# Skeleton-based Fall Detection using Spatial Temporal Graph Convolutional Networks with Learnable Edges

*Alex Liang*

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-115
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-115.html

May 16, 2024

Acknowledgement

I would like to thank Professor Brian A. Barsky for his guidance and support throughout this project. I am also grateful to undergraduate researchers Allen Cao, Dhruv Kumar, and Ishaan Ghose for their valuable suggestions. Dhruv's idea of visual inference using Large Language Models helps me quickly find the direction in the early stages of this project. The pose estimation and action recognition experiments conducted by Ishaan Ghose and Allen Cao provided valuable insights that guided the selection of AlphaPose and STGCN as crucial baselines for this project. Lastly, I would like to acknowledge other students in the Barsky Lab at UC Berkeley, who shared their time and insights on the project during group meetings.

# Skeleton-based Fall Detection using Spatial Temporal Graph Convolutional Networks with Learnable Edges

Jinxuan Liang

## Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science**, **Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Brian A. Barsky

Professor Brian A. Barsky

Research Advisor

15 May 2024

(Date)

* * * * * * *

Professor Avideh Zakhor

Second Reader

5/16/2024

(Date)

# Acknowledgements

I would like to thank Professor Brian A. Barsky for his guidance and support throughout this project. I am also grateful to undergraduate researchers Allen Cao, Dhruv Kumar, and Ishaan Ghose for their valuable suggestions. Dhruv's idea of visual inference using Large Language Models helps me quickly find the direction in the early stages of this project. The pose estimation and action recognition experiments conducted by Ishaan Ghose and Allen Cao provided valuable insights that guided the selection of AlphaPose and STGCN as crucial baselines for this project. Lastly, I would like to acknowledge other students in the Barsky Lab at UC Berkeley, who shared their time and insights on the project during group meetings.

# Abstract

In response to the growing demographic of older individuals living alone and the heightened risks of falls they face, real-time fall detection systems using surveillance videos have emerged as crucial tools for ensuring prompt assistance. This report introduces a novel real-time fall detection method that integrates learnable edges into Spatial Temporal Graph Convolutional Networks (STGCN) for enhanced accuracy in classifying human actions. Leveraging short sub-sequences of skeleton data as inputs, the proposed model achieves rapid training and inference while demonstrating robust generalization across diverse environmental conditions. The proposed method underscores its efficacy in real-world fall detection tasks. Evaluation through a devised scheme, simulating real-time video streams, validates the model's effectiveness, quantified through metrics such as accuracy, specificity, and sensitivity.

# Contents

# Chapter 1

# Introduction

In today's society, the demographic shift toward an older population has been driven by advancements in healthcare and declining birth rates. Ensuring the well-being of elderly individuals has become a pressing concern for society. Falls are one of the primary health threats facing elderly individuals. According to the Centers for Disease Control and Prevention, 1 in 4 adults ages 65 years and older report falling each year [1]. When an elderly person falls, they must receive immediate medical attention to assess for serious injuries such as brain damage. If left unaddressed, these falls can prove fatal. However, research from the Pew Research Center indicates that approximately 27% of American adults aged 60 and older live alone [2]. This isolation poses significant risks for seniors. Living alone can make it challenging for elderly individuals to access immediate assistance in emergencies.

One existing solution to this problem is wearable senior alert devices such as smartwatches and necklaces [3]. However, these devices require constant battery recharging and cannot provide video feedback on the incidents. More importantly, the alert system would completely fail if the elderly individual forgets to wear the device.

Hence, a video-based alert system using surveillance videos for fall detection and sending out timely alerts and feedback can play a more crucial role in reducing the injury of elderly people and its subsequent consequences due to falls. Using the video stream from a simple surveillance camera installed at home, we can detect fall behaviors of the seniors, and notify their family members and medical services to ensure that the individual receives prompt medical assistance. Once installed, such surveillance cameras can provide 24/7 monitoring of individuals without requiring frequent maintenance.

A key challenge of video-based systems is their lack of direct access to individual body metrics. They must infer physical behaviors solely from RGB data provided by the surveillance camera. Therefore, there is a pressing need for an accurate and practically applicable fall detection algorithm.

In this report, a real-time skeleton-based fall detection method is proposed, which integrates learnable edges into Spatial Temporal Graph Convolutional Networks (STGCN) [4] [5] for improved accuracy in classifying human actions. Our method estimates human body keypoints for each frame of the RGB video input and constructs sub-sequences of the hu-

man skeleton. By utilizing short sub-sequences of skeleton data as inputs, fast training, and inference can be achieved. The model demonstrates robust generalization capabilities across various settings, including different daily activities, camera angles, and lighting conditions not present in the training data. Notably, it outperforms conventional STGCN architectures lacking learnable edges. To accurately assess the algorithm's performance under conditions closely resembling real-world scenarios, an evaluation scheme is devised. This scheme runs the algorithm on test videos emulating real-time streaming and quantifies its effectiveness through measures such as accuracy, specificity, and sensitivity. These metrics provide more direct insight into the applicability of the proposed method compared to the schemes used in related works. The code, skeleton data, and models of STGCN-LE are publicly available[1].

[1] https://github.com/degaliang/auto-senior-care-system

# Chapter 2

# Related work

The pose estimation that is employed as a pre-processing step for the skeleton-based approach is presented in this section. We examine the pose estimation method, as well as prior research endeavors in the realm of skeleton-based fall detection.

