Synthetic IMU Datasets and Protocols Can Simplify Fall Detection Experiments and Optimize Sensor Configuration

Jie Tang [ID], Bin He [ID], *Senior Member, IEEE*, Junkai Xu, Tian Tan [ID], Zhipeng Wang [ID],
Yanmin Zhou [ID], and Shuo Jiang [ID], *Member, IEEE*

Abstract—**Falls represent a significant cause of injury among the elderly population. Extensive research has been devoted to the utilization of wearable IMU sensors in conjunction with machine learning techniques for fall detection. To address the challenge of acquiring costly training data, this paper presents a novel method that generates a substantial volume of synthetic IMU data with minimal actual fall experiments. First, unmarked 3D motion capture technology is employed to reconstruct human movements. Subsequently, utilizing the biomechanical simulation platform Opensim and forward kinematic methods, an ample amount of training data from various body segments can be custom generated. Synthetic IMU data was then used to train a machine learning model, achieving testing accuracies of 91.99% and 86.62% on two distinct datasets of actual fall-related IMU data. Building upon the simulation framework, this paper further optimized the single IMU attachment position and multiple IMU combinations on fall detection. The proposed method simplifies fall detection data acquisition experiments, provides novel venue for generating low cost synthetic data in scenario where acquiring data for machine learning is challenging and paves the way for customizing machine learning configurations.**

Jie Tang, Bin He, Zhipeng Wang, Yanmin Zhou, and Shuo Jiang are with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China, and also with the Frontiers Science Center for Intelligent Autonomous Systems, Shanghai 200120, China (e-mail: tangjie1953479@tongji.edu.cn; hebin@tongji.edu.cn; wangzhipeng@tongji.edu.cn; yanmin.zhou@tongji.edu.cn; jiangshuo@tongji.edu.cn).
Junkai Xu is with IMU Master Technology Company Ltd., Shanghai 200240, China (e-mail: xujunkai@imumaster.com).
Tian Tan is with the Department of Radiology, Stanford University, Stanford, CA 94305 USA (e-mail: alanttan@stanford.edu).
Digital Object Identifier 10.1109/TNSRE.2024.3370396

Index Terms—**Fall detection, IMU, Opensim, data augmentation.**

I. INTRODUCTION

FALLING is one of the main causes of injuries among the elderly, exerting profound impacts on their health and quality of life. It is estimated that approximately 684,000 cases of fatal fall injuries occur annually, positioning them as the second leading cause of unintentional injury-related deaths, following closely behind road traffic injuries. Notably, within all fall-related incidents, individuals aged 60 and above exhibit the highest mortality rates. Elderly individuals face the greatest risk of death or severe injury due to falls, with this risk escalating as age advances. For instance, in the United States, 20% to 30% of elderly individuals involved in fall accidents endure moderate to severe injuries, encompassing abrasions, hip fractures, and cranial trauma, among others [1]. Consequently, the pursuit of fall detection research and its practical implementation hold crucial socio-medical significance, offering the potential to furnish comprehensive care and assurance for the well-being of the elderly.

A diverse array of existing methodologies exists within the realm of fall detection. From the sensing perspective, these can be broadly classified into two categories: camera-based and wearable Inertial Measurement Unit (IMU) based approaches [2]. In contrast to camera-based fall detection systems, wearable IMU-based systems are characterized by their ease of deployment and immunity to occlusion challenges. Early implementations of fall detection systems utilizing IMU sensors employed simplistic threshold-based detection algorithms, whereas the feasibility of machine learning techniques in enhancing the recognition of fall movements has been demonstrated [3], [4]. Wearable devices, particularly those incorporating IMUs, in combination with machine learning methodologies, play a pivotal role in fall detection. Queralta et al. established a fall detection system encompassing IMU sensor nodes, edge gateways, cloud services, and end-user applications [5]. Liu et al. fused IMU sensor networks with Doppler radar systems, resulting

in a significant reduction in false positive rates within fall detection systems [6]. Jian and Chen integrated IMU sensors into a vest tailored for the elderly, transmitting data via Bluetooth to a mobile smartphone running a fall detection program based on the k-Nearest Neighbors (KNN) algorithm, thereby enabling automatic alerts upon fall incidents [7]. Xu et al. introduced a fall detection algorithm founded on a CNN-LSTM (Convolutional Neural Network-Long Short-Term Memory) composite network. This approach leverages neural networks to automatically extract features and conduct classification, obviating intricate preprocessing procedures [8].

