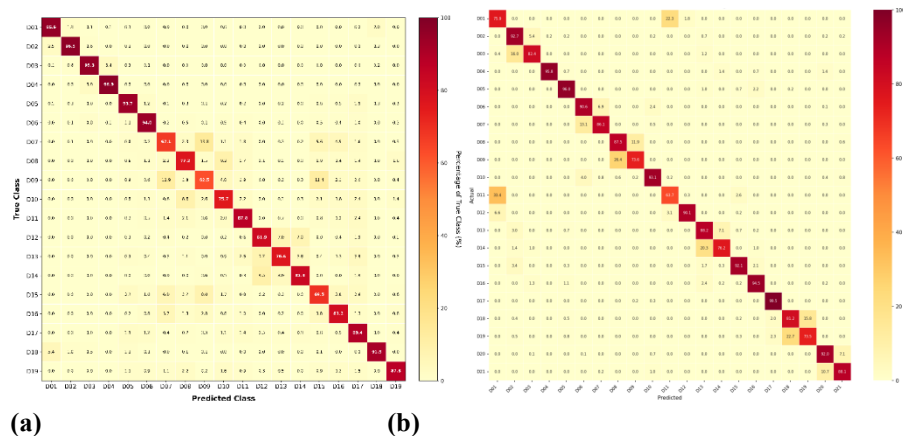


Figure 5. Normalized confusion matrices for multi-class human activity recognition (HAR) on the (a) SisFall (D01–D19) and (b) KFall (D01–D21) datasets under subject-independent five-fold cross-validation.

dataset-specific activity composition and sensor characteristics.

Fig. 6 shows strong OvR discrimination for most classes. Notably, some classes with lower F1 retained high AUC: SisFall D09 achieved F1 = 64.64% and AUC = 0.9508, while KFall D11 achieved F1 = 66.49% and AUC = 0.917. This divergence indicates that the model retains useful class-ranking capability despite threshold-dependent classification errors,

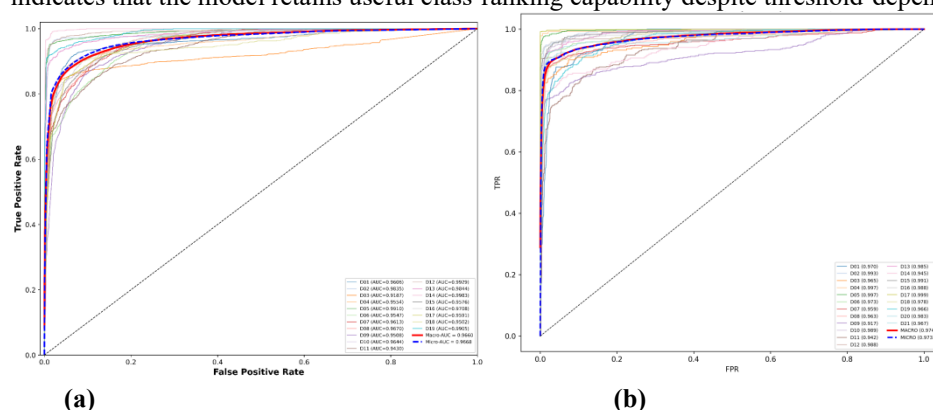


Figure 6. One-vs-rest (OvR) ROC curves for all activity classes obtained by the proposed CTX-FallNet on (a) SisFall (D01–D19) and (b) KFall (D01–D21).

suggesting that lower F1 primarily reflects class overlap and decision-boundary difficulty rather than a lack of discriminative information.

4.5 Ablation and Embedding Separability Analysis

Tables 5 and 6 present the ablation results for multi-class ADL recognition and binary fall detection, respectively, under the subject-independent stratified five-fold cross-validation protocol. Results are reported as mean $\pm$ standard deviation across the five folds and provide a descriptive comparison of predictive performance, model size, and inference latency across the evaluated configurations.