## 2.1 Human Pose Estimation

Human Pose Estimation entails predicting the 3D or 2D positions of human body joints and their corresponding skeletal structure from a photograph. Typically, this skeletal structure is represented as a graph, with joints serving as vertices connected by edges. This geometric abstraction of the human body facilitates understanding of human actions. In the fall detection methodology proposed in this work, AlphaPose is used for pose estimation to extract skeleton data as inputs for the fall detection model [6] [7] [8]. According to the pose estimation experiments conducted by undergraduate researchers in our group, **Allen Cao** and **Ishaan Ghose**, the other state-of-the-art pose estimation toolbox, OpenPose [9]–[12], is more vulnerable to lighting conditions and occlusions, and tends to produce more incorrect estimation. AlphaPose leverages Convolutional Neural Networks as its primary deep learning architecture for pose estimation, alongside techniques such as Symmetric Integral Keypoint Regression (SIKR) and Parametric Pose Non-Maximum-Suppression (P-NMS) to enhance both speed and accuracy. Notably, AlphaPose achieves accurate pose estimation in real-time, a critical requirement for a real-time fall detection system reliant on skeleton data inputs. However, it may encounter challenges in scenes with specific lighting conditions or low-resolution image data, potentially impacting downstream tasks like action recognition, which is further discussed in Section 5.3.

## 2.2 Skeleton-based Fall Detection

The skeleton-based fall detection methods utilize skeleton data extracted from video streams for classification. Several skeleton-based detection techniques have been proposed in the lit-

erature. Yan [13] is one previous attempt at using Spatial-Temporal Graph Convolutional Networks (STGCN) for fall detection. It uses the motion data of five inertial sensors for building the spatial-temporal graph that STGCN operates on. The model architecture follows the one proposed in Yan [4].

In comparison, this report proposes to apply STGCN to skeleton data extracted from RGB data. Learnable edge weights are added to the original STGCN for enhanced performance. Chen [14] proposed to detect falls using decision conditions defined on second-order features computed from skeleton data such as the speed of joints, the angle between the center-line of the human body and the ground, and the width-to-height ratio of the human bounding box. Some deep-learning-based methods like [15] use neural networks, LSTM in this case, to classify human skeleton sequences. Both [16] [17] used the Support Vector Machine (SVM) to predict using features extracted from the human skeleton.

While these papers all reported a high accuracy on fall detection tasks, they did not explicitly evaluate the method using a well-defined scheme to assess the system performance in a practical scenario. In addition, previous work tends to use slightly different metrics for evaluating the performance of their methods, which makes it difficult to compare the existing skeleton-based fall detection method. In Chen [14], the author categorizes human actions as falling actions, similar falling actions (squat), and daily actions (walking), and evaluates the algorithm based on the classification results of these categories. While classifying each action independently, this ignores the case that some of these actions can happen in sequence within a short period. Given a video clip, an algorithm should be able to differentiate falls from all other actions present in the video. The proposed evaluation scheme does not account for this. Although Chen [16] proposes to evaluate using full videos of human action, the author does not specify the metric used to determine the classification correctness of each video. In the work of Jeong [15], the author only reports the accuracy of the algorithm and does not include true-positive and false-positive rates. These two or equivalent measurements are important for assessing an alert system as an ideal algorithm should obtain high accuracy with the least amount of false alerts.

To address these shortcomings, this paper not only introduces a new skeleton-based fall detection method but also provides a generic evaluation scheme that can be used to assess the performance of fall detection systems.

Additionally, the above-mentioned methods use different datasets for training and testing. Since this work is based on the Le2i Fall Detection Dataset, the results are compared with methods that utilize the same dataset [18][19][20]. They employ various approaches, such as Dual-Channel Feature, Body Geometry, and Kinematic Theory. The comparison of performance results with methods utilizing the same dataset underscores the effectiveness and robustness of the proposed approach.

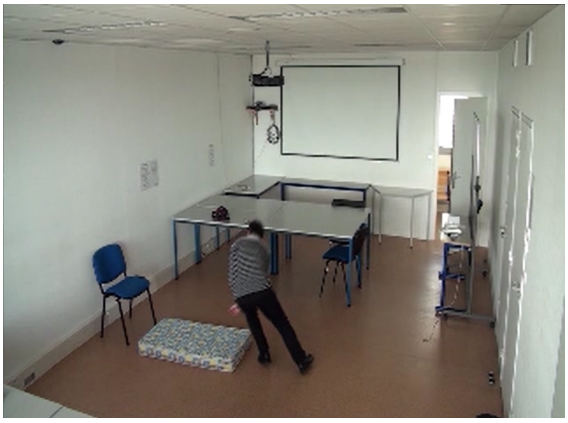
# Chapter 3

# Dataset

While many fall detection datasets have been proposed and used in previous work, this work uses the Le2i (ImVia) Fall Detection Dataset [21]. The dataset contains 191 realistic videos of both falls and activities of daily living captured by a single camera. The frame rate is 25 FPS and the resolution is 320×240 pixels. The videos are recorded in four different locations: home, coffee room, office, and lecture room (Figure 3.1). More importantly, the authors also provide starting and end frames of the fall events for the videos captured in the coffee room and at home. We create sub-sequences of "Fall" and "Non-Fall" actions according to the annotations and use the AlphaPose toolbox to estimate the human body keypoints on each frame of the videos in COCO format, which contains 17 joints. For each joint, AlphaPose produces 2D coordinates $(X, Y)$ in pixel space and a confidence score $C$. The confidence score is dropped in this case. In addition, the toolbox also outputs the frame index corresponding to each set of joints. Frames where AlphaPose does not detect joints will not be reported.