The utilization of wearable IMUs in conjunction with machine learning for fall detection necessitates a substantial quantity of IMU data encompassing both fall and non-fall activities. However, the cost of acquiring fall detection data is high. Conventional data collection approaches typically involve tracking the IMU data of elderly individuals or recruiting younger participants to perform motion experiments for data collection. Lisowska et al. collected data from 20 volunteers wearing Silmee devices during simulated falls and daily activities [9]. Some researchers have publicly released experimental datasets, such as the MobiFall dataset [10] and the SisFall dataset [11]. Bagala et al. established the FARSEEING database, noting that a comprehensive shared real-world fall database could enhance the understanding of fall processes [12].

Gathering IMU data through actual experiments necessitates addressing two vital pre-experiment considerations: experimental protocol design and IMU sensor placement. Once the experimental protocol is established, post-experiment parameter adjustments become challenging, often requiring costly reiterations of the experiments. Despite comprehensive experimental protocol designs being proposed to contrast fall and non-fall movements [13], variations in action designs persist across experiments and datasets. This diversity may stem from considerations of experimental complexity or differing research foci. Özdemir and Barchan included 20 fall movements and 16 non-fall movements in their experiments [14], whereas Alfred Wertner et al.'s research examined only 4 fall and 10 non-fall movements [15]. Wang et al. highlights that the datasets utilized in most papers often fail to encompass all types of falls found in real-world scenarios [16]. Variations in IMU quantities and placement positions are also evidently different across different datasets. Tong et al. propose that the wrist, hip, and thigh are not suitable positions for accelerometer-based devices due to their high frequency and complexity of motion. Instead, they suggest that regions below the neck and above the waist, specifically the upper torso, are more appropriate for distinguishing fall-related characteristics using acceleration data [17]. Nukala et al. designed an artificial neural network for fall detection using IMU data from the waist and back, achieving a sensitivity of 97.0% and specificity of 97.2% [18]. In the experiments conducted by Ojetola et al., two SHIMMER sensors were attached to the participant's chest and thigh [19]. Martínez-Villaseñor et al. proposed a fall detection method based on multi-inertial sensor skeletons using ST-GCN and employed the Up-Fall dataset [20], which places IMUs on

the neck, waist, thigh, wrist, and ankle for model training and testing [21]. In summary, the requirement for participants to wear IMU devices during motion experiments for data collection is associated with both high costs and limited flexibility. Furthermore, the integration of these datasets presents challenges due to variations in the experimental designs across different studies, potentially resulting in a diminution of data set quality.

To address data limitations, researchers have proposed various approaches: (1) Data Augmentation Techniques: Researchers have employed techniques like rotation to augment limited fall-related IMU datasets. Theodoridis et al. augmented training data by rotating the original data around random angles within the range of $[-10, 10]$ degrees, enhancing the model's robustness to IMU rotations [22]. Yhdego et al. inverted the sign of sensor data across all three axes to simulate sensor inversion [23]. (2) Transfer Learning Methods: Butt et al. utilized transfer learning by retraining a VGG-16 convolutional neural network (CNN), initially trained on a large-scale image dataset, to learn features relevant to human actions for fall detection tasks [24]. Yhdego et al. repurposed the AlexNet architecture of deep convolutional neural networks, modifying and retraining the last three layers of the AlexNet architecture. They employed wavelet-based time-frequency analysis to represent sensor data through scaleograms, which were then fed into the AlexNet framework [23]. (3) Biomechanical Simulation Tools for Synthetic Data: Mastorakis et al. utilized Opensim simulation software to generate synthetic data. They extracted a person's height and body orientation before a fall from depth camera-captured videos. These pieces of information are utilized to tailor the height of skeletal models and the falling motion, thereby generating vertical acceleration at the center of mass [25].

