Revisiting Fall Prediction Tasks: An Empirical Study of Transformer-GAN with Data Representation and Training Strategy *

Yu-Jung Hsu[1] and Yi-Chieh Wu[2][0000−0002−2973−5735]

[1] Dept. of Computer Science, National Chengchi University, Taipei, Taiwan
114753201@nccu.edu.tw
[2] Interdisciplinary Artificial Intelligence Center, National Chengchi University, Taipei, Taiwan matywu@gmail.com

Abstract. Fall detection and prediction play a critical role in elderly care, where timely response is essential for reducing injury severity and improving emergency outcomes. Skeleton-based approaches have shown promise due to their privacy-preserving nature; however, their performance often degrades under diverse camera configurations and non-standard viewpoints commonly encountered in real-world surveillance environments.

In this work, we revisit skeleton-based fall prediction from a data representation and training perspective, employing a transformer-based generative adversarial network (Transformer-GAN) . We design a unified pipeline that accommodates heterogeneous camera inputs with varying resolutions and viewpoints by normalizing skeleton coordinates into a resolution-independent space, enabling consistent skeleton representation. To improve model convergence and robustness under adjusted skeleton inputs, focal loss is incorporated into the Transformer-GAN training process.

Furthermore, we conduct an empirical evaluation of camera viewpoint correction for skeleton-based fall prediction. Experimental results demonstrate that viewpoint correction yields an average accuracy improvement of over 6.2% on the external ADL dataset and improves accuracy from 68% to 82% on the held-out Lecture Room and Office scenes. On our self-collected dataset, the system achieves an accuracy and F1-score exceeding 95%.

These results indicate that jointly reconsidering data representation, training strategy, and camera viewpoint effects can significantly enhance the robustness and generalization capability of skeleton-based fall prediction systems in practical elderly care scenarios.

Keywords: Fall Detection · Transformer-GAN · Image Viewpoint Correction · Edge AI

* This work was supported by the National Science and Technology Council, Taiwan (Grant No. NSTC 114-2222-E-004-001).

1 Introduction

Fall detection and prediction have long been critical issues in elderly care. As the global aging population continues to grow, the health risks and medical burdens caused by falls are becoming increasingly severe. Timely detection and rapid response are essential for reducing injury severity and improving emergency care outcomes after a fall. Advances in hardware architecture have enabled higher computational efficiency without sacrificing performance, allowing lightweight inference to be deployed on edge AI devices such as the NVIDIA Orin NX. Such on-device computation is well suited to elderly care scenarios, where real-time processing is crucial for immediate assistance and emergency response.

In our previous work [1], we formulated fall detection as a generative prediction problem, where the model learns to generate expected future pose trajectories, and fall events are identified through deviations between predicted and observed sequences. This formulation offers a key advantage over direct classification approaches: rather than learning normal motion patterns, the model is trained on fall sequences to learn the distribution of anomalous motion, such that a smaller deviation between the generated and observed sequences indicates a higher likelihood of a fall event, while a larger deviation suggests its absence. To implement this framework, we adopted a Conditional Generative Adversarial Network (CGAN) [2] rather than diffusion-based generative models. Unlike diffusion models, which introduce stochastic variation across samples, GANs produce deterministic outputs conditioned on the input, enabling stable and consistent generation of fall motion sequences. This deterministic property ensures that deviations caused by the absence of fall events remain reliable and consistently detectable through distance-based metrics. Both the generator and discriminator are built upon Transformer-based architectures [3], supporting effective sequence modeling within a compact model design suitable for lightweight deployment on edge devices. We employed the OpenPose model [4] to extract human joint keypoints for model training, achieving an average F1-score of 93% on the validation dataset. Consistent performance was also observed on external datasets.

The above architecture performs well in most surveillance scenarios. However, two key limitations remain. First, the model relies on absolute pixel coordinates as input, making it sensitive to image resolution and limiting its applicability to fixed camera configurations. Second, when images are captured under more extreme camera angles, a noticeable degradation in prediction accuracy is observed, which can be attributed to viewpoint-dependent variations in human pose representation, distortions in joint geometry caused by perspective effects, and inconsistencies between training and deployment camera configurations. These limitations motivate us to develop a more robust fall prediction pipeline that addresses resolution dependency through input normalization and mitigates the impact of diverse camera viewpoints through geometry-based skeleton adjustment, enabling the existing model to maintain stable performance across varying deployment conditions.

