Computer Modeling in
Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.066517

ARTICLE

Enhancing Fall Detection in Alzheimer's Patients Using Unsupervised Domain Adaptation

Nadhmi A. Gazem[1], **Sultan Noman Qasem**[2,3], **Umair Naeem**[4], **Shahid Latif**[5], **Ibtehal Nafea**[6], **Faisal Saeed**[7] **and Mujeeb Ur Rehman**[8,*]

[1]Department of Information Systems, College of Business Administration-Yanbu, Taibah University, Medina, 42353, Saudi Arabia
[2]Computer Science Department, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia
[3]King Salman Center for Disability Research, Riyadh, 11614, Saudi Arabia
[4]Department of Mechanical, Manufacturing and Mechatronics Engineering, RMIT University, Melbourne, VIC 3001, Australia
[5]School of Computing and Creative Technologies, University of the West of England, Bristol, BS16 1QY, UK
[6]College of Computer Science and Engineering, Taibah University, Medina, 41477, Saudi Arabia
[7]College of Computing, Birmingham City University, Birmingham, B4 7XG, UK
[8]School of Computer Science and Informatics, Institute of Artificial Intelligence, De Montfort University, Leicester, LE1 9BH, UK

*Corresponding Author: Mujeeb Ur Rehman. Email: mujeeb.rehman@dmu.ac.uk

Received: 10 April 2025; Accepted: 23 June 2025; Published: 31 July 2025

ABSTRACT: Falls are a leading cause of injury and morbidity among older adults, especially those with Alzheimer's disease (AD), who face increased risks due to cognitive decline, gait instability, and impaired spatial awareness. While wearable sensor-based fall detection systems offer promising solutions, their effectiveness is often hindered by domain shifts resulting from variations in sensor placement, sampling frequencies, and discrepancies in dataset distributions. To address these challenges, this paper proposes a novel unsupervised domain adaptation (UDA) framework specifically designed for cross-dataset fall detection in Alzheimer's disease (AD) patients, utilizing advanced transfer learning to enhance generalizability. The proposed method incorporates a ResNet-Transformer Network (ResT) as a feature extractor, along with a novel DualAlign Loss formulation that aims to align feature distributions while maintaining class separability. Experiments on the preprocessed KFall and SisFall datasets demonstrate significant improvements in F1-score and recall, crucial metrics for reliable fall detection, outperforming existing UDA methods, including a convolutional neural network (CNN), DeepCORAL, DANN, and CDAN. By addressing domain shifts, the proposed approach enhances the practical viability of fall detection systems for AD patients, providing a scalable solution to minimize injury risks and improve caregiving outcomes in real-world environments.

KEYWORDS: Fall detection; Alzheimer's Disease (AD); transfer learning; wearable sensors; KFall dataset

1 Introduction

Alzheimer's disease (AD) is a progressive, brain-destroying illness that leads to declining mental abilities, memory problems, and issues with movement control. The global number of people with Alzheimer's disease is projected to exceed 150 million by 2050, creating an unprecedented burden on healthcare systems [1]. The combination of cognitive symptoms in AD results in an increased risk of falls as patients experience unstable walking, spatial orientation problems, and medication-related side effects. Research indicates that individuals with AD experience between one and two falls per year [2], which is higher than the annual fall rate among healthy older adults. These falls can lead to fractures, traumatic brain injuries, and accelerated functional deterioration. The prevalence of falls among older adults with AD poses a significant public health issue, leading to fatal injuries, diminished quality of life, and elevated mortality rates. According to the World Health Organization (WHO), falls rank as the second leading cause of unintentional injury-related deaths globally, affecting between 28% and 35% of people aged 65 and older each year, with incidence increasing after age 70 [3,4]. The growing elderly population worldwide, projected to reach 2 billion by 2050, necessitates the immediate implementation of effective fall detection and prevention methods.

The implementation of Fall Detection Systems (FDS) reduces fall-related injuries through immediate medical response capabilities. The AD population requires prompt medical assistance because cognitive difficulties hinder their ability to seek help. Wearable sensors serve as the primary technology for traditional FDS, offering portability, cost-effectiveness, and privacy protection compared to vision-based systems [5,6]. Current solutions struggle with robustness when monitoring AD patients due to unpredictable fall patterns, and patients may sometimes neglect to wear devices or exhibit agitation toward sensor use [7,8]. Achieving real-world reliability necessitates addressing these limitations to ensure proper functionality.

Recent advancements in machine learning, particularly deep learning and domain adaptation techniques, have shown promising results in overcoming these challenges [9,10]. These methods have demonstrated the capability to learn robust and generalizable representations from sensor data, effectively bridging the gap between laboratory conditions and real-world environments [11]. Domain adaptation, specifically, has emerged as an effective method for leveraging labeled data from a source domain to enhance performance in an unlabeled or sparsely labeled target domain by addressing cross-domain distribution discrepancies [12]. Techniques such as adversarial learning [13] and contrastive learning [14] have been explored in the context of sensor-based human activity recognition (HAR), showing considerable improvement in cross-domain generalization.

This work addresses the existing limitations of current FDS methods by proposing an Unsupervised Domain Adaptation (UDA) framework based on transfer learning, explicitly designed to enhance generalization capabilities across datasets with varying sensor configurations and activity patterns. The main contributions of this work are:

- **Unified Cross-Dataset Preprocessing Framework:** A comprehensive preprocessing pipeline is proposed that standardizes dataset characteristics, including resampling frequency, windowing strategies, Z-score normalization, label unification, and stratified training and testing splits. This pipeline ensures valid cross-dataset generalization.
- **Novel ResNet Transformer Network (ResT):** A ResNet Transformer Network (ResT) is introduced, combining Residual Networks (ResNet) for stable deep feature learning with Transformers' robust global dependency modeling. This hybrid architecture effectively captures the complex temporal dependencies inherent in sensor-based fall detection tasks, thereby overcoming the limitations of traditional convolutional models.
- **DualAlign Loss for Robust Domain Adaptation:** A DualAlign Loss is developed, integrating Local Maximum Mean Discrepancy (LMMD) for class-wise domain alignment with Supervised Contrastive Loss (SupCon) for enhanced semantic compactness. This dual-loss strategy ensures discriminative, yet domain-invariant, feature representations across datasets.
- **Extensive Cross-Dataset Performance Evaluation:** A comprehensive experimental assessment was conducted using the KFall and SisFall datasets across various standards, revealing that the proposed scheme achieves substantial performance improvements compared to leading baselines in domain adaptation, as measured by important evaluation matrices, including F1-score and recall. These outcomes affirm the applicability and efficacy of the designed framework in real-world fall detection systems.

The remainder of the paper is organized as follows: Section 2 presents an overview of related studies. Section 3 provides a detailed description of the proposed methodology, including the ResT architecture and the DualAlign Loss. Section 4 describes the experimental setup, dataset preprocessing, and evaluation metrics. Results, comparative analyses, and ablation study are presented in Section 5. Finally, Section 6 concludes the paper and outlines future research directions.

2 Related Works

This section reviews recent studies on fall detection frameworks. Jain and Semwal [15] developed a pre-impact fall detection system (FDS) employing deep learning that can identify falls within 0.5 s of the onset of the fall phase. The authors created an automated feature extraction method to gather time-based characteristics from wearable sensor data during various types of falls. The core classifier utilized an ensemble of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. The proposed framework introduced both a transitional window concept and dataset fusion as strategies to enhance system generalizability. Yu et al. [16] developed TinyCNN, a lightweight convolutional neural network for wearable inertial-sensor-based fall detection that operates efficiently. The authors designed a two-stage feature extraction system, which improved computational efficiency before embedding the model onto an ultra-low-power microcontroller for real-time operation. The framework achieved better interpretability through the implementation of class activation mapping (CAM). A comprehensive wearable prototype was developed, featuring a mobile app that demonstrated the application of TinyCNN in real-world fall detection scenarios. Syed et al. [17] introduced a ConvLSTM-attention network that incorporated fall direction and severity as part of its classification process. The proposed system went beyond simple binary classification by addressing both non-binary fall detection and the combined recognition of activities of daily living. The approach employed attention mechanisms within a temporal modeling system to improve both time series classification accuracy and interpretability. This research enhances fall detection in ambient assisted living through advanced context-specific modeling.