For multi-class ADL recognition (Table 5), the proposed CNN + Transformer + XGBoost configuration achieved the highest accuracy and Macro-F1 on both datasets. On SisFall, it reached $84.92 \pm 0.63\%$ accuracy and $83.65 \pm 0.69\%$ Macro-F1, versus $78.08 \pm 2.18\%$ and $77.74 \pm 2.03\%$ for CNN + Transformer + Random Forest — gains of 6.84 and 5.91 percentage points, respectively. On KFall, the corresponding improvements over Random Forest were smaller but consistent, at 2.57 points in accuracy and 3.39 points in Macro-F1. Notably, the accuracy gain exceeded the Macro-F1 gain on SisFall, while the reverse held on KFall, suggesting that XGBoost's contribution as a classifier head is dataset-dependent, though consistently beneficial. Critically, these gains come at negligible computational cost: the proposed model matches the CNN + Transformer + Random Forest variant in parameter count (0.73 M) and adds only 0.2 ms latency on both SisFall (87.1 vs. 87.3 ms) and KFall (103.2 vs. 103.4 ms). Relative to the Softmax baseline, XGBoost incurs just 0.3 ms overhead on both datasets, a negligible cost for the observed performance improvement.

For binary fall detection (Table 6), the proposed model also achieved the highest accuracy, sensitivity, and specificity on both datasets. On SisFall, it attained $99.63 \pm 0.21\%$ accuracy, $95.41 \pm 1.75\%$ sensitivity, and $99.62 \pm 0.23\%$ specificity, while on KFall it achieved $99.13 \pm 0.23\%$, $98.60 \pm 0.81\%$, and $99.32 \pm 0.38\%$, respectively. Relative to CNN + Transformer with Softmax, the proposed configuration improved accuracy by 5.61 percentage points on SisFall and 0.78 percentage points on KFall. Relative to CNN + Transformer + Random Forest, the corresponding gains were 2.92 and 0.14 percentage points. The larger improvements on SisFall indicate that the XGBoost classifier contributes more substantially to binary fall discrimination on that dataset, whereas the CNN–Transformer representation already provides stronger binary separation on KFall.

4.6 Embedding separability (linear probe)

To assess backbone representation quality independently of XGBoost, a linear probe was trained on frozen embeddings from each subject-independent fold. On SisFall, the probe achieved 0.9963 ± 0.0018 ROC-AUC and 0.9755 ± 0.0169 PR-AUC. On KFall, the corresponding values were 0.9996 ± 0.0003 and 0.9991 ± 0.0005 , respectively, as reported in Table 7. Although Silhouette values were modest, this unsupervised measure does not directly reflect supervised class separability. The t-SNE projections in Fig. 7 show recurring fall–ADL structures across folds, qualitatively supporting stable, discriminative representations under subject-independent evaluation.

Table 5 Ablation study for multi-class ADL recognition under subject-independent stratified five-fold cross-validation on the SisFall and KFall datasets

Dataset	Variant	Shared Backbone	Class Weighting	Params (M)	Latency (ms)	Accuracy (%)	Macro-F1 (%)
SisFall	CNN-only (Softmax)	✗	✓	0.29	19.8	65.37 ± 2.68	61.45 ± 2.22
	Transformer-only (Softmax)	✗	✓	0.40	64.1	71.46 ± 2.22	69.09 ± 2.36
	CNN + XGBoost	✗	✓	0.29	21.5	77.09 ± 0.38	75.52 ± 1.23
	CNN + Transformer (Softmax)	✓	✓	0.73	86.8	76.21 ± 2.10	74.94 ± 2.08
	CNN + Transformer + Random Forest	✓	✓	0.73	87.3	78.08 ± 2.18	77.74 ± 2.03
	CNN + Transformer + XGBoost (Proposed)	✓	✓	0.73	87.1	84.92 ± 0.63	83.65 ± 0.69
KFall	CNN-only (Softmax)	✗	✓	0.29	25.2	67.48 ± 1.36	66.56 ± 1.80
	Transformer-only (Softmax)	✗	✓	0.40	79.3	78.86 ± 4.60	78.85 ± 4.27
	CNN + XGBoost	✗	✓	0.29	23.3	76.94 ± 6.23	77.34 ± 5.43
	CNN + Transformer (Softmax)	✓	✓	0.73	102.9	79.50 ± 3.60	78.49 ± 2.87
	CNN + Transformer + Random Forest	✓	✓	0.73	103.4	81.25 ± 4.40	80.22 ± 3.42
	CNN + Transformer + XGBoost (Proposed)	✓	✓	0.73	103.2	83.82 ± 3.32	83.61 ± 2.90