The main contributions of this work are summarized as follows:

- We design a pipeline that supports diverse camera inputs with varying resolutions and viewpoints by normalizing skeleton coordinates into a scale-independent space, enabling resolution-invariant skeleton representation without frame resizing.
- We empirically demonstrate that camera viewpoint correction improves skeleton-based fall prediction, yielding an average accuracy gain of over 6.2% on the external ADL dataset and improving accuracy from 68% to 82% on the held-out Lecture Room and Office scenes.
- We incorporate focal loss into the transformer-based GAN training process to improve convergence on viewpoint-corrected skeleton inputs, achieving a validation accuracy exceeding 95%.

The structure of this paper is as follows. Section 2 provides an overview of our previous work. Section 3 introduces the system architecture for this study, including camera viewpoint estimation, data preprocessing, and the proposed training strategy. Section 4 discusses the performance of our approach. Finally, Section 5 presents the conclusions of this paper.

2 Related Work

Many existing human pose estimation approaches can be applied to fall detection by extracting body keypoints and modeling their spatial relationships through Part Affinity Fields (PAFs) to construct skeleton representations [5, 6]. However, these models are generally designed for generic pose estimation rather than fall detection, and thus rely on additional pose analysis or heuristic post-processing to identify fall events. Moreover, most publicly available RGB-based pose datasets are collected from younger populations [7, 8], whose motion characteristics differ substantially from those of non-young-adult demographic groups such as elderly individuals and children. This demographic mismatch poses challenges for deploying pose-based fall detection systems in real-world elderly care scenarios and motivates further investigation into inferring fall-related motion patterns using existing datasets.

Recent studies have also explored the use of Generative Adversarial Networks (GANs) to reduce the cost and difficulty of motion data collection and annotation, primarily through data augmentation strategies [9]. In addition, GAN-based frameworks have been proposed to enhance the accuracy and robustness of human pose estimation [10]. Transformer-based architectures have also demonstrated strong capacity in capturing long-range temporal dependencies for sequential data modeling [3][11]. While these works demonstrate the potential of generative and attention-based models in motion analysis, most existing approaches focus on either spatial or temporal aspects in isolation. The joint modeling of visual, spatial, and temporal information for fall prediction remains relatively underexplored in the literature.

Although prior work, including our own [1], has demonstrated the potential of GAN-based temporal modeling for fall prediction, limitations in resolution

dependency and viewpoint sensitivity remain unaddressed. These gaps form the basis of the proposed approach presented in the following sections.

3 Methodologies

In this section, we introduce the methodology and experimental design of our transformer-based GAN framework to generate human skeleton trajectories. The core pipeline mostly follows our previous work [1], while newly introduced components and modifications are described in detail where applicable. After sequence generation, classification techniques are applied to identify fall events by evaluating multiple distance measures and selecting an optimal decision threshold.

3.1 Datasets

The dataset utilized in this study consists of 130 videos from an open-source collection [12], with annotations indicating the start and end frames of fall events. In nearly all videos, the subjects remain in a fallen state until the video terminates. To standardize the annotations, we preserved the original fall onset frames and redefined the fall end frames as the final frame of each video. The model was trained using videos from the Coffee Room and Home scenes, while the Lecture Room and Office scenes were held-out for evaluation.

For evaluation, three data sources were employed: (1) the held-out scenes from the training dataset, (2) an external dataset [13] in which each scenario, encompassing both ADL and fall events, was simultaneously captured by five web cameras, and (3) self-collected data recorded in daily-life environments. To ensure label consistency, we manually reviewed all fall scenes and identified the exact frame at which each fall begins and ends, except for the held-out scenes, which were already annotated.

3.2 Preprocessing

The preprocessing protocol includes skeleton instance selection and missing joint imputation, both applied during training and inference. When multiple instances appear, the skeleton with the maximum number of detected joints is selected. Missing joint coordinates are imputed using a Breadth-First Search (BFS) over the skeletal topology, propagating values from the nearest valid joints. This approach preserves biomechanical plausibility and spatial consistency, as shown in Algorithm 1. Only fall sequences are used during training for GAN-based sequence mapping.

The proposed model operates on 2D pose coordinates extracted from pose estimation results. As the model input requires only the 2D coordinates of 18 body keypoints, it is compatible with both standard and lightweight pose estimators, enabling the adoption of a lightweight OpenPose variant [14] during deployment on an NVIDIA Jetson Orin NX (8 GB) edge device for real-time inference under limited computational resources. To further enhance deployment flexibility, the coordinates are normalized into a scale-independent space using

$(x', y') = (x/W, y/H)$, where W and H denote the frame width and height, allowing the model to process videos with varying resolutions without frame resizing.