Yu et al. [18] introduced a novel semi-supervised model, Semi-PFD, for pre-impact fall detection that reduces reliance on labeled fall data by leveraging abundant, easily obtainable data from activities of daily living (ADLs). This approach addressed the challenges of collecting and labeling fall events, which are rare and time-consuming to annotate. By incorporating unsupervised training, Semi-PFD effectively utilized limited fall data while maintaining detection performance, demonstrating its potential for practical deployment in wearable fall detection systems. Gangadhar et al. [19] proposed a wearable surveillance system designed to detect accidents at their onset and progression, triggering alarms to minimize harm and issuing external alerts upon impact. The authors introduced a composite classification framework that combines Random Forest (RF) for effective feature extraction from speed and inertial data and Support Vector Machines (SVM) for estimation and classification. Each module in the system targets specific fall stages, enabling context-aware detection. This approach aimed to enhance accident prevention and response for older individuals through intelligent monitoring. Yi et al. [20] proposed an unsupervised fall detection method based on a denoising LSTM-based convolutional variational autoencoder (CVAE) to address the scarcity of real-world fall data and the challenges of deploying deep learning models on resource-constrained wearable devices. The authors introduced data debugging and hierarchical data balancing techniques to enhance model performance and efficiency. The resulting lightweight model was designed for real-time fall detection, making it well-suited for practical integration into wearable healthcare systems for the elderly. Mao et al. [21] designed a multilayer mobile edge computing (MLMEC) framework for fall detection, addressing the limitations in model size and data transmission latency commonly faced in traditional edge-to-cloud architectures. The MLMEC structure distributes neural network models across multiple stations, enabling hierarchical processing based on device capability. To enhance detection accuracy at the edge, a knowledge distillation (KD) technique was incorporated, where powerful back-end models guide lightweight front-end models. This approach strikes a balance between accuracy and latency, making it well-suited for real-time fall detection in assistive systems.

3 Methodology

Time-series data collected from different sources often exhibit domain shifts, where variations arise due to differences in sensor specifications, environmental conditions, or subject-specific factors. In fall detection, domain shifts occur when models trained on one dataset fail to generalize effectively to another due to variations in signal distributions. This is particularly evident in datasets like KFall [22] and SisFall [23], where differences in sensor placement, sampling rates, and motion patterns introduce significant cross-domain discrepancies.

Conventional deep learning models rely on the assumption that training and testing data share the same distribution. However, in real-world fall detection applications, models must be robust to unseen target domains without requiring extensive labeled data from each new environment. To address this, Unsupervised Domain Adaptation (UDA) is employed, leveraging transfer learning techniques to improve generalization across datasets without requiring labels from the target domain.

Recent advancements in domain adaptation for sensor-based activity recognition have explored methods like adversarial learning, contrastive learning, and self-supervised feature alignment [24–26]. Inspired by these approaches, a ResNet-Transformer (ResT) based feature extractor, in conjunction with a DualAlign loss, is proposed, facilitating more effective cross-domain feature alignment while preserving discriminative features. The proposed model follows a structured adaptation pipeline, as illustrated in Fig. 1.

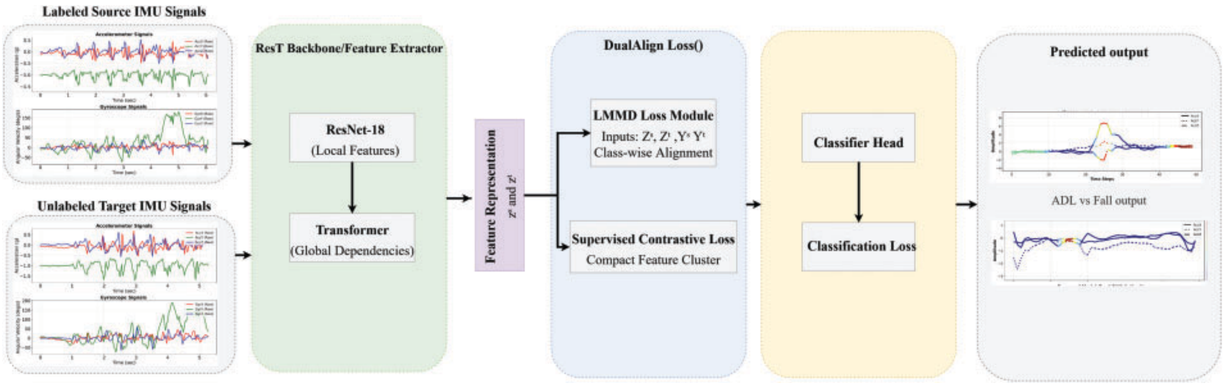


Figure 1: Overview of the proposed UDA framework with ResT and DualAlign Loss

3.1 Problem Formulation

Let $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ denote the labeled source domain dataset, where $x_i^s \in \mathbb{R}^{C\times L}$ represents a time-series sample with C channels and sequence length L, and $y_i^s \in \{1, \ldots, K\}$ denotes the corresponding label among K classes. Similarly, let $\mathcal{D}_t = \{(x_i^t)\}_{i=1}^{N_t}$ be the unlabeled target domain dataset.

The main objective is to train a model $f_\theta(x)$ parameterized by θ that generalizes well from $\mathcal{D}_s$ to $\mathcal{D}_t$, despite distribution shifts:

$$P_s(x, y) \neq P_t(x, y) \quad (1)$$

where $P_s(x, y)$ and $P_t(x, y)$ represent the joint distributions of features and labels in source and target domains, respectively.

To address this, a UDA is employed by minimizing both the classification loss in the source domain and the domain discrepancy between $\mathcal{D}_s$ and $\mathcal{D}_t$ through feature alignment.

The objective of UDA is to learn a domain-invariant feature representation $\Phi(\cdot)$ such that the classifier trained on the source domain can generalize effectively to the target domain.

The classification problem in UDA can be formulated as:

$$\min_{\theta_f, \theta_c} \mathbb{E}(x^s, y^s) \sim \mathcal{D}s\left[\mathcal{L}\mathrm{CE}(C(\Phi(x^s)), y^s)\right] + \lambda\mathcal{L}\mathrm{DA}(\Phi(x^s), \Phi(x^t)) \tag{2}$$

where $\Phi(x)$ is the feature extractor (ResT in our case), parameterized by θ_f; $C(\cdot)$ is the classifier, parameterized by θ_c; $\mathcal{L}_{\mathrm{CE}}$ represents the cross-entropy loss for source domain classification; $\mathcal{L}_{\mathrm{DA}}$ is the domain alignment loss, ensuring the source and target feature distributions are aligned; and λ is a trade-off parameter balancing classification and domain adaptation losses.

In the proposed DualAlign Loss, Local Maximum Mean Discrepancy (LMMD) [27] and Supervised Contrastive Loss (SupCon) [14] are combined to learn representations that are both domain-invariant and class-discriminative. Specifically, LMMD extends traditional MMD [28] by incorporating class-conditional alignment, thereby ensuring that the source and target domain distributions align not only globally but also locally within each class. This facilitates precise class-wise adaptation and enhances generalization capability. Simultaneously, SupCon optimizes semantic compactness by encouraging representations from the same class to cluster closely while maintaining clear separability between different classes.

3.2 Domain Adaptation Strategy

Domain adaptation relies on extracting transferable features that bridge the gap between source and target domains. In fall detection, dataset discrepancies arise due to variations in sensor types, sampling rates, and human motion patterns across different studies. KFall and SisFall, for example, exhibit different signal characteristics (data shift) due to differences in data collection points and protocols. These domain shifts lead to degraded model performance when applying a classifier trained on one dataset to another.

To mitigate these shifts, UDA aims to transfer knowledge from a labeled source domain to an unlabeled target domain. The key questions in domain adaptation are:

- **What to transfer?** In the proposed setting, the main aim is to transfer discriminative feature representations that remain invariant across datasets while preserving class-specific characteristics.
- **How to transfer?** To effectively learn domain-invariant yet class-discriminative representations, a Residual Transformer Network (ResT) is employed as a feature extractor. To facilitate efficient knowledge transfer, a loss function termed DualAlign Loss is proposed, which simultaneously ensures class-wise domain alignment and enhanced feature discriminability.

3.2.1 Integration of ResNet and Transformers: The ResT Architecture

In cross-domain applications, Convolutional Neural Networks (CNNs) have been widely adopted due to their ability to extract spatial features through hierarchical representations. However, CNN architectures struggle to effectively capture the long-range temporal dependencies inherent in time-series data, primarily due to their localized receptive fields [29]. To address this, Residual Networks (ResNet) introduced skip connections, facilitating improved gradient flow and enabling deeper network structures [30]. ResNet's residual mapping significantly improves convergence stability and generalization over several demanding tasks, including image and time-series classification. Additionally, Transformers, which employ self-attention mechanisms, enable networks to capture global context and long-range dependencies more effectively by simultaneously weighing the significance of different input tokens [31]. Motivated by these developments, ResNet and Transformer designs are combined to create the ResNet-Transformer Network (ResT). This hybrid model combines the benefits of both architectures: ResNet's ability to learn residual mappings and Transformers' expertise in capturing global dependencies via self-attention mechanisms. In this study, ResNet-18 was selected as the feature extractor backbone to balance depth and computational efficiency while minimizing the risk of overfitting on limited datasets. The Transformer encoder was configured with two layers, each employing 64-dimensional attention heads and a 256-dimensional embedding size, as determined through empirical tuning. This configuration provided an optimal trade-off between representation capacity, model complexity, and generalization performance across domains.

Mathematically, given an input sequence $x \in \mathbb{R}^{C \times L}$, where C is the number of channels and L is the sequence length, the ResT model processes x as follows:

1. Residual Block Processing:

$$h_{\text{res}} = \text{ResNetBlock}(x)$$

Here, h_{res} represents the output of the residual blocks, capturing local features.

2. Transformer Encoding:

$$h_{\text{trans}} = \text{TransformerEncoder}(h_{\text{res}})$$

The Transformer encoder applies self-attention mechanisms to model global dependencies within the sequence.