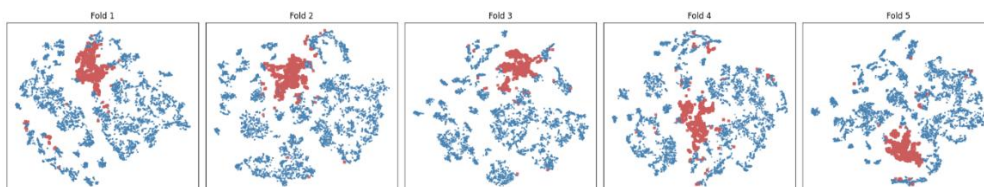
Table 6 Ablation study for binary fall detection under subject-independent stratified five-fold cross-validation on the SisFall and KFall

Dataset	Variant	Shared Backbone	Class Weighting	Params (M)	Latency (ms)	Accuracy (%)	Sensitivity (%)	Specificity (%)
SisFall	CNN-only (Softmax)	X	X	0.29	19.8	90.42 ± 1.12	86.15 ± 2.10	92.30 ± 1.05
	Transformer-only (Softmax)	X	X	0.40	64.1	91.68 ± 1.05	88.02 ± 1.98	93.45 ± 0.97
	CNN + XGBoost	X	✓	0.29	21.5	93.14 ± 0.89	90.27 ± 1.72	94.88 ± 0.81
	CNN + Transformer (Softmax)	✓	X	0.73	86.8	94.02 ± 0.76	91.55 ± 1.63	96.10 ± 0.68
	CNN + Transformer + Random Forest	✓	✓	0.73	87.3	96.71 ± 0.45	93.20 ± 1.88	98.14 ± 0.39
	CNN + Transformer + XGBoost (Proposed)	✓	✓	0.73	87.1	99.63 ± 0.21	95.41 ± 1.75	99.62 ± 0.23
KFall	CNN-only (Softmax)	X	X	0.29	25.1	90.20 ± 0.37	89.64 ± 0.54	90.40 ± 0.58
	Transformer-only (Softmax)	X	X	0.40	79.0	93.53 ± 0.48	93.61 ± 0.08	93.50 ± 0.64
	CNN + XGBoost	X	✓	0.29	25.6	95.73 ± 0.23	95.67 ± 0.31	95.74 ± 0.13
	CNN + Transformer (Softmax)	✓	X	0.73	102.9	98.35 ± 0.22	97.89 ± 0.67	98.51 ± 0.26
	CNN + Transformer + Random Forest	✓	✓	0.73	103.4	98.99 ± 0.19	98.15 ± 0.80	99.28 ± 0.28
	CNN + Transformer + XGBoost (Proposed)	✓	✓	0.73	103.2	99.13 ± 0.23	98.60 ± 0.81	99.32 ± 0.38

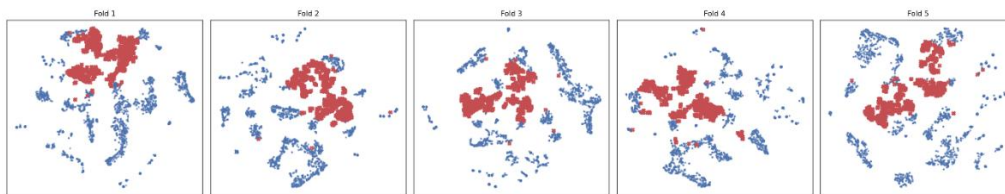
Note: Values are reported as mean ± standard deviation (SD) across the five subject-independent cross-validation folds.