Algorithm 1 BFS-Based Missing Joint Coordinate Recovery Using Skeletal Topology

```
Require: skeleton data formulate as tree
Ensure: Complete skeleton data with filled missing joint values
Initialize missing_joints_list and find all missing values
for each missing_joint in missing_joints_list do
    visited ← new Set()
    found_value ← False
    current_layer ← [missing_joint]
    visited.add(missing_joint)
    while current_layer is not empty and found_value ≠ True do
        next_layer ← []
        for each node in current_layer do
            for each neighbor in node do
                if neighbor ∉ visited then
                    visited.add(neighbor)
                    if joint_value(neighbor) is not missing then
                        joint_value(missing_joint) ← joint_value(neighbor)
                        found_value ← True
                        break
                    end if
                    Add neighbor to next_layer
                end if
            end for
            if found_value = True then
                break
            end if
        end for
        current_layer ← next_layer
    end while
end for
return skeleton data
```

3.3 GAN Architecture

Fig. 1 illustrates the core approach, which consists of two main components: a generator and a discriminator, both built upon Transformer encoder architectures. An additional conditional vector serves as supplementary information for both components, guiding the generation process and ensuring structurally plausible results. By arranging consecutive temporal frames into a stacked spatial representation, each input block is processed as a unified unit, and the model generates the corresponding near-future skeleton representations as output.

Generator The generator takes pose sequences of 30 consecutive frames as input. To ensure that all fall frames can be utilized, each sequence starts 15 frames before the fall onset, such that the first 15 frames may contain non-fall motion while the remaining frames capture the fall event. As a result, fall sequences constitute the majority of the training data. A 15-frame overlap between adjacent sequences is introduced to maintain temporal continuity, such that each sequence contributes 15 new frames while sharing the remaining frames with the previous one.

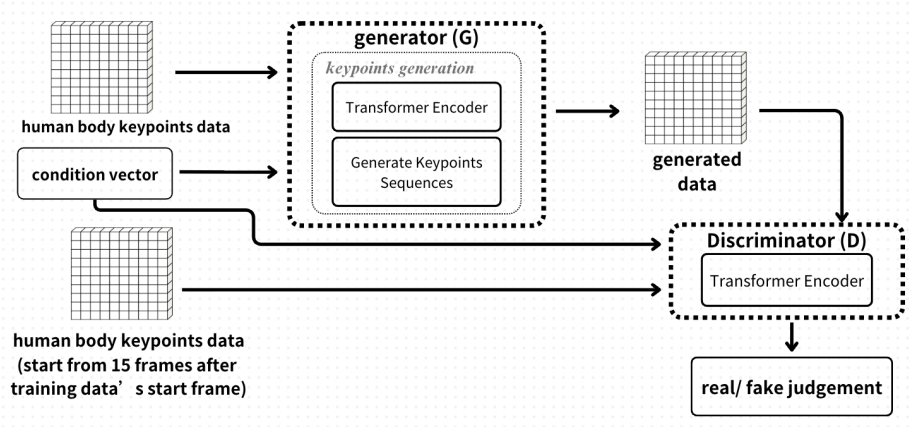


Fig. 1: Transformer-based GAN model for skeleton keypoint generation

Regarding the generator loss, it consists of four components: an adversarial loss $\mathcal{L}_{G_{\text{adv}}}$ that encourages the generator to produce skeleton sequences indistinguishable from real samples by the discriminator; a reconstruction loss $\mathcal{L}_{G_{\text{rec}}}$ that enforces spatial accuracy between generated and real skeleton sequences in the normalized ratio domain; a temporal difference loss $\mathcal{L}_{G_{\text{diff}}}$ that preserves frame-to-frame motion consistency by penalizing unrealistic temporal variations; and an auxiliary fall/non-fall classification loss $\mathcal{L}_{G_{\text{cls}}}$ that guides the generator to produce motion patterns consistent with the annotated motion category.

Since coordinate normalization results in numerically small reconstruction errors, focal weighting and onset-centered temporal weights are adopted to emphasize hard-to-fit samples and critical temporal regions during training. The total generator loss is defined as:

$$\mathcal{L}_G = \mathcal{L}_{G_{\text{adv}}} + \lambda_{\text{rec}}\mathcal{L}_{G_{\text{rec}}} + \lambda_{\text{diff}}\mathcal{L}_{G_{\text{diff}}} + \lambda_{\text{cls}}\mathcal{L}_{G_{\text{cls}}}, \tag{1}$$

$$\begin{aligned}
\mathcal{L}_{G_{adv}} &= \mathbb{E}_{x,c}\left[\text{BCEWithLogits}\left(D_{auth}(G(x,c),c),1\right)\right] \\
\mathcal{L}_{G_{rec}} &= \mathbb{E}_{x,c}\left[\text{FocalMSE}\left(G(x,c),x;w_t\right)\right] \\
\mathcal{L}_{G_{diff}} &= \mathbb{E}_{x,c}\left[\text{FocalMSE}\left(\Delta G(x,c),\Delta x;w_t\right)\right] \\
&\quad \text{where } \Delta x_t = x_t - x_{t-1} \\
\mathcal{L}_{G_{cls}} &= \mathbb{E}_{x,c,y}\left[\text{FocalBCE}\left(D_{cls}(G(x,c),c),y\right)\right]
\end{aligned}$$