3. Classification Head:

$$y = \text{Softmax}(Wh_{\text{trans}} + b)$$

where W and b are learnable parameters, and y denotes the output class probabilities.

In the designed framework, the ResT architecture is configured with the following settings:

- Residual Blocks: Comprising three ResNet blocks, each with convolutional layers followed by batch normalization and rectified linear unit (ReLU) activation functions.

- Transformer Encoder: Consisting of two layers with multi-head self-attention mechanisms, each head having a dimensionality of 64, and position-wise feed-forward networks.

- Sequence Length (L): Set to 128, aligning with the temporal resolution of time-series data.

- Embedding Dimension (d_{model}): Configured to 256, ensuring a rich representation of input features.

This configuration ensures that the model effectively captures both local patterns and global dependencies in the data, facilitating robust knowledge transfer across domains.

3.2.2 DualAlign Loss: Joint Distribution Alignment with Contrastive Discriminability

A loss function called extbfDualAlign Loss is presented to acquire domain-invariant yet class-discriminative representations efficiently. Local Maximum Mean Discrepancy (LMMD) and Supervised Contrastive Learning (SupCon) combine local distribution alignment and semantic compactness, respectively, to achieve this objective. Formally, the complete objective for the proposed method is defined as:

$$\mathcal{L}_{\text{DualAlign}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{LMMD}} \cdot \mathcal{L}_{\text{LMMD}} + \lambda_{\text{SupCon}} \cdot \mathcal{L}_{\text{SupCon}} \tag{3}$$

where $\mathcal{L}_{cls}$ is the standard cross-entropy loss for source classification. $\mathcal{L}_{lmmd}$ is the Local Maximum Mean Discrepancy (LMMD) loss, which aligns feature distributions across domains at a class level. $\mathcal{L}_{supcon}$ is the Supervised Contrastive Loss, which enhances intra-class compactness and inter-class separation. λ_{lmmd} and λ_{supcon} are hyperparameters controlling the relative contributions of the alignment and contrastive losses.

The LMMD loss is formally written as:

$$\mathcal{L}_{\text{LMMD}} = \sum_{k=1}^{K} \left\| \frac{1}{n_s^k} \sum_{i=1}^{n_s^k} \phi(x_i^s) - \frac{1}{n_t^k} \sum_{j=1}^{n_t^k} \phi(x_j^t) \right\|^2 \tag{4}$$

where p_k represents the prior probability of class k. N_s^k and N_t^k are the number of source and target samples for class k. $\phi(\cdot)$ maps features to a high-dimensional reproducing kernel Hilbert space (RKHS) to improve alignment.

By incorporating class-specific weighting, LMMD achieves finer-grained domain alignment, resulting in improved cross-domain adaptation. To further enhance feature representations, Supervised Contrastive Learning (SupCon) is incorporated, which encourages samples of the same class to cluster together while pushing samples from different classes apart. The SupCon loss is defined as:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum\limits_{a \in A(i)} \exp(z_i \cdot z_a / \tau)} \tag{5}$$

where z_i is the normalized feature vector for sample i. τ is the temperature parameter. $P(i)$ represents the set of positive pairs (same-class samples). $A(i)$ represents the full batch containing both positive and negative samples.

The DualAlign Loss balances domain alignment and class discriminability through LMMD and SupCon components. Overemphasis on either term may compromise the other objective. To mitigate this, the loss weights are empirically tuned, ensuring that both components contribute complementarily and avoid over-alignment or reduced discriminability. This balance improved cross-domain generalization as confirmed in the ablation studies.

By integrating ResT for feature extraction, DualAlign Loss for class-wise domain adaptation, the proposed approach achieves robust feature transferability across datasets. This leads to improved generalization in real-world fall detection applications. A visual representation of the designed framework is illustrated in Fig. 1.

4 Dataset Preparation and Evaluation Metrics

The effectiveness of a fall detection model depends on the diversity and quality of the datasets used for training and evaluation. In the proposed study, the KFall and SisFall datasets are utilized, which contain time-series signals collected from wearable sensors. However, these datasets exhibit cross-domain variations due to differences in sensor placement, sampling rates, hardware specifications, and participant demographics. To develop a robust model capable of generalizing across domains, these discrepancies must be addressed through careful preprocessing and unification.

4.1 Datasets Overview

In the context of wearable-based fall detection, the choice of datasets plays a critical role in model development and evaluation. Two key challenges in this domain are: (i) selecting datasets that include both fall and ADL events, and (ii) applying consistent preprocessing techniques to ensure compatibility between datasets. To enable effective domain adaptation and transfer learning, it is essential that the datasets exhibit comparable sensor placement and capture diverse subject activities under realistic conditions. This study utilizes two widely adopted publicly available datasets: KFall (Korean Fall Detection Dataset) [22] and SisFall (SisBen Fall Dataset) [23].

The KFall dataset comprises accelerometer data collected from wearable devices worn by subjects performing both falls and various ADL scenarios. In contrast, the SisFall dataset includes inertial measurement unit (IMU) signals acquired from a broader age range of participants, particularly older adults and younger individuals, recorded under controlled environments using multiple wearable sensors.

Several inherent differences exist between the two datasets, including variations in sampling frequency, sensor specifications, number of subjects, and the complexity and duration of recorded activities. These discrepancies result in a domain shift, a divergence in data distributions. Training a model on one dataset and directly testing on another typically leads to performance degradation. This motivates the use of domain adaptation techniques, which aim to bridge this gap and enable robust cross-domain generalization.

Table 1 summarizes key characteristics of these datasets, adapted from [4].

Table 1: Comparison of fall detection datasets used in this study

Dataset	Ref.	No. of subjects	ADL/Fall types	Samples (ADLs/Falls)	Trace duration (s)	Sampling rate (Hz)	Sensor placement	Sensor used
KFall	[22]	32	21/15	**5075** (2729/2346)	[2.03–40.86]	100	Waist	1 × Acc
SisFall	[23]	19–75	19/15	**4505** (2707/1798)	[9.99–180.0]	200	Waist	2 × Acc, 1 × Gyro

4.2 Preprocessing and Unification

To enable cross-dataset learning, the properties of selected datasets must be aligned in terms of Resampling Frequency, Windowing Strategy, Normalization and Scaling, and Label Unification.

4.2.1 Resampling Frequency: Standardizing to 50 Hz

The selected datasets are recorded at different sampling rates, KFall: 100 Hz and SisFall: 200 Hz. Maintaining different frequencies causes misalignment in time-series patterns, which affects the model's learning performance. The solution is to downsample both datasets to a reasonable frequency of 50 Hz using the downsampling factor. Resampling to 50 Hz reduces the temporal resolution compared to the original recordings, previous studies [4] have demonstrated that this rate is sufficient to capture the essential dynamics of fall events, including sharp acceleration peaks and abrupt transitions. The preliminary analysis confirmed that fall-specific signatures, particularly the impact phase characterized by high-magnitude acceleration spikes, remain prominently visible and are not compromised at this frequency. Additionally, by employing a peak-centered windowing strategy, the risk of missing critical fall-related patterns due to downsampling is further mitigated. The downsampling factor D is defined as:

$$D = \frac{f_{original}}{f_{target}} \tag{6}$$

$$X' = \text{Resample}(D, f_{target}) \tag{7}$$

where $f_{target} = 50$ Hz which takes every second sample point for Kfall and 4th for Sisfall, effectively resampling the frequency.

4.2.2 Windowing Strategy

In both datasets, ADL activities exhibit continuous sensor patterns, whereas falls show sudden spikes in readings. To effectively distinguish fall events, the windowing strategy ensures each segment captures the fall impact. ADLs, being more uniform, remain largely unaffected by segmentation, while long-duration activities (>25 s) with single trials are split into multiple samples to maintain balance across ADL categories. To ensure sufficient motion context while maintaining a computationally feasible input size for deep learning models, a 1-s window is selected for segmentation. Given the 50 Hz sampling rate, each window consists of 50 consecutive sensor readings. Each event in the dataset is annotated with a start and end index. However, the actual fall impact may occur at an unknown time within this range. To ensure each extracted window captures the fall impact, the window is centered on the peak acceleration event.

To ensure that each sample represents a meaningful fall event, a fixed-length window of 1s (50 samples at 50 Hz) was extracted around the peak acceleration. This was determined using the Euclidean norm of the accelerometer signals:

$$a_{norm}(t) = \sqrt{AccX^2(t) + AccY^2(t) + AccZ^2(t)} \tag{8}$$

The peak index is determined as:

$$t_{peak} = \arg\max a_{norm}(t) \tag{9}$$

Ensuring Window Coverage the peak index t_{peak} within the fall period $[start, end]$ was selected, and a 1s window centered around this peak was extracted:

$$x_{start} = \max(0, \min(start + \ t_{peak} - \frac{W}{2}, \text{len}(data) - W)) \tag{10}$$

$$x_{end} = x_{start} + 50 \tag{11}$$

where W is the window length in samples, this guarantees that all windows are exactly the same number of sample points in length, regardless of edge cases. This approach prioritizes the most dynamic phase of the fall event, ensuring the window captures the critical impact moment. For falls with multiple peaks or gradual onset, the strategy ensures that the highest-intensity segment is always selected, maintaining input consistency and maximizing discriminative potential. Although this may omit some pre-fall or post-fall context, it aligns with established practices in fall detection literature, striking a balance between window standardization and sensitivity to impactful motion patterns.