Table 10. Linear probe performance and embedding separability metrics across the five subject-independent test folds on the SisFall and KFall datasets.

Fold	SisFall				KFall			
	Probe ROC-AUC	Probe PR-AUC	Silhouette	Centroid L2	Probe ROC-AUC	Probe PR-AUC	Silhouette	Centroid L2
1	0.9962	0.9773	0.0436	8.32	0.9997	0.9992	0.0505	5.03
2	0.9936	0.9472	0.0320	8.22	0.9999	0.9997	0.0436	4.76
3	0.9985	0.9871	0.0462	9.11	0.9997	0.9991	0.0282	4.35
4	0.9960	0.9764	0.0428	7.47	0.9991	0.9982	0.0602	5.37
5	0.9973	0.9897	0.0758	8.90	0.9997	0.9991	0.0283	4.21
Mean ± SD	0.9963 ± 0.0018	0.9755 ± 0.0169	0.0481 ± 0.0164	8.40 ± 0.65	0.9996 ± 0.0003	0.9991 ± 0.0005	0.0421 ± 0.0140	4.75 ± 0.48



(a)



(b)

Figure 7. t-SNE visualization of backbone embeddings for fall and ADL samples across the five subject-independent test folds of (a) the SisFall (b) Kfall dataset.

5 DISCUSSION

5.1 Deep Interpretation of Results

CTX-FallNet demonstrates strong and consistent performance under subject-independent five-fold cross-validation on both datasets. For multi-class ADL recognition, it achieves $84.92 \pm 0.63\%$ and $83.82 \pm 3.32\%$ accuracy, with Macro-F1 of $83.65 \pm 0.69\%$ and $83.61 \pm 2.90\%$ and Macro-AUC of 0.9667 ± 0.0075 and 0.9743 ± 0.0091 on SisFall and KFall, respectively. For binary fall detection, accuracy reaches $99.63 \pm 0.21\%$ and $99.13 \pm 0.23\%$, with ROC-AUC of 0.998 ± 0.001 and 0.9995 ± 0.0003 . Calibration is also reliable, with ECE of 0.0017 and 0.0029 and Brier scores of 0.0061 and 0.0070. These results demonstrate stable performance under subject-independent evaluation without implying cross-dataset generalization.

Despite high sensitivity, false-positive rates may create an alert burden in continuous monitoring because each misclassified window is counted independently without temporal aggregation. Published accelerometer-based detectors report 3–85 false alarms per day across thirteen algorithms [49], highlighting this challenge. Thus, the operating threshold should balance missed falls against unnecessary alerts according to the application.

The representation analysis shows that the learned embeddings contain strong discriminative information, as indicated by the high linear-probe AUC values. Modest Silhouette coefficients reflect different aspects of embedding geometry and do not directly contradict supervised separability. Although the backbone is trained only on ADL windows, its representations are also applied to fall windows, enabling the classifier to exploit shared motion characteristics without implying fall-specific feature learning. Fall windows are event-anchored using dataset-specific annotations, which provide consistent evaluation but may involve less temporal ambiguity than continuous real-world monitoring. Therefore, the binary results should be interpreted within this evaluation setting.