In $\mathcal{L}_{G_{rec}}$ and $\mathcal{L}_{G_{diff}}$, a focal mean squared error is adopted to amplify large reconstruction errors in the normalized coordinate domain. Specifically, the focal MSE is formulated as $\text{FocalMSE}(e;w_t) = w_t\left(\frac{|e|}{\tau}\right)^{\gamma} e^2$, where e denotes the reconstruction error and w_t is the temporal weighting factor. The focusing parameter is set to $\gamma = 2.0$. To account for different error scales, we use $\tau = 0.03$ for the reconstruction loss $\mathcal{L}_{G_{rec}}$ and a smaller value $\tau = 0.02$ for the temporal difference loss $\mathcal{L}_{G_{diff}}$.

In the generator loss, λ_{rec}, λ_{diff}, and λ_{cls} are weighting coefficients that balance the contributions of the reconstruction, temporal difference, and auxiliary classification losses, respectively. Based on empirical evaluation, we set $\lambda_{rec} = 10$ and $\lambda_{diff} = 5$, which yield the best generation performance. The classification weight λ_{cls} is gradually increased during training and capped at a maximum value of 2 to stabilize adversarial optimization.

Discriminator The main objective of the discriminator is to distinguish real skeleton sequences from generated ones. Spectral normalization is applied to the linear projection layers of the discriminator to stabilize adversarial training [15]. The discriminator is optimized with a dual-objective loss: an adversarial term for sequence authenticity discrimination and an auxiliary fall-versus-non-fall classification term for learning motion-discriminative features:

$$\mathcal{L}_D = \mathcal{L}_{D_{\text{adv}}}^{\text{real}} + \mathcal{L}_{D_{\text{adv}}}^{\text{fake}} + \lambda_{\text{cls}} \mathcal{L}_{D_{\text{cls}}}. \tag{2}$$

$$\begin{aligned}
\mathcal{L}_{D_{\text{adv}}}^{\text{real}} &= \mathbb{E}_{x,c}\left[\text{BCEWithLogits}\left(D_{\text{auth}}(x,c),\, 1\right)\right] \\
\mathcal{L}_{D_{\text{adv}}}^{\text{fake}} &= \mathbb{E}_{x,c}\left[\text{BCEWithLogits}\left(D_{\text{auth}}(G(x,c),c),\, 0\right)\right] \\
\mathcal{L}_{D_{\text{cls}}} &= \mathbb{E}_{x,c,y}\left[\text{FocalBCE}\left(D_{\text{cls}}(x,c),\, y\right)\right]
\end{aligned}$$

The focal formulation alleviates class imbalance and emphasizes hard-to-classify transitional motions, such as recovery after falling.

The overall loss function combines the two optimization objectives above:

$$\mathcal{L}_{GAN} = \min_G \max_D \left[\mathcal{L}_D + \mathcal{L}_G\right] \tag{3}$$

3.4 Predict Pipeline

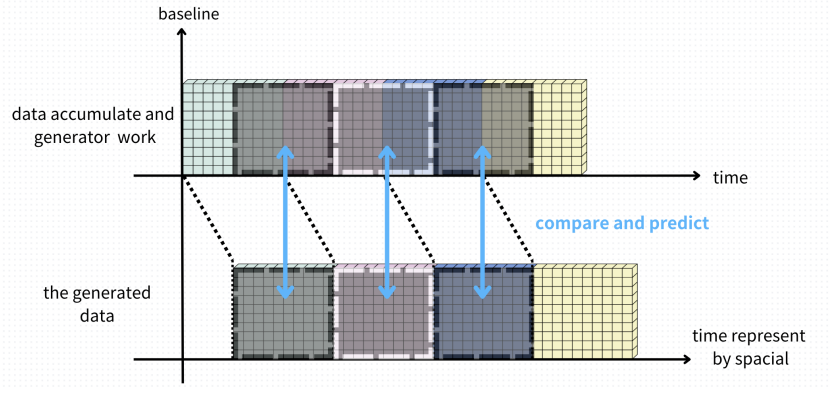


Fig. 2: Spatial and Temporal interaction using the GAN model.

Fig. 2 illustrates the analysis framework, showing how temporal and spatial dimensions are jointly modeled. The top section depicts the data processing pipeline where each generation cycle takes 30 input frames and produces a corresponding 30-frame generated sequence, with the dashed lines indicating the correspondence between input and generated data. Adjacent sequences overlap by 15 frames, ensuring temporal continuity across cycles. The lower section shows that while the full generated sequence is compared against the real data, the latter 15 frames of each cycle provide the new decision information for fall detection, as the first 15 frames are shared with the previous cycle. This enables concurrent generation and validation with just 15 new frames per cycle. As predictions are made before the complete observation window is available, temporal consistency and real-time inference are both achieved.