4.2.3 Normalization, Label Unification, and Train-Test Splitting

Normalization: Z-score Scaling—To ensure consistency across datasets, a Z-score normalization is applied to each sensor axis:

$$X_{\text{norm}} = \frac{X - \mu}{\sigma} \tag{12}$$

where μ and σ represent the mean and standard deviation of each feature, respectively. This transformation ensures that all sensor signals have zero mean and unit variance, improving robustness to differences in sensor calibration, subject variability, and environmental noise [32]. A more effective feature alignment is facilitated by standardizing both datasets, thereby minimizing the impact of domain discrepancies.

Unifying Labels: ADL vs Fall—The two datasets contain multiple activity classes, including various ADLs and different types of falls. However, to formulate a binary classification problem, the labels are mapped as ADL → 0 and Fall → 1.

This conversion ensures consistency across datasets, simplifying the task to a fall detection model rather than a multi-class activity recognition problem.

Train-Test Splitting: Stratified Sampling—To ensure a fair evaluation of model performance, both KFall and SisFall datasets are independently split into 70% training and 30% testing subsets, maintaining their original class distributions through stratified sampling. This guarantees an even representation of ADL and fall instances in both splits, preventing class imbalance issues.

Let D_{kfall} and D_{sisfall} denote the datasets with N_{kfall} and N_{sisfall} total samples, and let p_{fall} and p_{adl} be the proportions of falls and ADLs in each dataset. The train-test split is performed as:

$$D_{\text{kfall, train}}, D_{\text{kfall, test}} = \text{Stratify}(D_{\text{kfall}}, \text{ratio} = 70:30, p_{\text{fall}}, p_{\text{adl}}) \tag{13}$$

$$D_{\text{sisfall, train}}, D_{\text{sisfall, test}} = \text{Stratify}(D_{\text{sisfall}}, \text{ratio} = 70:30, p_{\text{fall}}, p_{\text{adl}}) \tag{14}$$

Stratified sampling ensures that the relative frequency of fall and ADL events is preserved within each dataset's train and test subsets. This approach prevents any data leakage while ensuring that the training data remains representative of real-world distributions. To further illustrate the dataset distribution, Fig. 2 presents a breakdown of the sample counts for both datasets, comparing ADL and fall instances.

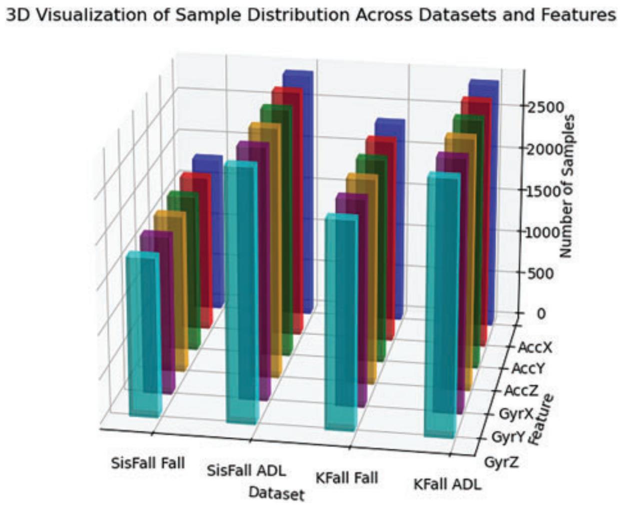
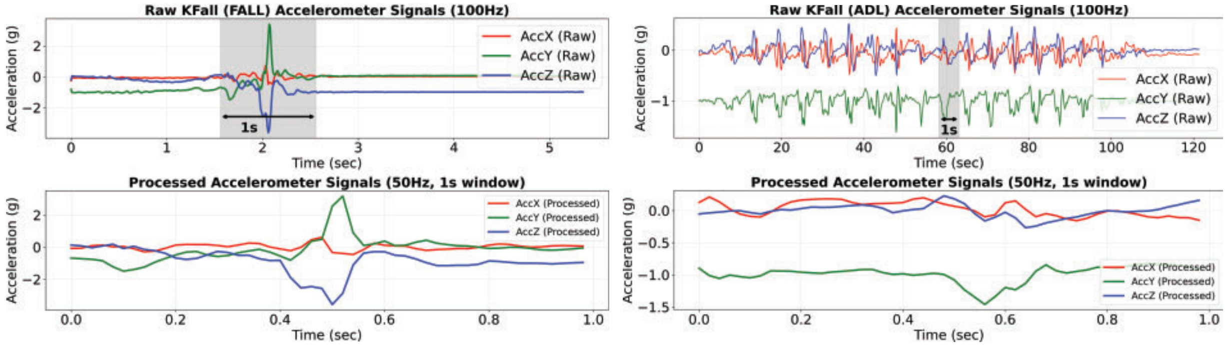


Figure 2: Sample distribution of ADL and Fall instances across KFall and SisFall datasets

Fig. 3 illustrates examples of preprocessed accelerometer data for both FALL and ADL scenarios. In the FALL case (Fig. 3a), the segment is extracted around the peak fall motion, capturing the most informative transition phase. For ADL activities (Fig. 3b), a centered 1-s window is selected to represent steady-state motion, such as walking. These extracted windows are resampled to 50 Hz and used as standardized inputs for the proposed classification models.



(a) FALL sample with raw signal and 1 s extracted segment around the fall peak.

(b) ADL sample with raw walking signal and centered 1 s segment.

Figure 3: Examples of raw accelerometer signals and extracted 1-s windows used for classification. (**a**) A representative *FALL* instance from the KFall dataset, where a 1 s window is extracted around the fall peak (resampled to 50 Hz). (**b**) A representative *ADL* instance, where a centered 1 s window is extracted during steady walking activity. These extracted segments are highlighted in the raw signals and used as standardized inputs to the classification model

4.3 Evaluation Metrics

To comprehensively evaluate the proposed domain adaptation approach for fall detection, several performance metrics are utilized. Given the inherent class imbalance in fall detection datasets, where fall events are significantly less frequent than ADLs, Recall and F1-score are prioritized. These metrics are crucial for assessing a model's ability to accurately identify fall events without being biased by the majority class. The following key metrics are reported for both class-wise and overall model performance:

- **Accuracy:** Measures overall correctness of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

- **Precision:** The proportion of correctly predicted falls among all fall predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

- **Recall (Sensitivity):** Measures the ability to detect actual falls, crucial for fall detection.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

- **F1-score:** The harmonic mean of precision and recall, balancing false positives and false negatives.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

- **AUROC (Area Under ROC Curve):** Evaluates the model's ability to distinguish falls from ADLs, measuring robustness across varying decision thresholds.
- **Confusion Matrix (CM):** To provide further interpretability, confusion matrices are included, which highlight the distribution of predictions across True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Since missing a fall event can have serious consequences in real-world applications, a high recall with low false negatives is crucial. Therefore, the most important evaluation metrics in the proposed study are recall and F1-score.

5 Experiments

5.1 Implementation Details

The implementation of the proposed scheme is based on AdaTime [33], an open-source library specifically designed for domain adaptation in time-series tasks. AdaTime provides a structured framework for implementing UDA models, enabling efficient handling of cross-domain discrepancies in sensor-based fall detection. Experiments were conducted on an NVIDIA RTX 4080 Super GPU with 16 GB VRAM, utilizing PyTorch 2.0 and CUDA 12.1 for efficient training. Setup is supported by an Intel Core i7-9300F CPU and 64 GB RAM. Each model was trained for 40 epochs with a batch size of 32, applying a weight decay of 1e-4 for regularization. The initial learning rate was 1e-3, which decayed by a factor of 0.5 every 10 epochs. For the proposed DualAlign Loss, the LMMD loss weight (λ_{LMMD}) and SupCon loss weight (λ_{SupCon}) were selected based on prior works on deep subdomain adaptation network (DSAN) [27] and supervised contrastive learning [14]. Initial values were adopted from the literature and refined using grid search to balance domain alignment and class discrimination. The optimal settings were found to be $\lambda_{\text{LMMD}} = 2.1$ and $\lambda_{\text{SupCon}} = 0.1$, providing the best trade-off between domain invariance and class-wise compactness while avoiding overfitting. Other training hyperparameters, such as learning rate and regularization were selected based on empirical validation and aligned with settings used in previous UDA studies for time-series data. The model was trained using source-only initialization, followed by unsupervised adaptation using training on with source domain and testing on the target domain. Model selection was performed based on the highest F1-score on the test set, ensuring optimal generalization to unseen target distributions.

5.2 Benchmark Models

UDA techniques aim to improve generalization across datasets by aligning feature distributions of the source and target domains. In fall detection, models trained on one dataset often fail to perform well on another due to domain shifts, necessitating the use of domain adaptation methods.