5.2 Comparison with Representative Published Studies

Direct comparison is limited by differences in task definitions, class granularity, and validation protocols. Studies reporting higher raw scores often use subject-mixed partitions, fewer classes, or lack explicit subject-level grouping, potentially yielding optimistic estimates. CTX-FallNet instead uses Stratified Group 5-fold CV with subjects confined to individual folds, providing a stringent unseen-subject evaluation. Among studies explicitly using subject-independent evaluation, CTX-FallNet outperforms the comparable CNN-CBAM-BiGRU baseline in accuracy, sensitivity, and F1-score. Comparisons with KFall studies are further limited because prior work primarily addresses binary or phase-specific detection rather than fine-grained ADL recognition. Thus, the results demonstrate competitive performance under a stringent subject-independent protocol rather than a definitive cross-study ranking.

5.3 Limitations and Future Work

Despite strong performance in binary fall detection and multi-class HAR, several limitations remain.

- Ecological validity. SisFall and KFall comprise scripted falls and ADLs collected under controlled conditions; therefore, results may not fully reflect free-living or clinical settings with greater motion variability and sensor noise.
- Generalization protocol. Evaluation used subject-independent StratifiedGroupKFold within each dataset, demonstrating unseen-subject generalization but not cross-dataset generalization, as neither cross-dataset nor LOSO evaluation was performed.
- Sensor configuration. Each dataset used a single IMU placement (waist for SisFall and lower back for KFall); robustness to alternative placements, devices, and multi-sensor fusion remains untested.
- Representation learning. The backbone is trained on ADL windows only, while XGBoost uses embeddings from both ADL and fall windows. Alternative strategies, such as joint multi-task or fall-aware training, warrant further investigation.

Residual confusion and deployment. Residual misclassification among kinematically similar ADLs suggests potential benefits from richer temporal representations. Clinical acceptability of the false-positive burden and real-time embedded performance, including latency, energy, and memory requirements, remain unvalidated.

Future Work: Future research will focus on validating CTX-FallNet using real-world and clinically supervised fall datasets, investigating cross-dataset and leave-one-subject-out evaluation protocols, and assessing robustness across diverse sensor placements and multi-sensor fusion scenarios. Additional work will explore alternative backbone supervision strategies, model compression and knowledge distillation for efficient on-device inference, and streaming, latency-aware implementations for continuous wearable fall monitoring.

5.4 Implications of the Study

The proposed CTX-FallNet framework has several practical implications for wearable human activity recognition and fall detection. First, the shared CNN–Transformer backbone enables both multi-class HAR and binary fall detection to be performed from a common latent representation, avoiding the need to train separate deep feature extractors for the two tasks. This design simplifies the overall learning pipeline while maintaining strong performance under subject-independent evaluation.

Second, the embedding analyses and linear probe experiments indicate that the learned latent representation preserves meaningful motion characteristics that support both activity recognition and fall discrimination. These findings suggest that a shared representation can effectively capture complementary information required by the two downstream classification tasks.

Third, the proposed framework provides both detailed activity context and binary fall alerts. In addition to identifying fall events, the activity recognition component can provide contextual information about the user's movement preceding or following a fall, which may assist subsequent clinical interpretation and long-term mobility assessment. Although further validation in real-world settings is required, the proposed framework demonstrates the potential of shared representation learning for wearable health-monitoring applications.

6 CONCLUSION

This work presented CTX-FallNet, a shared-representation framework that couples multi-class human activity recognition (HAR) and binary fall detection through a single CNN–Transformer backbone. Within each fold, the backbone is trained on activities-of-daily-living (ADL) windows and then frozen, and its 128-dimensional embedding is reused by two independently trained XGBoost classifiers, enabling both tasks to use a common latent representation without training a separate deep feature extractor for each.

Under subject-independent StratifiedGroupKFold cross-validation, CTX-FallNet achieved binary fall-detection accuracies of $99.63 \pm 0.21\%$ on SisFall and $99.13 \pm 0.23\%$ on KFall, with ROC-AUC values of 0.998 ± 0.001 and 0.9995 ± 0.0003 , together with low ECE and Brier scores. For multi-class HAR, accuracies of $84.92 \pm 0.63\%$ and $83.82 \pm 3.32\%$ were obtained across 19 and 21 activity classes, respectively, with macro-F1 scores of $83.65 \pm 0.69\%$ and $83.61 \pm 2.90\%$. Linear-probe evaluation, embedding visualizations, and the ablation study indicate that the frozen backbone provides discriminative features for both tasks, supporting the use of a shared latent representation.