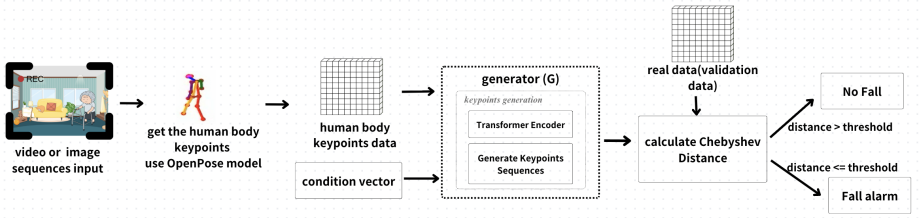


Fig. 3: Inference pipeline.

Fig. 3 illustrates the overall deployment pipeline. Given an input video stream, human skeleton keypoints are extracted using an OpenPose-based model and fed into the proposed model for motion analysis. For deployment on edge devices, the lightweight OpenPose variant [14] is adopted, which is directly compatible with the existing pipeline. The resulting motion representations are then evaluated using Chebyshev distance to classify each sequence as a fall or non-fall event.

For decision making, we evaluate lock-step distance measures using the Minkowski distance with different values of p (i.e., $p = 1, 2, \infty$) and cosine distance. Unlike our previous work [1], which required Dynamic Time Warping (DTW) to handle temporal misalignment, the proposed coordinate normalization into a scale-independent space mitigates scale- and magnitude-induced variations, rendering lock-step measures sufficient without DTW.

Based on our training samples, Bayesian decision theory is applied to estimate a threshold that achieves approximately equal classification risk. With refined annotations, fall events constitute only 22.02% of the dataset, leading to a highly imbalanced class distribution. Under this imbalance, assuming equal costs would bias the decision boundary toward non-fall events and increase the risk of missed fall detections. To address this issue, a cost-sensitive Bayesian decision framework is adopted, in which the class distribution informs a heavier penalty on false negatives than false positives, with $C_{\mathrm{FP}} : C_{\mathrm{FN}} = 1 : 3.5$, as

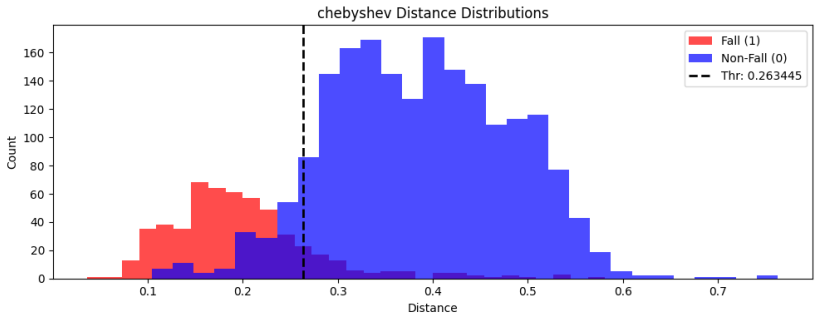


Fig. 4: Illustration of the class-wise data distribution and the decision boundary defined by the classification threshold.

defined by

$$R(t) = \pi_0 \, C_{\mathrm{FP}} \, \mathrm{FPR}(t) + \pi_1 \, C_{\mathrm{FN}} \, \mathrm{FNR}(t). \tag{4}$$

Under this setting, the final decision threshold is obtained by minimizing the expected Bayes risk estimated from the sample distribution. Based on empirical evaluation, Chebyshev distance ($p = \infty$) yields the most stable classification performance and is adopted as the primary decision metric.

3.5 Adaptive Skeleton Adjustment for Camera Viewpoint

To examine the impact of camera viewpoint on model performance, we leverage the external dataset [13], in which each scenario was captured simultaneously by five web cameras, enabling a systematic evaluation across diverse viewpoints.

Since such a multi-camera configuration is not present in the training dataset, we first examine the discrepancy between the camera poses of the external test cameras and those of the training dataset. For the training dataset, the Coffee Room environment is selected as the reference due to its more uniform background and larger sample count. Camera intrinsics and extrinsics are estimated from the first frame of each video within the Coffee Room folder using VGGT [16]. The estimates are then averaged to produce a representative camera configuration. For the external test dataset, camera parameters are estimated separately for each scenario using the corresponding multi-view recordings.