As a baseline, widely used methods from previous UDA studies on sensor-based human activity recognition were considered. Among these methods, the Domain Adversarial Neural Network (DANN) has been frequently employed for domain alignment through adversarial training [34]. Given its proven effectiveness, DANN with a CNN backbone was chosen as a benchmark model. In addition, comparisons were made with another adversarial method, CDAN [35], as well as popular discrepancy-based UDA methods, including Deep CORAL [12] and MMD [28], to evaluate their performance in cross-dataset fall detection.

To ensure a fair comparison across all domain adaptation methods, a standardized CNN-based feature extractor was utilized across all models. The feature extractor consists of three convolutional blocks, each comprising 1D convolutional layers, batch normalization, ReLU activation, and max-pooling operations, which progressively extract hierarchical temporal features from time-series data. The final output is passed through an adaptive average pooling layer, ensuring a fixed-length feature representation regardless of input sequence length.

This CNN backbone adheres to a well-established architecture for time-series classification, striking a balance between expressiveness and computational efficiency. It has been designed to capture both local temporal dependencies through convolutional layers and global representations using adaptive pooling. Each domain adaptation method is trained using this shared feature extractor, ensuring that performance differences arise from adaptation techniques rather than variations in backbone architecture.

By maintaining a consistent CNN-based feature extractor across methods, the impact of different domain adaptation strategies on model performance can be isolated and evaluated. This setup enables robust evaluation of how each adaptation approach enhances the model's ability to generalize across domain shifts in fall detection datasets.

To ensure fair evaluation, the mean ± standard deviation over five runs was reported. Performance of various UDA methods under two domain adaptation (source → target) settings: SisFall → KFall and KFall → SisFall. Table 2 displays the overall AUROC, Accuracy, F1-score, Precision, and Recall, focusing on combined class performance to provide a holistic view of model effectiveness.

Table 2: Benchmark comparison for domain adaptation from SisFall to KFall and vice versa. Results are reported as mean ± standard deviation over multiple runs

Model	**SisFall → KFall**					**KFall → SisFall**				
	Accuracy	**F1**	**AUROC**	**Precision**	**Recall**	**Accuracy**	**F1**	**AUROC**	**Precision**	**Recall**
Source only	$0.619_{\pm 0.04}$	$0.607_{\pm 0.05}$	$0.710_{\pm 0.06}$	$0.655_{\pm 0.05}$	$0.619_{\pm 0.04}$	$0.801_{\pm 0.01}$	$0.810_{\pm 0.02}$	$0.843_{\pm 0.06}$	$0.836_{\pm 0.03}$	$0.801_{\pm 0.01}$
DeepCORAL	$0.640_{\pm 0.04}$	$0.630_{\pm 0.05}$	$0.712_{\pm 0.07}$	$0.672_{\pm 0.02}$	$0.640_{\pm 0.04}$	$0.734_{\pm 0.03}$	$0.737_{\pm 0.04}$	$0.817_{\pm 0.04}$	$0.734_{\pm 0.04}$	$0.735_{\pm 0.03}$
DANN	$0.816_{\pm 0.002}$	$0.819_{\pm 0.001}$	$0.865_{\pm 0.06}$	$0.829_{\pm 0.001}$	$0.816_{\pm 0.002}$	$0.936_{\pm 0.006}$	$0.926_{\pm 0.004}$	$0.984_{\pm 0.006}$	$0.921_{\pm 0.004}$	$0.936_{\pm 0.006}$
CDAN	$0.816_{\pm 0.003}$	$0.819_{\pm 0.003}$	$0.794_{\pm 0.06}$	$0.820_{\pm 0.002}$	$0.816_{\pm 0.003}$	$0.950_{\pm 0.004}$	$0.939_{\pm 0.006}$	$0.978_{\pm 0.01}$	$0.933_{\pm 0.005}$	$0.950_{\pm 0.004}$
Proposed	$0.961_{\pm 0.02}$	$0.961_{\pm 0.01}$	$0.980_{\pm 0.005}$	$0.961_{\pm 0.02}$	$0.961_{\pm 0.03}$	$0.922_{\pm 0.01}$	$0.927_{\pm 0.03}$	$0.940_{\pm 0.05}$	$0.939_{\pm 0.02}$	$0.922_{\pm 0.04}$

5.2.1 Key Observations

The results in Table 2 demonstrate the substantial impact of domain shift on fall detection, reinforcing the necessity of Unsupervised Domain Adaptation (UDA). The Source-Only model, trained solely on the source dataset, exhibits poor generalization, particularly in the SisFall → KFall setting. This highlights inherent distribution mismatches arising from differences in sensor placement, sampling rates, and subject demographics.

Among discrepancy-based methods, DeepCORAL yields only minor improvements over the Source-Only baseline, indicating that aligning global feature distributions alone is insufficient for robust adaptation. Adversarial-based methods such as DANN and CDAN demonstrate better overall performance by explicitly modeling domain invariance through adversarial training. While CDAN achieves notable gains in the KFall → SisFall scenario, its improvements are less substantial in the reverse adaptation (SisFall → KFall), suggesting sensitivity to dataset-specific factors.

In contrast, the proposed approach employing ResT combined with the novel DualAlign Loss consistently outperforms all other methods across both adaptation scenarios (SisFall → KFall and KFall → SisFall) in terms of key performance metrics such as F1-score and recall for both ADL and FALL classes (Table 3). Specifically, the proposed model achieves the highest F1-scores and recall rates, substantially improving fall-class detection, which is critical for practical deployment. Although the performance slightly varies between the two scenarios, the proposed scheme exhibits the best overall generalization capability, indicating enhanced robustness to cross-dataset variability.

Table 3: Class-wise results for domain adaptation benchmarks

Model	SisFall → KFall						KFall → SisFall					
	F1 (ADL)	F1 (Fall)	P (ADL)	P (Fall)	R (ADL)	R (Fall)	F1 (ADL)	F1 (Fall)	P (ADL)	P (Fall)	R (ADL)	R (Fall)
Source only	**0.724** ± 0.032	**0.490** ± 0.073	**0.628** ± 0.033	**0.683** ± 0.077	**0.855** ± 0.034	**0.383** ± 0.068	**0.866** ± 0.020	**0.755** ± 0.024	**0.824** ± 0.021	**0.847** ± 0.085	**0.915** ± 0.056	**0.688** ± 0.058
DeepCORAL	**0.732** ± 0.016	**0.528** ± 0.096	**0.648** ± 0.037	**0.696** ± 0.032	**0.845** ± 0.043	**0.435** ± 0.117	**0.790** ± 0.041	**0.677** ± 0.045	**0.798** ± 0.023	**0.671** ± 0.072	**0.784** ± 0.064	**0.685** ± 0.035
CDAN	**0.848** ± 0.002	**0.791** ± 0.004	**0.806** ± 0.004	**0.852** ± 0.002	**0.895** ± 0.002	**0.738** ± 0.006	**0.949** ± 0.005	**0.928** ± 0.006	**0.993** ± 0.002	**0.874** ± 0.012	**0.910** ± 0.010	**0.990** ± 0.003
DANN	**0.848** ± 0.001	**0.791** ± 0.002	**0.806** ± 0.002	**0.853** ± 0.002	**0.896** ± 0.003	**0.737** ± 0.005	**0.939** ± 0.003	**0.913** ± 0.006	**0.981** ± 0.010	**0.860** ± 0.005	**0.900** ± 0.005	**0.973** ± 0.015
Proposed	**0.965** ± 0.01	**0.958** ± 0.01	**0.969** ± 0.01	**0.954** ± 0.02	**0.961** ± 0.02	**0.962** ± 0.1	**0.947** ± 0.02	**0.908** ± 0.05	**0.926** ± 0.05	**0.951** ± 0.02	**0.971** ± 0.01	**0.873** ± 0.09

A deeper analysis of class-wise results (Table 3) reveals that, despite significant improvements in adaptation, fall detection remains inherently challenging due to class imbalance and inter-domain shifts. FALL-class performance consistently trails behind ADL-class results across all methods, underscoring the persistent difficulty posed by variability in sensor placement, activity patterns, and subject-specific noise.

These observations highlight that while adversarial approaches (DANN, CDAN) outperform basic distribution-alignment methods (DeepCORAL), fall detection tasks remain sensitive to domain-specific variations. Hence, robust generalization requires methods that effectively balance feature alignment, class separability, and domain invariance. Motivated by this insight, architectural refinements and enhanced loss functions were further explored in an ablation study (Section 5.3), with a particular focus on improving FALL-class adaptation and mitigating the challenges posed by domain shifts. To avoid redundancy, the CNN-only baseline is represented by the Source Only model evaluated in Section 5.2, which utilizes the same CNN-based feature extractor as the foundation for comparison. As this baseline provides a direct reference for assessing the benefits of the proposed hybrid ResT architecture, the ablation study here focuses on highlighting the incremental improvements brought by integrating ResT and DualAlign loss, rather than repeating the CNN-only baseline within the ablation space.

5.3 *Ablation Study Analysis*

The results in Table 4 demonstrate the impact of ResT and DualAlign in domain adaptation for fall detection. The individual and combined contributions of these components are analyzed using macro F1-score, AUROC, precision, and recall.