These findings are based on within-dataset, subject-independent evaluation on the SisFall and KFall benchmarks and should not be interpreted as evidence of cross-dataset or real-world generalization. Future work will investigate real-world and clinically supervised fall datasets, leave-one-subject-out and cross-dataset evaluation, robustness to heterogeneous wearable sensors and multi-sensor fusion, and lightweight implementations for continuous real-time deployment.

REFERENCES

1. SaFalls. Available online: <http://www.who.int/en/news-room/fact-sheets/detail/falls> (accessed on 20 November 2024).
2. Parmar, R., Trapasiya, S. A Comprehensive Survey of Various Approaches on Human Fall Detection for Elderly People. *Wireless Pers Commun* 126, 1679–1703 (2022). doi:10.1007/s11277-022-09816-6
3. Newaz NT, Hanada E. The Methods of Fall Detection: A Literature Review. *Sensors (Basel)*. 2023 May 30;23(11):5212. doi:10.3390/s23115212.
4. T. Alanazi, M. Khaliluzzaman, and M. Alruwaili, "A Robust and Automated Vision-Based Human Fall Detection Using Deep Learning," *Applied Sciences*, vol. 13, no. 12, p. 6916, 2023. doi:10.3390/app13126916.
5. Tingwei Chen, Xiaoyang Li, Hang Li, Guangxu Zhu, Deep learning-based fall detection using commodity Wi-Fi, *Journal of Information and Intelligence*, Volume 2, Issue 4, 2024, Pages 355-364, doi:10.1016/j.jiixd.2024.04.001.
6. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. SisFall: A Fall and Movement Dataset. *Sensors* 2017, 17, 198. doi:10.3390/s17010198
7. Yu X, Jang J and Xiong S (2021) A Large-Scale Open Motion Dataset (KFall) and Benchmark Algorithms for Detecting Pre-impact Fall of the Elderly Using Wearable Inertial Sensors. *Front. Aging Neurosci.* 13:692865. doi: 10.3389/fnagi.2021.692865
8. Kibet, D.; So, M.S.; Kang, H.; Han, Y.; Shin, J.-H. Sudden Fall Detection of Human Body Using Transformer Model. *Sensors* 2024, 24, 8051. doi:10.3390/s24248051
9. Mohammed A.A. Al-qaness, Abdelghani Dahou, Mohamed Abd Elaziz, Ahmed M. Helmi, Human activity recognition and fall detection using convolutional neural network and transformer-based architecture, *Biomedical Signal Processing and Control*, Volume 95, Part B, 2024, 106412, ISSN 1746-8094, doi:10.1016/j.bspc.2024.106412.
10. Syed, A. S., Sierra-Sosa, D., Kumar, A., & Elmaghraby, A. (2022). A Deep Convolutional Neural Network-XGB for Direction and Severity Aware Fall Detection and Activity Recognition. *Sensors*, 22(7), 2547. doi:10.3390/s22072547
11. Mohammad, Z.; Anwary, A.R.; Mridha, M.F.; Shovon, M.S.H.; Vassallo, M. An Enhanced Ensemble Deep Neural Network Approach for Elderly Fall Detection System Based on Wearable Sensors. *Sensors* 2023, 23, 4774. doi:10.3390/s23104774
12. G. Vavoulas, M. Padiaditis, E. G. Spanakis and M. Tsiknakis, "The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones," 13th IEEE International Conference on BioInformatics and BioEngineering, Chania, Greece, 2013, pp. 1-4, doi:10.1109/BIBE.2013.6701629.
13. Eduardo Casilari, Jose A. Santoyo-Ramón, Jose M. Cano-García, UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection, *Procedia Computer Science*, Volume 110, 2017, Pages 32-39, doi:10.1016/j.procs.2017.06.110.
14. C. Li, Y. Wang, and Z. Liu, "Research on CNN-BiLSTM Fall Detection Algorithm Based on Improved Attention Mechanism," *Applied Sciences*, vol. 12, no. 19, p. 9671, 2022. doi:10.3390/app12199671.
15. M. S. Ahirwar, A. Kumar, and P. Singh, "Optimized Fall Detection Using Hybrid BiLSTM-BiGRU Additive Attention Model with BAOA Feature Selection," *Scientific Reports*, vol. 15, 2025. doi:10.1038/s41598-025-22909-z
16. A. Yadav and S. Gupta, "A Novel Additive Attention-Based MICNN-BiLSTM Model for Fall Detection Using Wearable Inertial Sensors," in *Proc. IEEE RAIT 2025*. DOI: doi:10.1109/RAIT65068.2025.11089122