During the analysis, variations in camera intrinsics are found to contribute marginally to the observed performance differences. Instead, the relative rotation and translation encoded in the camera extrinsics are hypothesized to play a more significant role. The decomposition and analysis are conceptually inspired by the plane+parallax formulation [17, 18], which separates global homography-induced effects from residual parallax components. Based on the equivalence of relative pose, the effect of camera translation can be equivalently transferred to a displacement of objects in the scene. We therefore adopt a coordinate frame in which camera motion is parameterized by rotation only, with the translational component implicitly absorbed into the motion of subjects present in the footage, information that is already encoded in the observed motion dynamics of

the video data and thus requires no separate modeling. In contrast, camera rotation alters the viewing direction and consequently affects the perspective and occlusion structure of the scene, and cannot be compensated by object displacement alone. It must therefore be modeled explicitly. On this basis, we assume the primary difference between training and test cameras lies in rotation. The following analysis examines how this affects scene appearance and generative model behavior. Under this assumption, image alignment can be described by a rotation-induced homography

$$H_{\text{rot}} = K_2 R_{\text{rel}} K_1^{-1}, \tag{5}$$

where K_1 and K_2 denote the intrinsic matrices of the training and test cameras, respectively, and R_{rel} is the relative rotation between them. A parallax-based score is derived by aggregating projection discrepancies across sampled pixel locations and depth hypotheses, serving as an approximate indicator of viewpoint difference relative to the training configuration.

To prevent joints from falling outside the image boundaries, the transformation considers only the rotational component and ignores parallax effects. Moreover, a partial transfer strategy is adopted, in which each transformed joint position is defined as a linear interpolation between the original location and its rotation-aligned counterpart. The interpolation weight is adaptively determined based on the proportion of out-of-boundary joints after transformation, balancing viewpoint correction strength and skeleton integrity. A post-processing similarity adjustment is further applied to reposition any out-of-boundary joints within the visible image region.

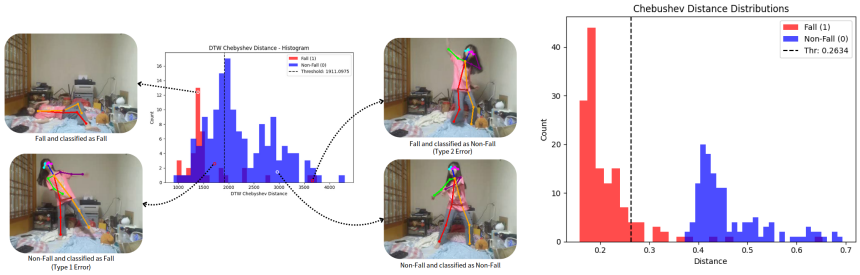
It should be noted that this viewpoint transformation strategy does not guarantee consistent performance improvements across all cross-camera configurations. Its effectiveness remains dependent on factors such as subject-camera distance and the quality of the original skeletal keypoints.

4 Results and Discussions

Table 1: Classification Results on Our Collected Fall Dataset

Class	**Precision**	**Recall**	**F1 Score**
Non-Fall	0.90	1.00	0.95
Fall	1.00	0.89	0.94
Accuracy		0.95	

To assess the effectiveness of the proposed viewpoint adjustment, we evaluate the model on the external ADL dataset across all five camera views. The dataset was captured using five fixed cameras, each placed at a different location within the same physical space. We compute a parallax score for each camera to



(a) (b)

Fig. 5: Comparison between (a) previous work[1] and (b) this study

quantify its deviation from the training configuration. The Chebyshev distance threshold of 0.2634, estimated from the training data, is applied for decision making across all evaluation sets. We first evaluate all recording sessions using the primary camera view annotated by the dataset, which indicates the most informative viewpoint for each recording session. Under this setting, the overall classification accuracy reaches 65.3%. We further analyze performance across individual camera views, as reported in Table 3, and observe substantial variation in classification accuracy among different camera IDs, indicating a strong dependence on camera viewpoint.

To investigate the underlying factors, we examine the relationship between camera placement and recognition performance using the geometry-based parallax score. Based on this analysis, the results for Camera 4, Camera 3, and Camera 2 exhibit a consistent trend in which lower parallax scores correspond to higher classification accuracy, suggesting that viewpoint similarity plays an important role in cross-camera generalization. However, Camera 1 and Camera 5 do not strictly follow this monotonic relationship, implying that factors beyond geometric alignment, such as occlusion, motion completeness, or capture quality, also affect fall recognition performance.