Table 4: Ablation study results for domain adaptation with ResT and DualAlign

Model	Accuracy		F1-Score		AUROC	Precision		Recall	
	ADL	Fall	ADL	Fall	Macro	ADL	Fall	ADL	Fall
					SisFall → KFall				
ResT	**0.947** $_{\pm 0.085}$	**0.874** $_{\pm 0.129}$	**0.925** $_{\pm 0.086}$	**0.901** $_{\pm 0.112}$	**0.940** $_{\pm 0.084}$	**0.905** $_{\pm 0.098}$	**0.933** $_{\pm 0.108}$	**0.947** $_{\pm 0.085}$	**0.874** $_{\pm 0.129}$
DualAlign	**0.961** $_{\pm 0.038}$	**0.893** $_{\pm 0.134}$	**0.940** $_{\pm 0.061}$	**0.917** $_{\pm 0.098}$	**0.924** $_{\pm 0.098}$	**0.923** $_{\pm 0.095}$	**0.949** $_{\pm 0.054}$	**0.961** $_{\pm 0.038}$	**0.893** $_{\pm 0.134}$
ResT + DualAlign	**0.961** $_{\pm 0.02}$1	**0.962** $_{\pm 0.018}$	**0.965** $_{\pm 0.012}$	**0.958** $_{\pm 0.015}$	**0.968** $_{\pm 0.006}$	**0.969** $_{\pm 0.015}$	**0.954** $_{\pm 0.02}$4	**0.961** $_{\pm 0.02}$1	**0.962** $_{\pm 0.018}$
					KFall → SisFall				
ResT	**0.983** $_{\pm 0.007}$	**0.876** $_{\pm 0.088}$	**0.954** $_{\pm 0.02}$6	**0.919** $_{\pm 0.052}$	**0.958** $_{\pm 0.038}$	**0.928** $_{\pm 0.048}$	**0.970** $_{\pm 0.013}$	**0.983** $_{\pm 0.007}$	**0.876** $_{\pm 0.088}$
DualAlign	**0.946** $_{\pm 0.044}$	**0.996** $_{\pm 0.003}$	**0.971** $_{\pm 0.02}$3	**0.958** $_{\pm 0.034}$	**0.988** $_{\pm 0.010}$	**0.961** $_{\pm 0.036}$	**0.924** $_{\pm 0.010}$	**0.946** $_{\pm 0.044}$	**0.996** $_{\pm 0.003}$
ResT + DualAlign	**0.971** $_{\pm 0.015}$	**0.873** $_{\pm 0.097}$	**0.947** $_{\pm 0.02}$5	**0.908** $_{\pm 0.052}$	**0.940** $_{\pm 0.038}$	**0.926** $_{\pm 0.051}$	**0.952** $_{\pm 0.02}$3	**0.971** $_{\pm 0.015}$	**0.873** $_{\pm 0.097}$

Impact of Residual Transformer (ResT): ResT significantly improves feature extraction and generalization by incorporating long-range dependencies. In the SisFall → KFall adaptation, ResT achieves an AUROC of 0.940, with class-wise F1-scores of 0.925 for ADL and 0.901 for Fall, indicating reduced feature misalignment. In the KFall → SisFall setting, AUROC improves to 0.958, confirming enhanced separability between ADL and Fall events.

Impact of DualAlign: DualAlign enhances intra-class compactness and inter-class separation, reinforcing robustness against domain shifts. In SisFall → KFall, class-wise F1-scores are 0.940 for ADL and 0.917 for Fall, while in KFall → SisFall, class-wise F1-scores are 0.971 for ADL and 0.958 for Fall, reflecting improved detection sensitivity.

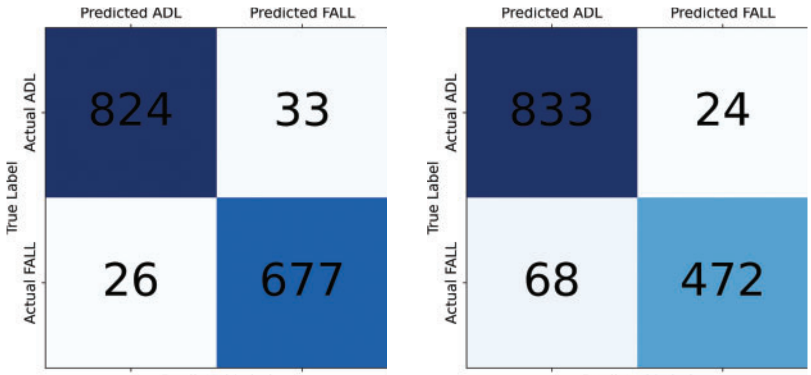
Combined Effect (ResT + DualAlign): The combination of ResT and DualAlign achieves the highest performance across all evaluated metrics, highlighting their synergistic effects. In the SisFall → KFall scenario, Fall-class F1 reaches 0.958 with an AUROC of 0.968, reflecting superior fall detection and high-confidence predictions. Similarly, in the KFall → SisFall adaptation, the approach achieves a Fall-class F1 of 0.908 and AUROC of 0.940, demonstrating robust generalization and effective fall-class discrimination across domains.

Further, the confusion matrices from the best-performing models for both adaptation settings are provided in Fig. 4, illustrating the model's final classification performance. These results confirm that ResT enhances feature alignment, while DualAlign strengthens class separability, and their combination minimizes the impact of domain shifts, making them highly effective for fall detection in real-world deployment scenarios. ResT improves overall classification performance, reducing false positives (FP) by enhancing feature alignment and minimizing misclassification of ADL instances as falls. DualAlign significantly reduces false negatives (FN) by improving intra-class compactness, thereby minimizing missed fall detections. The combined model (ResT + DualAlign) achieves the most balanced performance, with the highest TP rates, fewer FN (missed falls), and reduced FP (false alarms) compared to individual modifications. In the SisFall → KFall adaptation, FP drops to 33, while FN is 26.5, indicating improved detection of both ADL and falls. For the KFall → SisFall direction, FN remains higher at 68.6, suggesting that while DualAlign helps reduce missed falls, further refinements may be needed for domain generalization in this setting. These results confirm that ResT enhances domain alignment, DualAlign strengthens class separation, and their combination leads to a more robust UDA model for fall detection.

5.4 Qualitative Results

To gain deeper insights into the model's learning behavior, qualitative evaluations are presented through two key visualizations:

- **Visualization of Feature Space:** A t-distributed stochastic neighbor embedding (t-SNE) is presented to project high-dimensional feature representations into a 2D space, which allows for observing how well the model aligns source and target distributions.
- **Visualization of Activation Maps:** Grad-CAM is applied to highlight regions in input samples that contribute most to the model's predictions, illustrating the interpretability of the proposed approach.



(a) SisFall → KFall

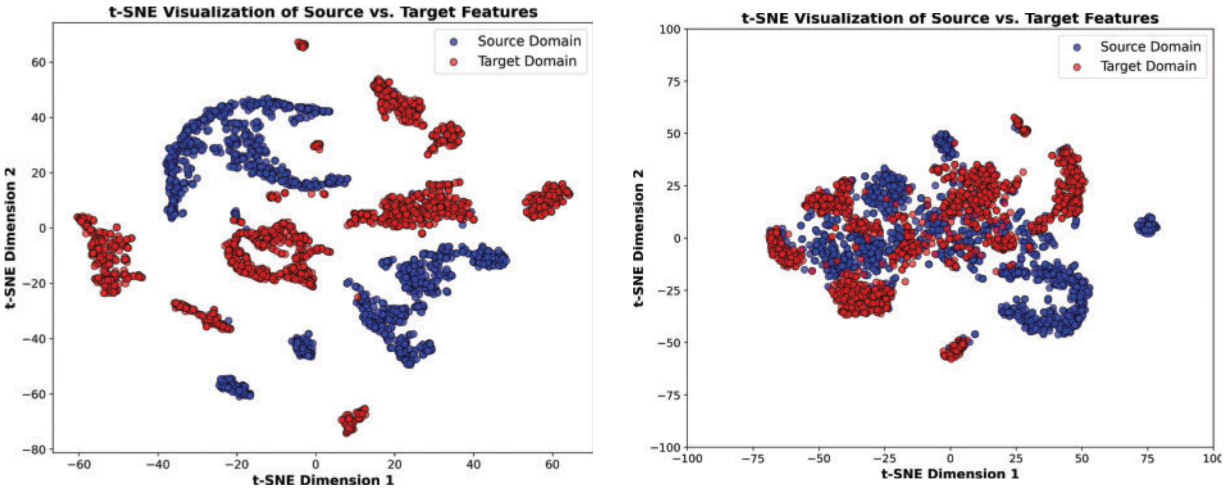
(b) KFall → SisFall

Figure 4: Confusion matrices for the best-performing models in both domain adaptation settings: (**a**) shows SisFall → KFall indicating effective fall classification, and (**b**) shows KFall → SisFall reinforcing model generalization across datasets

5.4.1 Visualization of Feature Space

t-SNE embeddings of extracted features provide an intuitive understanding of domain alignment. Fig. 5a,c shows the t-SNE projections of the baseline model for both adaptation settings (KFall → SisFall and SisFall → KFall). It is evident that the source and target distributions remain largely dispersed and misaligned, indicating that the model fails to learn domain-invariant representations.