(a) Coffee Room

(b) Home



(c) Lecture Room

(d) Office

Figure 3.1: Sample images from the Le2i dataset

# Chapter 4

# Method

## 4.1 Skeleton Sub-sequence Training

In the proposed method, the STGCN-LE model takes a sequence of human skeleton data as the input. The input sequence is constructed by sampling $L_{in}$ consecutive frames from a surveillance video or a real-time I/O stream, as shown in Figure 4.1. During training, each input sequence is labeled as "Fall" or "Non-Fall" according to the annotations in the Le2i dataset. The $L_{in}$ is picked to be 45 because no falling sequences exceed 45 frames in the dataset. At test time, the input sequence is sampled as the current frame plus 44 consecutive frames in the past. This sub-sampling strategy ensures that the model only sees the least number of frames required to classify a skeleton sequence, which reduces training and inference time significantly. It effectively excludes random body movements that precede or follow the fall behavior so that the model will not be misled by these noises.

## 4.2 Batch-balanced Sampling

The Le2i dataset contains an equal number of "Fall" and "Non-Fall" videos. When we sample skeleton sub-sequence from each video, we will get exactly 1 "Fall" sub-sequence and many "Non-Fall" sub-sequences, which leads to an unbalanced class distribution. To mitigate this, we apply batch-balanced sampling at training time. That is, we maintain a 1:1 class ratio in the batch sampler. This measure is essential for stabilizing the training process and minimizing the risk of overfitting.

## 4.3 STGCN with Learnable Edges (STGCN-LE)

Spatial Temporal Graph Convolutional Networks (STGCN), introduced in Yan [4], is an extension of the Graph Convolutional Networks (GCN) architecture. It enhances traditional graph convolutions in the spatial dimension of GCN by incorporating temporal dimensions.

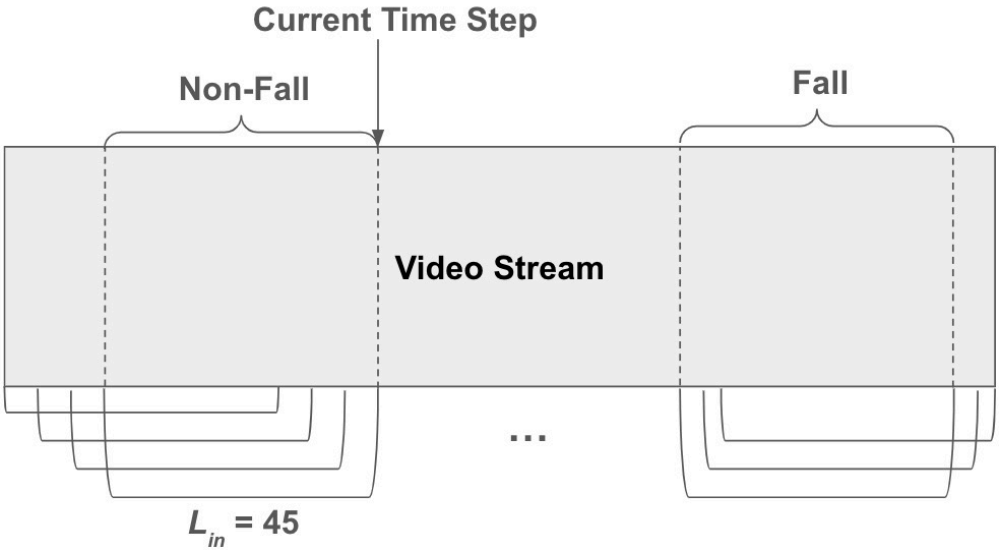



Figure 4.1: Skeleton sub-sequence sampling for training.

The STGCN operates on a spatial-temporal graph constructed from a sequence of hierarchical data that can be represented as a graph (see Figure 4.2). In this work, the spatial-temporal graph is constructed as an undirected graph $G = (V, E)$ with $N$ human body joints per frame and a total of $T$ frames. The number of frames is consistent with the sampling window size mentioned in **Section 4.1**, so $T = L_{in} = 45$. In the graph, joints from the same frame are connected by intra-body edges (blue edges in Figure 4.2), and each joint is also connected with its counterpart in different frames by inter-frame edges (green edges in Figure 4.2). Formally, we define nodes to be $V = \{v_{ti} | t = 1, ..., T, i = 1, ..., N\}$ and edges to be $E = \{E_S, E_T\}, E_S = \{(v_{ti}, v_{tj}) \mid i, j \in C, \forall t \in \{1, 2, \ldots, T\}\}, E_T = \{(v_{ti}, v_{(t+1)i}) \mid \forall t \in \{1, 2, \ldots, T-1\}\}$, where $C$ is the natural joints connections as defined in COCO-POSE [22]. The edge connections are represented using a standard adjacency matrix $A$ of size $N \times N$. Each node is represented by a $C$-dimensional feature vector, where $C$ is the number of channels. Hence, we have a feature mapping $f : V \rightarrow R^C$. $C$ will be 2 in the first layer, which corresponds to the 2D coordinates of each joint. In each layer in the STGCN, one convolution operation is a standard graph convolution in the spatial dimension of the spatial-temporal graph, followed by a convolution in the temporal dimension. For each node in the graph, a spatial convolution that transforms the input feature $f_{in}$ of $i$th node in $t$th frame to the output feature $f_{out}$ is defined as:

$$\mathbf{f_{out}}(v_{ti}) = \mathbf{w}(v_{ti}) \odot \sum_{v_{tj} \in B(v_{ti})} \mathbf{f_{in}}(v_{tj}) \tag{4.1}$$

where $B(v_{ti})$ represents the set of neighbors of $v_{ti}$ and $\odot$ is the element-wise multiplication. The weight function $\mathbf{w}(\cdot)$ returns the weight vector, which is multiplied by the feature vector