Among these approaches, the biomechanical simulation-based method has gained increasing attention from researchers and has seen successful applications in other research domains. Recinos et al. utilized an integrated framework of Opensim and Unity to estimate ground reaction forces in a simulation environment using IMU data collected from actual motion experiments [26]. Jiang et al. used fatigue detection motion experiment data from a limited number of participants to drive model movements in Opensim. Simulated individuals with different physiological attributes executed movements, thereby enhancing the IMU dataset. Their research demonstrated that this model-based data augmentation approach outperformed traditional methods such as rotation, random noise, amplitude warping, time warping, scaling, and proposed methods [27]. Uhlenberg et al. introduced a joint simulation framework consisting of a biomechanical human model and wearable inertial sensor models. A total of 960 virtual inertial sensors were connected to the lower limbs and shoe model of the human body in Opensim. They simulated gait patterns of hemiparetic patients using motion capture data, generating synthetic accelerations and angular velocities based on the inertial sensor model, enabling comprehensive analysis of gait events [28]. The aforementioned studies have substantiated the feasibility of employing simulation techniques to synthesize

IMU data. This approach holds potential significance for addressing fall detection challenges in scenarios characterized by substantial data paucity.

Due to the significant application value and market demand for fall detection systems, continuous research endeavors have led to ongoing innovations in this domain. Matos-Carvalho et al. demonstrated that a single LSTM layer can identify patterns related to falls by processing data from sensors such as accelerometers or gyroscopes, thus making it applicable for fall detection [29]. They further noted that employing multiple LSTM layers enhances system performance. Manikandan et al. proposed a fall detection solution utilizing a vision-based approach [30]. They extracted a set of keypoints of the subject from RGB images recorded by a single camera using the pre-trained AlphaPose network. A combined architecture of CNN and LSTM was applied to make the system more robust in distinguishing between falls and fall-like activities. Amsaprabhaa et al. introduced a multi-modal feature fusion approach, the MSTSK-GFF framework [31]. This method leverages a Spatio-Temporal Graph Convolutional Network (STGCN) and a 1D-CNN network model to generate two sets of spatio-temporal kinematic gait features. The features generated by these two models are then combined through a cascaded feature fusion process, constructing a classification model for fall detection.

As per our investigation, existing IMU datasets in the field of fall detection suffer from issues such as limited data volume and inconsistent experimental paradigms. In contrast, an abundance of video datasets capturing falls is available, offering the potential to replicate more realistic fall scenarios. Additionally, the feasibility of utilizing biomechanical simulations to generate synthetic IMU data has been established and met with successful application, while its potential remains underutilized in the domain of fall detection. Considering the above factors collectively, the integration of vision-based 3D motion reconstruction with Opensim biomechanical simulation enables cross-modal data generation (from video to IMU). This methodology holds promise within the domain of fall detection, as it effectively addresses the challenge of scarce fall-specific data availability. However, research on this approach remains scarce.

To address the challenges of acquiring costly fall training data, optimizing sensor positions and combinations, this paper proposes a methodology for fall detection that leverages biomechanical simulation software, such as Opensim, to generate synthetic IMU data for training machine learning models. The resulting models exhibit favorable performance on actual fall experiment IMU datasets. The contributions of this paper are summarized as follows:

1. Cost-effective Data Generation: In response to the high cost of fall-related data, this paper employs Opensim to generate synthetic IMU data of human movements reconstructed from video. The dataset encompasses various fall and non-fall movements. By manipulating model dimensions and IMU placement poses, the methodology enhances synthetic data through data augmentation, yielding a robust synthetic dataset.

2. Customized IMU positions: Through simulation, this study customizes the generation of synthetic IMU data from distinct body regions, investigating the impact of IMU placement on fall detection.

3. Optimized the Combination of Multiple IMUs: Based on simulation, the combination of synthetic IMU data from multiple body regions has been optimized to provide a comprehensive representation of falling actions.