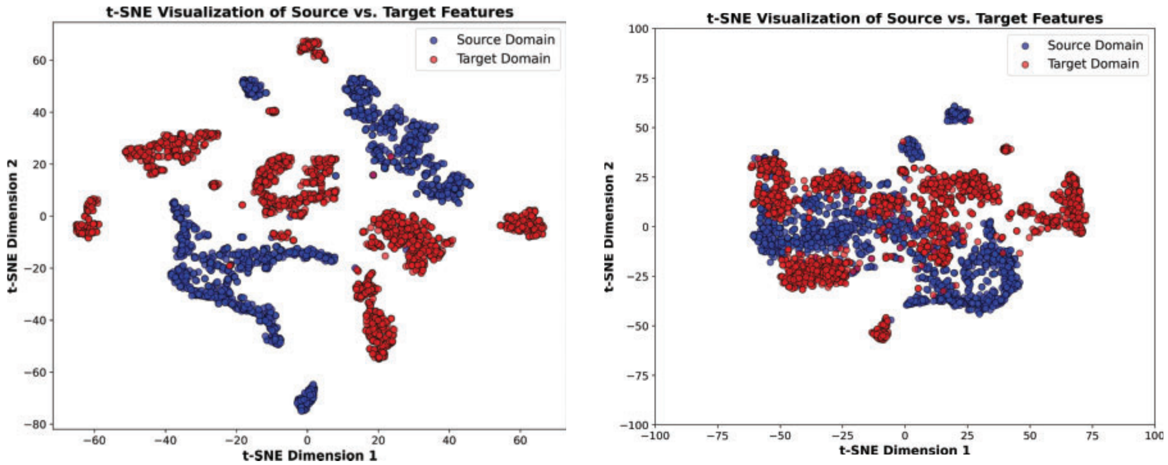
In contrast, Fig. 5b,d presents the t-SNE visualizations for the proposed ResT + DualAlign model. Unlike the baseline, the designed framework effectively clusters source and target features together, demonstrating a higher degree of alignment between domains. The improved structure in the feature space suggests that the model has learned to extract discriminative, domain-invariant representations, ultimately enhancing generalization across datasets.



(a) Baseline TSNE for KFall → SisFall

(b) Ours TSNE for KFall → SisFall

Figure 5: (Continued)



(c) Baseline TSNE for SisFall → KFall

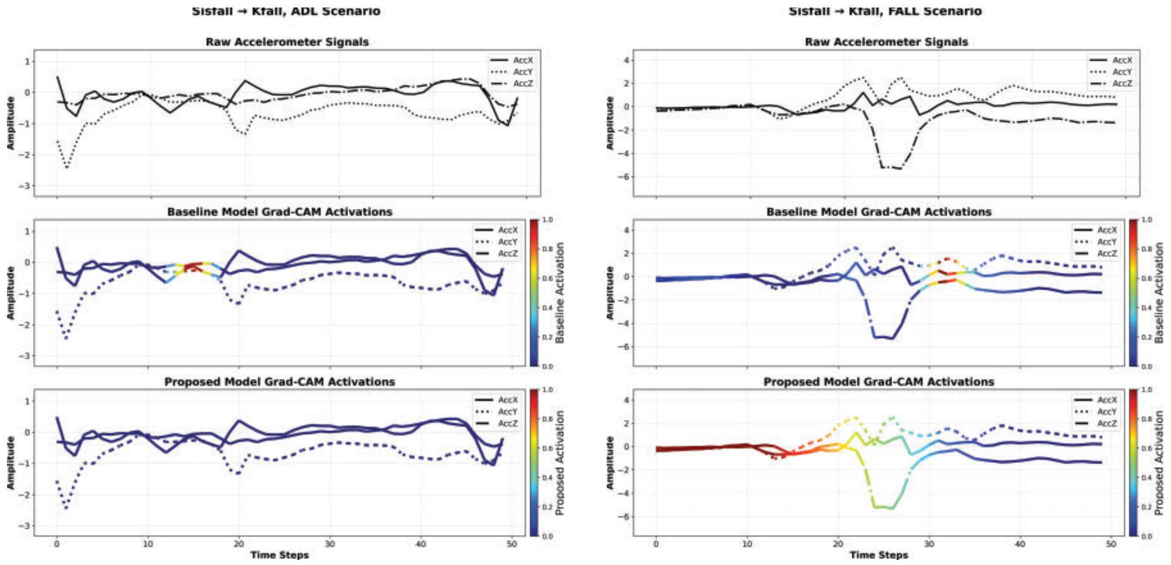
(d) Ours TSNE for SisFall → KFall

Figure 5: t-SNE visualizations of feature distributions. (**a**) and (**c**) represent the baseline model, showing high dispersion and poor alignment across domains. (**b**) and (**d**) show the proposed model, the source and target features form compact, overlapping clusters, indicating that the model learns domain-invariant representations

5.4.2 Visualization of Activation Maps

To further interpret the model's decision-making process, Grad-CAM is utilized to visualize class-specific activation maps. These maps highlight the most relevant regions in input samples that contribute to model predictions, providing insight into how effectively the model localizes fall-related patterns.

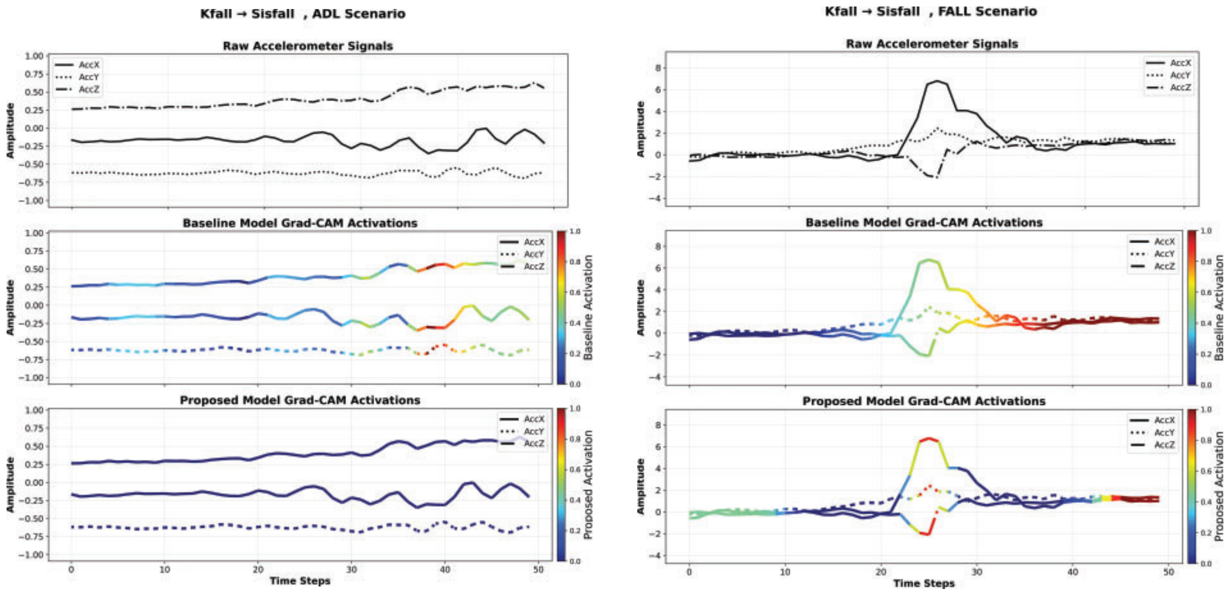
Fig. 6 shows Grad-CAM results for both adaptation settings and both classes (ADL and Fall). Often highlighting background noise instead of significant motion patterns, the Grad-CAM of the baseline model reveals erratic attention. By suppressing irrelevant features, the proposed model's Grad-CAM reveals concentrated activations that highlight important body areas associated with falls.



(a) ADL (SisFall → KFall)

(b) Fall (SisFall → KFall)

Figure 6: (Continued)



(c) ADL (KFall → SisFall)

(d) FALL (KFall → SisFall)

Figure 6: Grad-CAM visualization for baseline and proposed model across adaptation settings. The proposed model exhibits improved localization of class-relevant regions, demonstrating its effectiveness in identifying key motion patterns associated with falls

This qualitative evidence supports the proposed model's superior ability to learn task-relevant and domain-invariant features, leading to improved fall detection performance.

6 Conclusion

This study introduced a UDA framework specifically designed for wearable sensor-based fall detection, aiming to tackle significant cross-dataset generalization challenges. To bridge domain discrepancies arising from differences in sensor placement, sampling rates, and subject demographics, a comprehensive preprocessing pipeline was developed that standardizes resampling frequency, windowing strategy, normalization, label unification, and dataset splitting strategies across multiple benchmark datasets (KFall and SisFall).