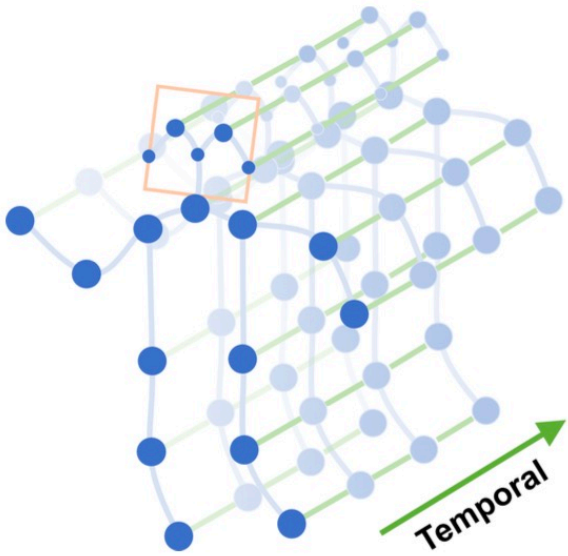



Figure 4.2: The spatial-temporal graph of a skeleton sequence where the STGCN-LE operates on [4]. Each blue node represents a human body joint, which is connected by edges to form the skeleton. Each joint also has temporal connections with its counterparts in the previous and the following frames.

element-wise. The summation can be replaced by other permutation equivariant functions like MEAN or MAX.

As shown in Figure 4.3, the skeleton structure produced by AlphaPose follows the geometric structure of the human body. If we use this structure to construct the spatial-temporal graph, it will construct the graph to have the same structure as the human body and assign all edges with weight 1 [4]. However, this limits the expressive power and the learning ability of the STGCN model. When aggregating information from a neighborhood, it is important to recognize that each vertex may hold varying degrees of significance in detecting an action. The lower-body movements might be more important than the movements of the neck and head. Moreover, some latent relationships may exist between unconnected vertices in the graph. For example, it may be important for "wrist" and "ankle" to share information directly with each other. However, this is not possible as they are not connected in the graph constructed based on the human body geometric. To address this, instead of using a pre-determined adjacency matrix, we make it learnable. We initialize $A$ to be the graph structure output by AlphaPose and let the network gradually learn to adjust the importance of each existing edge. We also allow the model to add new edges to the spatial-temporal graph as needed. This encourages the model to discover hidden relationships between joints that are not directly connected in the original graph. The creation of new edges also in-

Figure 4.3: Human body skeleton produced by AlphaPose.

creases the speed of information flow. For example, in the original graph, it would take at least 7 message passing operations for the information at "wrist" to reach "ankle" because the distance between the two vertices is 7 in the graph. By adding a direct edge with an appropriate weight, the information can be shared in one pass. The learnable edge connections are achieved by adding a weight mask and a bias mask that are of the same shape as $A$ in each STGCN layer. The weight mask, denoted as $\mathbf{M_w^l}$, is initialized to have the same value

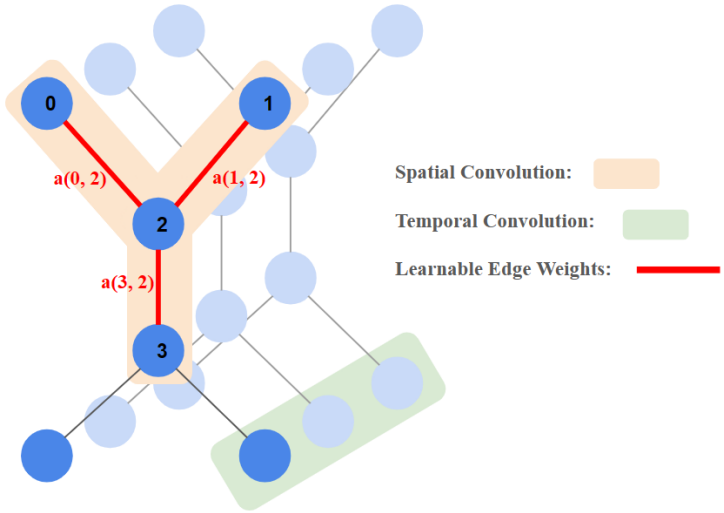



Figure 4.4: The illustration of a forward pass in STGCN-LE. The spatial convolution aggregates the information of neighbors vertex 0, 1, and 3 for 2, using learnable edge weights. The temporal convolution aggregates information of 3 adjacent frames, which corresponds to $K_T = 3$.

as $A$, where $l$ the layer index. The bias mask, denoted as $\mathbf{M_b^l}$, is initialized to all zeros. The transformed adjacency matrix of $l$th layer $\mathbf{A_m^l}$ is computed as:

$$\mathbf{A_m^l} = \mathbf{A} \odot \mathbf{M_w^l} + \mathbf{M_b^l} \tag{4.2}$$

Thus, the spatial convolution is redefined as:

$$\mathbf{f_{out}}(v_{ti}) = \mathbf{w}(v_{ti}) \odot \sum_{v_{tj} \in B(v_{ti})} a_m^l(v_{ti}, v_{tj}) \cdot \mathbf{f_{in}}(v_{tj}) \tag{4.3}$$

where $a_m^l(v_{ti}, v_{tj}) = \mathbf{A_m^l}[v_{ti}][v_{tj}]$ is the learned edge weight between $v_{ti}$ and $v_{tj}$ in $l$th layer.

In each STGCN-LE layer, a forward pass consists of a spatial convolution (Equation 4.3) followed by a temporal convolution. The temporal convolution is implemented as a 1D convolution in the temporal dimension with a temporal kernel of width $K_T$. One can choose to compress the temporal dimension by adjusting the stride. An illustration of such operations is shown in Figure 4.4.