Overall, these results indicate that discrepancies in camera viewpoints between training and testing data have a tangible impact on skeleton-based fall recognition. We therefore apply the proposed viewpoint adjustment to transform test skeletons into the training camera configuration. As also shown in Table 3, cameras with larger initial parallax scores tend to exhibit lower baseline accuracy, while noticeable performance gains are observed for Camera 1 and Camera 5 after adjustment. This trend is further supported by the distance distributions in Fig. 7, where increased separation between class-wise distributions after adjustment leads to improved classification accuracy. Based on the numerical results and observed patterns, we empirically select a parallax score of 0.05 as a threshold to determine whether viewpoint adjustment is needed. Although the adjusted accuracy of Camera 2 slightly decreases, its distance distribution exhibits improved class-wise separation, suggesting enhanced discriminability; therefore, it is still considered in estimating the overall average accuracy gain of around 6.2%.

To further validate the effectiveness of the proposed viewpoint adjustment, we evaluate the model on the held-out Lecture Room and Office scenes. To control for class imbalance effects, the dataset is resampled to match the class distribution of the training data. As reported in Table 2, the model achieves an accuracy of 68% before viewpoint adjustment. After applying the proposed adjustment, the accuracy improves to 82%, with notable gains in both precision and recall for the fall class. The corresponding distance distributions before and after adjustment are illustrated in Fig. 6, showing increased class-wise separation consistent with the improved classification performance. These results further support the effectiveness of the proposed viewpoint adjustment across different deployment environments.

Finally, we evaluate the model on our self-collected dataset recorded in daily-life environments. As the subjects in this dataset are captured at close range, the parallax effect becomes significant, meaning that camera translation would induce large apparent displacements that cannot be reliably absorbed into the subject's motion representation. Viewpoint adjustment is therefore not applied to this dataset.

The resulting classification performance is reported in Table 1, achieving an accuracy and F1-score of approximately 95%. As illustrated in Fig. 5, compared to the previous work without coordinate normalization, the proposed method yields more clearly separated distance distributions between fall and non-fall classes, supporting the benefit of scale-independent coordinate normalization for improving classification discriminability.

To confirm the feasibility of real-time deployment on resource-constrained hardware, we evaluate the complete pipeline on an NVIDIA Jetson Orin NX (8 GB) edge device. The pose estimation module operates at an average rate of approximately 13.7 FPS, while the complete end-to-end pipeline, including pose extraction and decision logic, achieves an average throughput of around 13.4 FPS. When evaluated over a sliding window of 30 frames, the system maintains an average processing rate of approximately 13.3 FPS, indicating limited temporal jitter. The GAN-based generation module is triggered once every 15 new frames and introduces an average additional latency of approximately 14 ms per activation, without affecting the per-frame processing rate.

Limitations. The requirement for frame-level annotations does not align with prevailing datasets, which predominantly provide second-level labels, thereby limiting evaluation under a unified protocol. Furthermore, existing benchmarks generally lack camera pose annotations. In our current approach, camera pose is inferred from a single frame using VGGT; however, systematic comparison requires datasets with ground-truth camera pose information.

5 Conclusion

In this work, we revisit skeleton-based fall prediction by reformulating the problem as a generative prediction task, where a Transformer-based GAN is

Table 2: Classification Results on Lecture Room and Office Scenes Before and After Camera Viewpoint Adjustment

	Class	Precision	Recall	F1 Score
Before	Non-Fall	0.91	0.65	0.76
	Fall	0.39	0.78	0.52
	Accuracy		0.68	
After	Non-Fall	0.92	0.85	0.88
	Fall	0.57	0.72	0.64
	Accuracy		0.82	

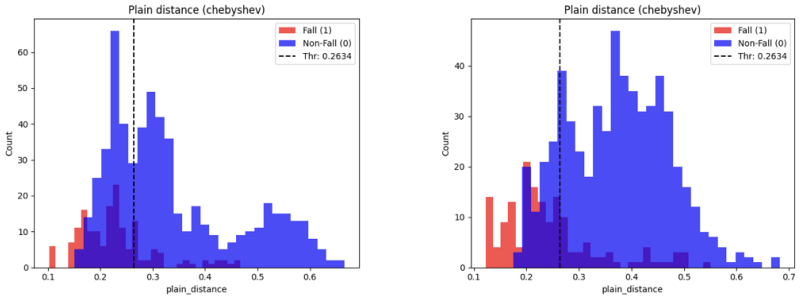


Fig. 6: Chebyshev distance distributions on Lecture Room and Office scenes before (left) and after (right) camera viewpoint adjustment.

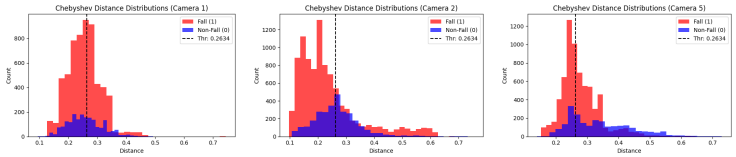
trained to generate future pose sequences, and fall events are identified through deviations between generated and observed motion. Close agreement between observed and generated motion indicates that the observed motion follows a fall-like trajectory, suggesting increased fall risk.