A Residual Transformer Network (ResT) was introduced, representing an innovative hybrid architecture that effectively captures both local and global dependencies in sensor-based temporal data. Furthermore, a DualAlign Loss was proposed, which integrates Local Maximum Mean Discrepancy (LMMD) for class-wise domain alignment and Supervised Contrastive Loss (SupCon) to enhance semantic feature discrimination. Extensive evaluations against state-of-the-art domain adaptation methods, including Deep-CORAL, DANN, and CDAN, demonstrated the superiority of the proposed method, achieving substantial improvements in key performance metrics such as F1-score, recall, and AUROC. Critically, the analyses highlighted the persistent challenges posed by class imbalance and domain-specific factors in fall detection tasks. The combined ResT and DualAlign strategy significantly reduced both false alarms and missed detections, thus enhancing reliability and practical effectiveness in realistic deployment scenarios.

While the proposed method demonstrates strong generalization and improved domain adaptation performance, it has certain limitations. First, the approach assumes the availability of sufficient labeled source data and representative unlabeled target data, which may not always be feasible in real-world scenarios. Second, the method has been evaluated on controlled benchmark datasets (KFall and SisFall), which may not fully capture the variability present in more diverse or noisy environments. Additionally, the computational overhead introduced by the Residual Transformer and the contrastive learning component may limit deployment on resource-constrained devices. Future work can focus on lightweight adaptations and evaluation under more realistic, heterogeneous sensor setups. The future research directions include extending the framework to additional sensor modalities, exploring few-shot and incremental learning paradigms for broader applicability, and validating the approach in large-scale, real-world deployments. Ultimately, the proposed methodology makes a significant contribution to developing robust and scalable fall detection solutions, which are crucial for ensuring the safety of the elderly and effective healthcare monitoring.

Acknowledgement: The authors extend their appreciation to the King Salman Center for Disability Research for funding this work through Research Group no. KSRG-2024-430.

Funding Statement: This work is funded by the King Salman Center for Disability Research through Research Group no. KSRG-2024-430.

Author Contributions: Nadhmi A. Gazem: Writing—original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. Sultan Noman Qasem: Writing—original draft, Visualization, Methodology, Investigation, and Formal analysis. Umair Naeem: Review & editing, Visualization, Methodology. Shahid Latif: Review & editing, Software, Project administration. Ibtehal Nafea: Writing—review & editing, Visualization, Validation. Faisal Saeed: Writing—original draft, Software, Methodology, Formal analysis. Mujeeb Ur Rehman: review & editing, Writing—original draft, Supervision, Project administration, Methodology, Conceptualization, Visualization, and Validation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We are happy to share the processed datasets and Jupyter Notebooks of the proposed scheme for research purposes upon request, subject to the approval of our research group.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Declaration of Generative AI in Scientific Writing: The authors state that Grammarly's AI tool was used solely for refining the English language in a few sections.

References

1. Tran Q. Worldwide dementia cases to triple by 2050 to over 150 million people; 2022 [cited 2025 Jun 22]. Available from: https://www.alzheimersresearchuk.org/news/worldwide-dementia-cases-to-triple-by-2050-to-over-150-million/.
2. Kehrer-Dunlap AL, Keleman AA, Bollinger RM, Stark SL. Falls and Alzheimer disease. Adv Geriatr Med Res. 2024;6(1):e240001.
3. WHO. Falls. 2021 [cited 2025 Mar 21]. Available from: https://www.who.int/news-room/fact-sheets/detail/falls.
4. Silva CA, Casilari E, García-Bermúdez R. Cross-dataset evaluation of wearable fall detection systems using data from real falls and long-term monitoring of daily life. Measurement. 2024;235:114992. doi:10.1016/j.measurement.2024.114992.
5. Wang X, Ellul J, Azzopardi G. Elderly fall detection systems: a literature survey. Front Rob AI. 2020;7:71. doi:10.3389/frobt.2020.00071.
6. Liu J, Li X, Huang S, Chao R, Cao Z, Wang S, et al. A review of wearable sensors based fall-related recognition systems. Eng Appl Artif Intell. 2023;121:105993. doi:10.1016/j.engappai.2023.105993.
7. Newaz NT, Hanada E. The methods of fall detection: a literature review. Sensors. 2023;23(11):5212. doi:10.3390/s23115212.

8. Purwar A, Chawla I. A systematic review on fall detection systems for elderly healthcare. Multimed Tools Appl. 2023 Oct;83(14):43277–302. doi:10.1007/s11042-023-17190-z.
9. Paul SK, Miah ASM, Paul RR, Hamid ME, Shin J, Rahim MA. IoT-based real-time medical-related human activity recognition using skeletons and multi-stage deep learning for healthcare. arXiv:2501.07039. 2025.
10. Wen S, Chen Y, Ma Y, Guo S, Gu Y, Qin X, et al. Time series adaptation network for sensor-based cross domain human activity recognition. In: 2023 International Joint Conference on Neural Networks (IJCNN); 2023 Jun 18–23; Gold Coast, QLD, Australia. p. 1–8.
11. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. arXiv:1706.03762. 2023.
12. Sun B, Saenko K. Deep CORAL: correlation alignment for deep domain adaptation. arXiv:1607.01719. 2016.
13. Wang J, Chen Y, Gu Y, Xiao Y, Pan H. SensoryGANs: an effective generative adversarial framework for sensor-based human activity recognition. In: 2018 International Joint Conference on Neural Networks (IJCNN); 2018 Jul 8–13; Rio de Janeiro, Brazil. p. 1–8.
14. Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, et al. Supervised contrastive learning. arXiv:2004.11362. 2021.
15. Jain R, Semwal VB. A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors. IEEE Sens J. 2022;22(23):22943–51. doi:10.1109/jsen.2022.3213814.
16. Yu X, Park S, Kim D, Kim E, Kim J, Kim W, et al. A practical wearable fall detection system based on tiny convolutional neural networks. Biomed Signal Process Control. 2023;86:105325. doi:10.1016/j.bspc.2023.105325.
17. Syed AS, Sierra-Sosa D, Kumar A, Elmaghraby AS. Cross dataset non-binary fall detection using a ConvLSTM-attention network. In: 2022 IEEE Globecom Workshops (GC Wkshps); 2022 Dec 4–8; Rio de Janeiro, Brazil. p. 1068–73.
18. Yu X, Wan J, An G, Yin X, Xiong S. A novel semi-supervised model for pre-impact fall detection with limited fall data. Eng Appl Artif Intell. 2024;132:108469. doi:10.1016/j.engappai.2024.108469.
19. Gangadhar C, Roy PP, Kumar RD, Ramesh JVN, Ravikanth S, Akhila N. Wearable sensor-based fall detection for elderly care using ensemble machine learning techniques. MeasSens. 2025;39:101870.
20. Yi MK, Han K, Hwang SO. Fall detection of the elderly using denoising LSTM-based convolutional variant autoencoder. IEEE Sens J. 2024;24(11):18556–67. doi:10.1109/jsen.2024.3388478.
21. Mao WL, Wang CC, Chou PH, Liu KC, Tsao Y. MECKD: deep learning-based fall detection in multi-layer mobile edge computing with knowledge distillation. IEEE Sens J. 2024;24(24):42195–209. doi:10.1109/jsen.2024.3456577.
22. Yu X, Jang J, Xiong S. 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. 2021;13:692865. doi:10.3389/fnagi.2021.692865.
23. Sucerquia A, López JD, Vargas-Bonilla JF. SisFall: a fall and movement dataset. Sensors. 2017;17(1):198. doi:10.3390/s17010198.
24. Suh S, Rey VF, Lukowicz P. TASKED: transformer-based adversarial learning for human activity recognition using wearable sensors via Self-KnowledgE Distillation. arXiv:2209.09092. 2022.
25. Chen H, Gouin-Vallerand C, Bouchard K, Gaboury S, Couture M, Bier N, et al. Contrastive self-supervised learning for sensor-based human activity recognition: a review. IEEE Access. 2024;12:152511–31. doi:10.1109/access.2024.3480814.
26. Khaertdinov B, Asteriadis S. Temporal feature alignment in contrastive self-supervised learning for human activity recognition. arXiv:2210.03382. 2022.
27. Zhu Y, Zhuang F, Wang J, Ke G, Chen J, Bian J, et al. Deep subdomain adaptation network for image classification. IEEE Trans Neural Netw Learn Syst. 2021;32(4):1713–22. doi:10.1109/tnnls.2020.2988928.
28. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A kernel two-sample test. J Mach Learn Res. 2012;13(25):723–73.
29. Fawaz HI. Deep learning for time series classification. arXiv:2010.00567. 2020.
30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv:1512.03385. 2015.
31. Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, et al. Transformers in time series: a survey. arXiv:2202.07125. 2023.

32. Tanaka T, Nambu I, Maruyama Y, Wada Y. Sliding-window normalization to improve the performance of machine-learning models for real-time motion prediction using electromyography. Sensors. 2022 Jul;22(13):5005. doi:10.3390/s22135005.
33. Ragab M, Eldele E, Tan WL, Foo CS, Chen Z, Wu M, et al. ADATIME: a benchmarking suite for domain adaptation on time series data. ACM Trans Knowl Discov Data. 2023 May;17(8):1–18. doi:10.1145/3587937.
34. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, et al. Domain-adversarial training of neural networks. arXiv:1505.07818. 2016.
35. Long M, Cao Z, Wang J, Jordan MI. Conditional adversarial domain adaptation. arXiv:1705.10667. 2018.