# Chapter 5

# Evaluation

In previous work on fall detection, researchers typically use accuracy, precision, sensitivity, and specificity of the raw prediction of the model as the major evaluation metrics [23] [24] [16] [17]. Although these metrics provide a good insight into the performance of the classification model, they do not effectively assess the fall detection method as a whole. This is because these metrics ignore the dependencies between different input samples of the model. For example, suppose there is a fall-non-fall pair drawn from the same video, the classification model predicts both of them to be "Fall." Using the above-mentioned metrics, these would be considered to be one True Positive and one False Positive. However, from the perspective of the whole video stream, the result is ambiguous as to whether we should consider this specific video stream to be correctly classified. In a practical sense, a fall detection system that produces many false alarms is not desirable. Hence, we need to define a new evaluation scheme to better assess the performance of a fall detection system.

## 5.1 Real-time Fall Detection Simulation

This work proposes a real-time fall detection simulation test as a more suitable metric for evaluating fall detection methods. Rather than basing evaluations solely on model predictions for input sequences, this approach utilizes actual video streams as test data. By aggregating the classification model's predictions across these continuous streams, we can produce a more accurate and realistic assessment of the fall detection system's effectiveness. Ideally, the test data should be sourced from a distinct environment compared to the training and validation datasets. Specifically, the test videos should be captured in settings featuring diverse camera angles and lighting conditions. In this case, the STGCN-LE is trained on the data taken from the locations "Coffee room" and "Home" and tested on videos taken in "Lecture room" and "Office." A test video is classified as:

- True positives (TP): if all "Fall" sub-sequences drawn from it are detected as "Fall" and all "Non-Fall" sub-sequences are detected as "Non-Fall".

| **Model Architecture and Hyperparameters** | | **Real-time Fall Detection Simulation** | | | **Training-time Accuracy** | |
|---|---|---|---|---|---|---|
| Learnable Edges | Batch-balanced Sampling | Accuracy | Specificity | Sensitivity | Training | Validation |
| × | × | 0.684+/-0.118 | 0.746+/-0.161 | 0.632+/-0.326 | 0.959+/-0.027 | 0.957+/-0.030 |
| × | ✓ | 0.804+/-0.051 | 0.692+/-0.114 | 0.897+/-0.077 | 0.971+/-0.012 | **0.971+/-0.007** |
| ✓ | × | 0.765+/-0.156 | **0.854+/-0.082** | 0.690+/-0.348 | 0.959+/-0.028 | 0.962+/-0.028 |
| ✓ | ✓ | **0.828+/-0.028** | 0.738+/-0.051 | **0.903+/-0.020** | **0.986+/-0.002** | 0.963+/-0.014 |

Figure 5.1: The comparison of models with different architecture and hyperparameters.

- False positives (FP): if the video does not contain "Fall" and at least one sub-sequence drawn from it is detected as "Fall".
- True negatives (TN): if the video does not contain "Fall" and all sub-sequences drawn from it are detected as "Non-Fall".
- False negatives (FN): if the video contains "Fall" and none of the sub-sequences drawn from it are detected as "Fall".

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.2}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5.3}$$

The sub-sequences of a video are drawn using a sliding window of size $L_{in}$ with some stride. We use stride 1 to evaluate our model. Ideally, this stride should match the stride used at the production stage. Accordingly, the stride should be set to $m$, if the fall detection system samples a sub-sequence from the real-time video stream every $m$ frames when deployed.

## 5.2 Results

To evaluate the proposed architecture, I used 5-fold cross-validation with "home" and "coffee room" subsets in the Le2i dataset as training data. For each fold, the model is evaluated with the real-time fall detection simulation test proposed in Section 5.1. The "office" and "lecture room" subsets from the dataset are used for testing. To evaluate the efficacy of our proposed

| Method | Accuracy | Specificity | Sensitivity | Precision | Recall |
|---|---|---|---|---|---|
| **STGCN-LE** | 96.43% | 100.00% | 95.24% | 100.00% | 95.24% |
| Dual-Channel Feature | 96.91% | 96.51% | 97.37% | 97.65% | - |
| Body Geometry | 84.60% | - | - | 90.00% | 90.00% |
| Kinematic Theory | 98.00% | 98.30% | - | 97.00% | 97.20% |

Figure 5.2: Comparison between evaluation results of STGCN-LE with existing methods on the Le2i Dataset. Dual-Channel Feature [18], Body Geometry [19], Kinematic Theory[20].