To enhance robustness under real-world conditions, several practical challenges are systematically addressed. Input resolution normalization automatically mitigates performance degradation caused by heterogeneous data sources. A cost-sensitive Bayesian decision strategy is further introduced to account for class imbalance during threshold estimation. Moreover, camera viewpoint differences are shown to have a substantial impact on model performance. By estimating camera intrinsics and extrinsics and applying geometry-based skeleton adjustment, performance improvements are achieved on both the external ADL dataset, with an average accuracy gain of over 6.2%, and the held-out Lecture Room and Office scenes, where accuracy improves from 68% to 82%. On our self-collected dataset, where close subject-camera distances limit the applicability of viewpoint adjustment, the model achieves an accuracy and F1-score of approximately 95% without adjustment.

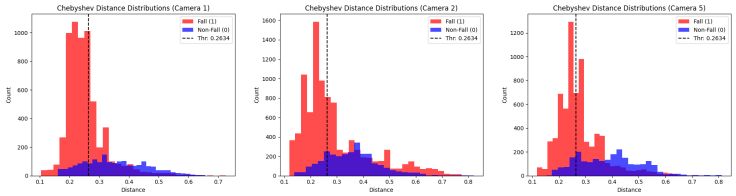
Finally, the entire pipeline is successfully deployed on an edge computing platform, demonstrating that the proposed approach imposes relatively low computational demands. The inclusion of a generative component does not introduce

Table 3: Classification accuracy before and after camera viewpoint adjustment, including baseline accuracy, parallax score, adjusted accuracy, and accuracy gain.

Camera ID	Baseline Acc.	Parallax Score	Adjusted Acc.	Δ Acc.
1	0.529	0.051*	0.709	+0.180
2	0.679	0.050*	0.630	−0.049
3	0.721	0.038	0.739	+0.018
4	0.744	0.020	0.678	−0.066
5	0.561	0.058*	0.616	+0.055



(a) Camera 1 (Before) (b) Camera 2 (Before) (c) Camera 5 (Before)



(d) Camera 1 (After) (e) Camera 2 (After) (f) Camera 5 (After)

Fig. 7: Distance distributions of Camera 1, Camera 2, and Camera 5 in the fall subset of ADL[13].

significant inference latency, making the framework well suited for deployment in resource-constrained environments. Compared to heavier detection-based methods that require substantial computational resources, the proposed lightweight design achieves competitive performance while maintaining efficiency, highlighting its practical potential for real-world fall risk assessment.

In future work, based on our fall prediction findings, extensive real-world testing will be crucial to validate the method's effectiveness and ensure its reliability in practical applications.

References

1. Yu-Jung Hsu and Yi-Chieh Wu. From temporal to spatial: A transformer-gan approach for fall prediction. In *2025 IEEE International Conference on Advanced Visual and Signal-Based Systems (AVSS)*, pages 1–4. IEEE, 2025.
2. Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

4. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:172–186, 2018.
5. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:172–186, 2018.
6. Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 23:13498–13511, 2021.
7. Edouard Auvinet, Franck Multon, Alain Saint-Arnaud, Jacqueline Rousseau, and Jean Meunier. Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Transactions on Information Technology in Biomedicine*, 15(2):290–300, 2011.
8. Bogdan Kwolek and Michal Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014.
9. Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, Wenjun Zeng, and Wei Wu. Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation. pages 2228–2238, 10 2023.
10. Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11855–11864, 2021.
11. Jeonghyeok Do and Munchurl Kim. Skateformer: skeletal-temporal transformer for human action recognition. In *European Conference on Computer Vision*, pages 401–420. Springer, 2025.
12. Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Optimized spatio-temporal descriptors for real-time fall detection: Comparison of support vector machine and adaboost-based classification. *Journal of Electronic Imaging*, 22:041106–041106, 10 2013.
13. Greet Baldewijns, Glen Debard, Gert Mertes, Bart Vanrumste, and Tom Croonenborghs. Bridging the gap between real-life data and simulated data by providing a highly realistic fall dataset for evaluating camera-based fall detection algorithms. *Healthcare Technology Letters*, 3:6–11, 2016.
14. Daniil Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. In *International Conference on Pattern Recognition Applications and Methods*, 2018.
15. Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ArXiv*, abs/1802.05957, 2018.
16. Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
17. Michal Irani, Prabu Anandan, and Meir Cohen. Direct recovery of planar-parallax from multiple frames. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:1528– 1534, 12 2002.
18. Michal Irani and P. Anandan. A Unified Approach to Moving Object Detection in 2D and 3D Scenes. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(06):577–589, June 1998.