17. Shin, J.; Miah, A.S.M.; Egawa, R.; Hassan, N.; Hirooka, K.; Tomioka, Y. Multimodal Fall Detection Using Spatial–Temporal Attention and Bi-LSTM-Based Feature Fusion. *Future Internet* 2025, 17, 173. doi:10.3390/fi17040173
18. Zhang J, Li Z, Liu Y, Li J, Qiu H, Li M, Hou G, Zhou Z, An Effective Deep Learning Framework for Fall Detection: Model Development and Study Design, *J Med Internet Res* 2024;26:e56750 doi:10.2196/56750
19. Mekruksavanich, S., & Jitpattanakul, A. (2026). Enhancing Wearable-Based Elderly Activity Recognition Through a Hybrid Deep Residual Network. *Machine Learning and Knowledge Extraction*, 8(4), 107. doi:10.3390/make8040107
20. Ahirwar, M.S., Soni, V. TCN-BiSRU-V2 fall detection model with performance evaluation and comparative analysis. *Discov Computing* 28, 186 (2025). doi:10.1007/s10791-025-09700-4
21. Chauhan and S. Vegad, “CVAE-ADS: A Deep Learning Framework for Traffic Accident Detection and Video Summarization ”, *j.electron.electromedical.eng.med.inform*, vol. 8, no. 1, pp. 185-205, Jan. 2026, DOI: 10.35882/jeeemi.v8i1.1139.
22. N. Gaud, M. Rathore, U. Suman and V. B. Semwal, "FIBiLS: Fall Detection of Healthy Elderly Using IMU Sensor and BiLSTM Model," in *IEEE Sensors Journal*, vol. 25, no. 19, pp. 37124-37131, 1 Oct.1, 2025, doi: 10.1109/JSEN.2025.3602945.
23. Jishnuraj K, Vergin Raja Sarobin M, Anbarasi J, Jasmine G and Panjanathan R (2026) Fall detection among elderly persons using FallCNN and transfer learning models. *Front. Artif. Intell.* 9:1734096. doi: 10.3389/frai.2026.1734096
24. Zhiyuan Jiang, Sike Ni, Mohammed A.A. Al-qaness, MDWD-KAN: Multilevel discrete wavelet decomposition with Kolmogorov–Arnold network for fall detection and activity recognition using wearable sensors, *Pervasive and Mobile Computing*, Volume 116, 2026, 102160, ISSN 1574-1192, <https://doi.org/10.1016/j.pmcj.2026.102160>.
25. Athar Noor Mohammad Rafee, Md Abu Obaida Zishan, Jannatun Noor, Decision tree-based real-time personalized Human Activity Recognition and fall detection utilizing cost effective highly resource-constrained microcontrollers, *Journal of Systems Architecture*, Volume 176, 2026, 103796, ISSN 1383-7621, <https://doi.org/10.1016/j.sysarc.2026.103796>.