II. METHODS

A. Overview

The system is broadly divided into three main components: driving Opensim skeletal models, generating synthetic IMU data, and training/testing machine learning models (Fig.1). OpenSim is a sophisticated biomechanical modeling and simulation software platform that enables the creation and analysis of musculoskeletal models to study human movement and biomechanics [32]. Initially, to drive Opensim skeletal models, multiple cameras capture human fall actions, followed by OpenPose detecting key body points in the videos. The three-dimensional coordinates of these key points are calculated using the triangulation algorithm. Subsequently, human movements are reconstructed via Opensim's marker-based inverse kinematics. Simulated IMUs can be placed freely on the skeletal model, and thus the synthetic IMU raw data can be extracted during model movements. Following this, multiple machine learning models, such as LSTM networks, are trained using the synthetic IMU data and tested on both publicly available IMU datasets and experimentally collected IMU datasets to demonstrated the feasibility of the proposed methods.

B. 2D Pose Estimation

Traditional methods for reconstructing human motion based on optical markers rely on expensive optical motion capture equipment. To achieve markerless motion reconstruction without specialized equipment, this study employs multiple cameras to record human motion videos. Human body pose estimation is performed on the recorded videos from each camera, yielding two-dimensional pixel coordinates of human keypoints (Fig.2).

OpenPose, a widely used 2D human pose estimation algorithm, accomplishes pose detection by analyzing human keypoints within the images [33]. It employs a multi-stage deep learning architecture, encompassing a convolutional neural network (CNN) stage for body part detection and a pose association stage for linking body parts. In the initial stage, CNN is employed to detect body parts in the images, such as the head, hands, and feet. The network employs convolution, pooling, and fully connected layers to extract features, generating confidence maps and positional information associated with each keypoint. In the subsequent stage, keypoint confidence maps generated in the preceding stage are utilized to associate body parts. This stage involves generating part affinity fields to describe the associational information connecting various body parts. By detecting associations between connected body parts, the algorithm discerns the
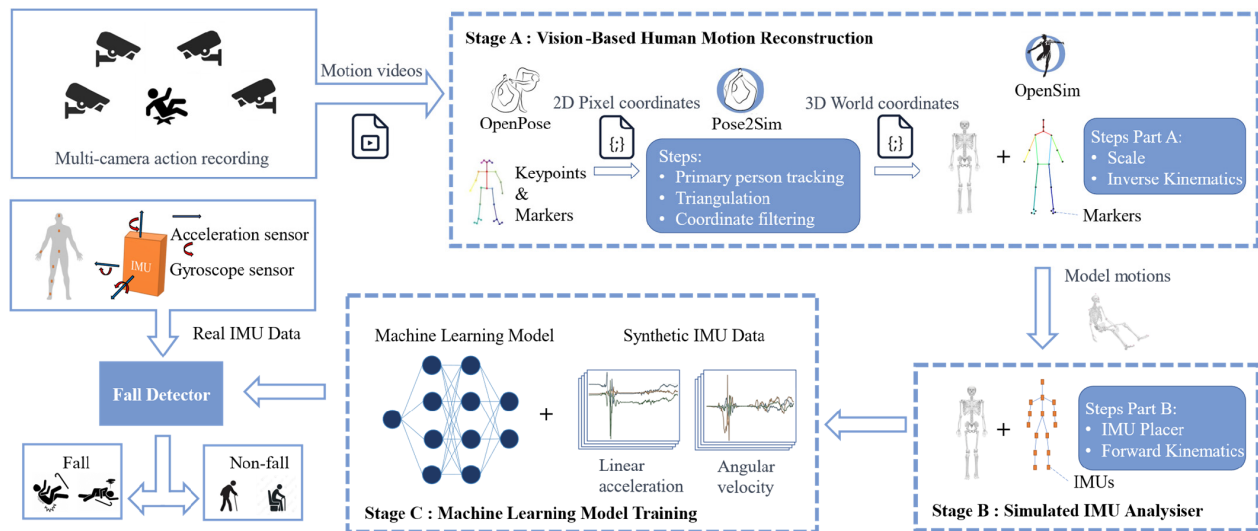
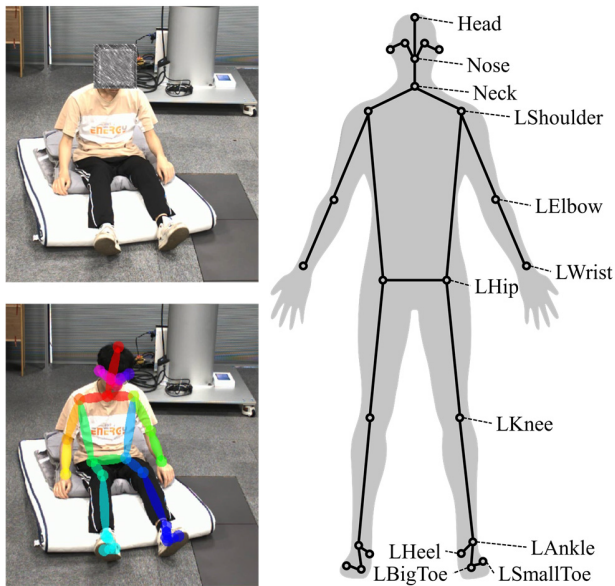Fig. 1.   Experimental framework of the proposed system.