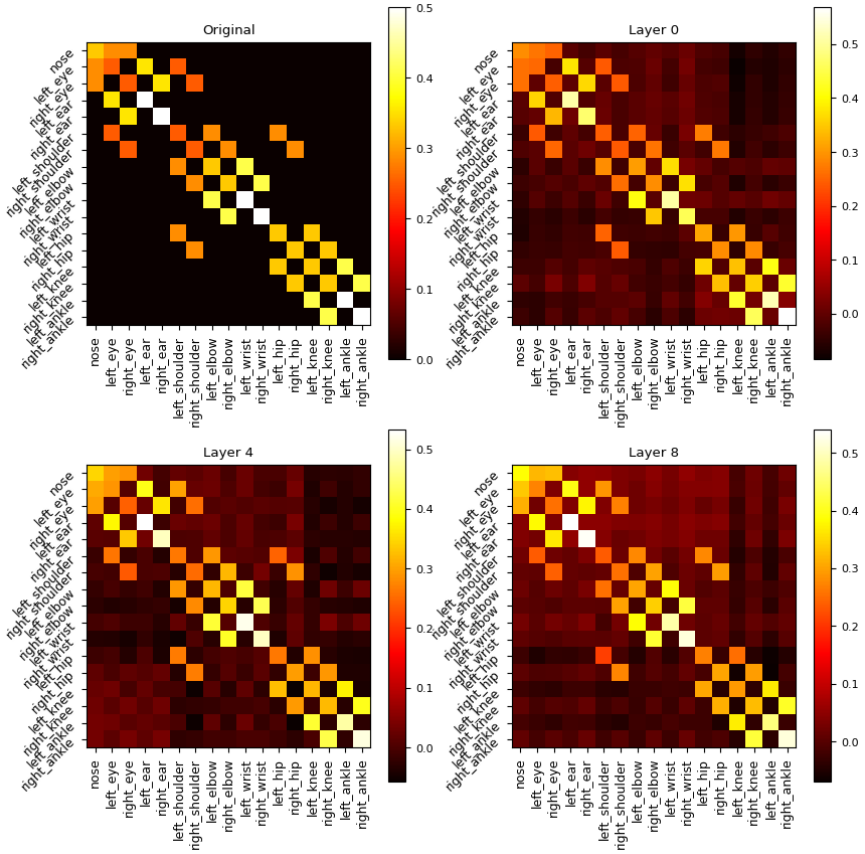



Figure 5.3: Adjacency matrices learned in the early, middle, and later layers of STGCN.

method, we conducted experiments by disabling the learnable edges and employing non-batch-balanced sampling. Subsequently, the performance of these configurations with our proposed architecture was compared. The assessment was based on cross-validation results, as depicted in Figure 5.1. All models were trained with Adam optimizer and a learning rate of 1e-3 for 35 epochs.

According to the ablation study, the proposed architecture can generalize better to unseen testing data. Despite exhibiting high training-time accuracy, the vanilla STGCN, lacking learnable edges and employing non-batch-balanced sampling, only achieved a test accuracy of 68.4%. By applying batch-balanced sampling, the test accuracy is boosted by 12% as it effectively regularizes the training so that the model does not overfit the data. On the other hand, the learnable edges encourage the model to discover hidden relationships among joints in the latent space and prevent it from only memorizing the pattern in the training data. In general, the proposed method not only leads to better generalization but also reduces the model variance. In Figure 5.3, we show the adjacency matrix $\mathbf{A_m^l}$ learned by different layers in STGCN. The model demonstrates an ability to incorporate additional edges with positive or negative weights while adhering to the pre-defined graph structure rooted in human body geometry.

Figure 5.2 shows the performance results of three other fall detection approaches on the Le2i Fall Detection Dataset. The STGCN-LE method proposed in this work is trained and tested using 80% and 20% of the Le2i dataset, respectively. Since the train-test split of the three approaches is not explicitly reported, the statistics shown in Figure 5.2 may differ from the true performance results.

## 5.3 Weaknesses

While the proposed method shows excellent performance on both training and validation data (test fall, test sit, validation fall), it is vulnerable to certain human actions in daily living activities. The primary reason for this limitation is the insufficient representation of human actions in the Le2i dataset, particularly those that occur commonly in daily life. Consequently, the model learns to differentiate only between falls and other human actions present in the training data. While the model demonstrates proficiency in distinguishing certain actions, such as sitting down or lifting an object, from falls, it may misclassify activities involving substantial body movements, such as squatting down to tie shoelaces (Figure 5.5).

To address this problem, one can scale up training data and train the model on a multi-class classification task instead of the binary classification employed in this paper. In theory, this approach can compel the model not only to learn the characteristics of fall behavior but also to discern other human actions and differentiate them from falls.

In addition, as mentioned in Section 2.1, the accuracy of skeleton-based action recognition methods heavily relies on the accuracy of the upstream task of pose estimation. The performance of pose estimation tools like AlphaPose can be influenced by various factors, in-

cluding lighting conditions, video resolution, and the scale of the backbone model. In Figure 5.5, we illustrate a failure case of AlphaPose on the Le2i dataset. In this instance, AlphaPose incorrectly estimates poses for shadows on the wall. Constructing a spatial-temporal graph based on such erroneous poses can lead to confusion in the fall detection model. Experimental results showed that including such samples in the training data may detrimentally affect the model's performance. One potential solution is to train the model using a cleaned dataset and employ a separate algorithm to filter out these incorrect poses during inference.

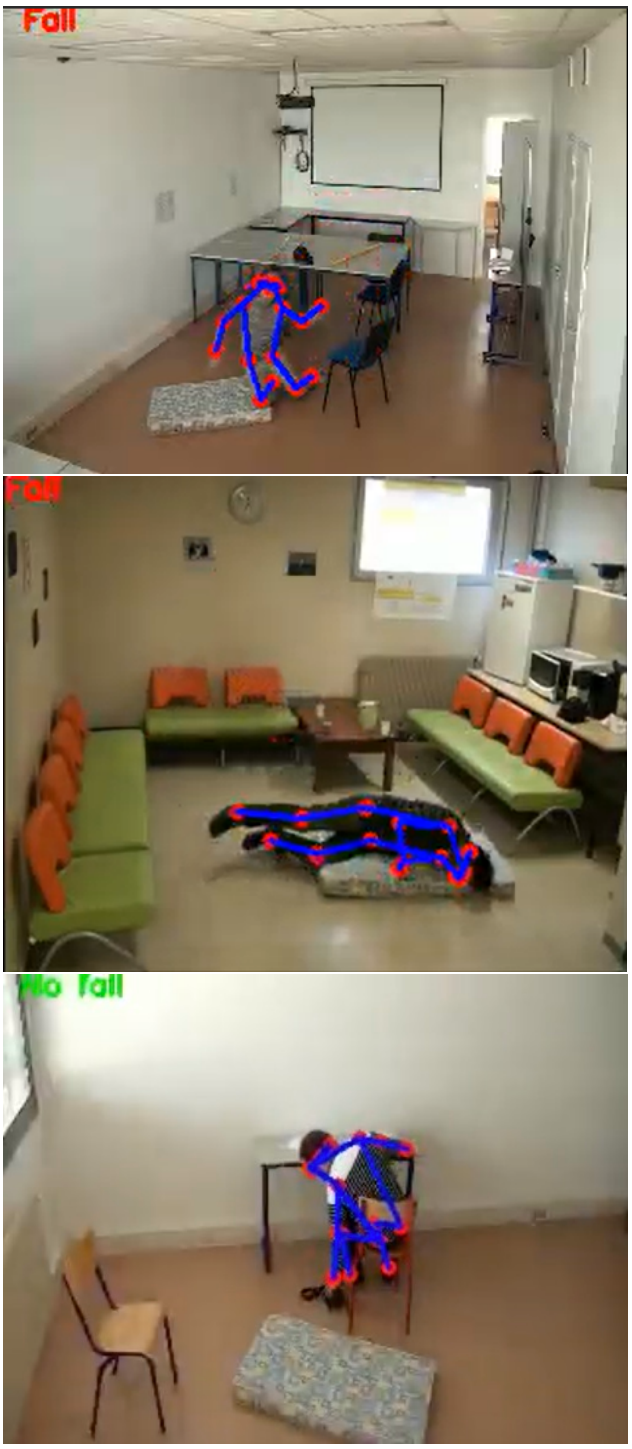



Figure 5.4: Demonstration of the proposed fall detection method. Top: detect "Fall" in the test video. Mid: detect "Fall" in validation video. Bottom: detect "Non-Fall" in the test video. Full videos: top, mid, bottom

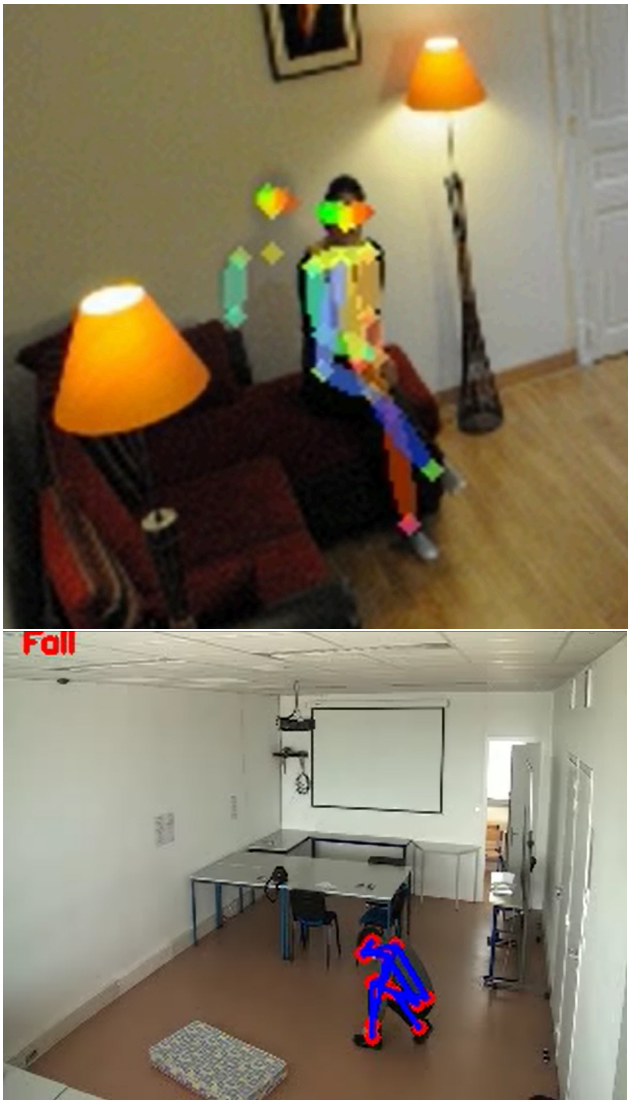



Figure 5.5: Failure cases of the proposed fall detection method. Top: incorrect human pose estimation by AlphaPose. Bottom: a failure case of STGCN-LE (full video)

# Chapter 6

# Conclusions

In summary, this report introduces a novel Graph Convolutional Network (GCN) architecture, termed Spatial Temporal Graph Neural Networks with Learnable Edges (STGCN-LE). The aim is to enhance the expressive capabilities of the original Spatial Temporal Graph Neural Networks (STGCN) architecture, particularly in the context of skeleton-based fall detection.

STGCN-LE operates on a spatial-temporal graph with human body joints as vertices and the human skeleton as edges. Each joint is spatially connected to neighboring joints within the same video frame, and temporally connected to its counterpart extracted from adjacent video frames. The AlphaPose toolbox is used to extract 2D joint coordinates from videos as the input features of each vertex. In each forward pass, graph convolutions are performed at both spatial and temporal dimensions to pass messages between neighbors. Instead of only using the initial adjacency matrix pre-defined by the human skeleton structure for message passing, the model is allowed to modify the adjacency matrix at different layers by adjusting edge weights or adding and removing edges. This is achieved by applying two learnable masks to the initial adjacency matrix. The mask weights are adjusted through backpropagation at training time and kept unchanged at inference time. This mechanism allows the networks to learn a spatial-temporal graph structure that is the most suitable for the downstream classification task. Experiments showed that learnable edges improve the expressive power of the networks, which leads to a better fall detection performance, compared to the original STGCN.

The efficacy of STGCN-LE is validated through cross-validation on test datasets, employing a proposed Real-time Fall Detection Simulation scheme. This evaluation method assesses the fall detection system in practical settings, incorporating lighting conditions and camera angles not present in the training data. It evaluates the fall detection method not on model classification accuracy, but on the algorithm's performance on videos, which highlights the system's effectiveness in real-world scenarios. This approach is motivated by a crucial observation regarding the evaluation of fall detection algorithms: the absence of a standardized evaluation framework across the field. Previous works often employed slightly different metrics and failed to report the train-test split of the datasets used. This lack of

standardization poses challenges for making meaningful comparisons between different approaches. Consequently, future work on fall detection algorithms should align the evaluation method with existing approaches.

Despite the method's susceptibility to certain human actions during daily living activities, stemming from the limited training data, the model still achieves performance comparable to other existing fall detection approaches. For our future work, we should train the STGCN-LE model on a multi-class human action dataset. This explicitly forces the networks to learn the characteristics of a variety of human actions in daily living activities, which should enhance the overall fall detection performance of the model.

Lastly, while the proposed skeleton-based fall detection method and other existing approaches exhibit good performance, ongoing advancements in pose estimation techniques are essential to further improve the methods. Skeleton-based fall detection relies heavily on pose estimation, where errors introduced at each stage of the pipeline can accumulate and impact the final classification outcome. Given the scarcity of fall detection data, the model may not be exposed to enough diverse examples during training to effectively learn the underlying patterns and variations in fall detection scenarios. As a result, the accumulated errors may not be adequately averaged out, leading to potential overfitting or limited generalization ability of the model. By addressing these challenges, future research endeavors can contribute to the continued progress and refinement of fall detection algorithms.

# Bibliography


[1] Ramakrishna Kakara, Gwen Bergen, Elizabeth Burns, and Mark Stevens, "Nonfatal and fatal falls among adults aged 65 years — united states, 2020–2021," *MMWR. Morbidity and Mortality Weekly Report*, vol. 72, no. 35, pp. 938–943, Sep. 2023. DOI: 10.15585/mmwr.mm7235a1.

[2] Kim Parker, Juliana Horowitz, and Brian Mahl, *Demographic trends and economic well-being*, Jun. 2016. [Online]. Available: https://www.pewresearch.org/social-trends/2016/06/27/1-demographic-trends-and-economic-well-being/.

[3] Alan Bradley, *Best medical alert systems with fall detection — u.s. news*, Feb. 2024. [Online]. Available: https://www.usnews.com/360-reviews/services/medical-alert-system/fall-detection.

[4] Sijie Yan, Yuanjun Xiong, and Dahua Lin, *Spatial temporal graph convolutional networks for skeleton-based action recognition*, arXiv:1801.07455, 2018.

[5] Sijie Yan, Yuanjun Xiong, Jingbo Wang, and Dahua Lin, *Mmskeleton*, https://github.com/open-mmlab/mmskeleton, 2019.

[6] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, *et al.*, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[7] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.

[8] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 863–10 872.

[9] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[10] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[12] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[13] Jianjun Yan, Xueqiang Wang, Jiangtao Shi, and Shuai Hu, "Skeleton-based fall detection with multiple inertial sensors using spatial-temporal graph convolutional networks," *Sensors*, vol. 23, no. 4, p. 2153, 2023.

[14] Weiming Chen, Zijie Jiang, Hailin Guo, and Xiaoyang Ni, "Fall detection based on key points of human-skeleton using openpose," *Symmetry*, vol. 12, no. 5, p. 744, 2020.

[15] Sungil Jeong, Seongwon Kang, and Injung Chun, "Human-skeleton based fall-detection method using lstm for manufacturing industries," in *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, IEEE, 2019, pp. 1–4. DOI: 10.1109/ITC-CSCC.2019.8793342.

[16] Yangsen Chen, Tianyi Zhou, Bowen Lei, Qiang Huang, Zicheng Zhang, and Wenqiang Ke, "Fall detection system based on real-time pose estimation and svm," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, IEEE, 2021, pp. 77–81. DOI: 10.1109/ICBAIE52039.2021.9390068.

[17] Muzaffer Aslan, Abdulkadir Sengur, and Mehmet Cem Ince, "Skeleton based efficient fall detection," *Journal of Faculty of Engineering and Architecture of Gazi University*, vol. 32, no. 4, pp. 1025–1034, 2017.

[18] Bo-Hua Wang, Jie Yu, Kuo Wang, Xuan-Yu Bao, and Ke-Ming Mao, "Fall detection based on dual-channel feature integration," *IEEE Access*, vol. 8, pp. 103 443–103 453, 2020.

[19] Beddiar Djamila Romaissa, Oussalah Mourad, Nini Brahim, and Bounab Yazid, "Fall detection using body geometry in video sequences," in *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2020, pp. 1–5. DOI: 10.1109/IPTA50016.2020.9286456.

[20] Vincenzo Dentamaro, Donato Impedovo, and Giuseppe Pirlo, "Fall detection by human pose estimation and kinematic theory," in *2020 25th international conference on pattern recognition (ICPR)*, IEEE, 2021, pp. 2328–2335.

[21] J. Dubois and J. Miteran, *Dat@ImViA: Fall Detection Dataset*, Search-Data.ubfc.fr, [Online; accessed 18 April 2024], 2014. [Online]. Available: http://search-data.ubfc.fr/imvia/FR-13002091000019-2024-04-09_Fall-Detection-Dataset.html.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, *et al.*, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].

[23] Lourdes Martínez-Villaseñor, Hiram Ponce, Jorge Brieva, Ernesto Moya-Albor, José Núñez-Martínez, and Carlos Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019.

[24] Heilym Ramirez, Sergio A Velastin, Ignacio Meza, Ernesto Fabregas, Dimitrios Makris, and Gonzalo Farias, "Fall detection and activity recognition using human skeleton features," *Ieee Access*, vol. 9, pp. 33 532–33 542, 2021.