Fig. 2.   2D human pose estimation in video. The Body_25B model, employed for human pose detection, encompasses a total of 25 distinct keypoints of the human body.

structure of the human pose. Upon obtaining preliminary keypoint and association information, refinement and filtering of detected poses ensue. Precise estimation of body part positions is achieved by leveraging associational information based on connected body parts, followed by thresholding and non-maximum suppression to eliminate redundant keypoints. The ultimate output comprises coordinates and confidences of detected human keypoints, alongside associated connection information, which collectively represent the pose of the human body in the image.

C. 3D Coordinate Calculation

The marker-based inverse kinematics solving algorithm of Opensim requires the three-dimensional world coordinates

of all marker points. This necessitates camera calibration to determine the camera's pose relative to the ground. Subsequently, the pixel coordinates of key points in camera videos are converted to world coordinates. This paper employed the classical chessboard calibration method [34]. Pose recognition is initially performed on the human motion videos recorded by each camera. Leveraging the multi-camera intrinsic and extrinsic parameters obtained during camera calibration, a triangulation method converts the 2D coordinates of human key points from different camera views captured at the same moment in a video frame into 3D coordinates in the world coordinate system. Finally, a Gaussian filter is applied to smooth the 3D coordinates of all key points, aiming to refine the motion obtained through the inverse kinematics calculation step.

Direct linear transformation is a classic triangulation algorithm, and incorporating weighted confidence of each keypoint provides an effective enhancement to this method [35]. Let $Q = (x, y, z, 1)^T$ be the three-dimensional homogeneous coordinates of a keypoint in the world coordinate system, $q = (u, v, 1)^T$ represents its two-dimensional homogeneous coordinates in the pixel coordinate system, $P = (P_1, P_2, P_3)^T$ is the camera projection matrix obtained from the camera calibration process, λ is the scale factor, and c is the confidence of the keypoints. According to camera geometry theory, the following equation can be deduced:

$$P Q = \lambda q \qquad (1)$$

Multiplying both sides of the equation by q, we get:

$$q \times P Q = 0 \qquad (2)$$

Utilizing the confidence of keypoints for weighting:

$$c \times (P_1 - u P_3) Q = 0$$
$$c \times (P_2 - v P_3) Q = 0 \qquad (3)$$

For N cameras, there are 2N equations, which can be represented as:
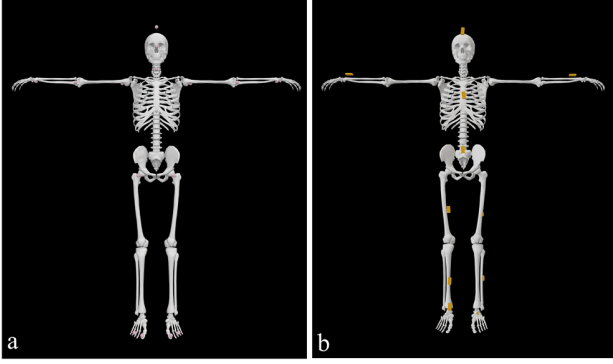
$$A Q = 0 \qquad (4)$$

Fig. 3. a. The Body_25B skeletal model of the human body. The pink spheres at the joints in the figure serve as both key points for human pose detection and virtual markers for inverse kinematics calculations. b. Simulated IMU placement. The orange cuboids rigidly attached to the skeleton represent the simulated IMUs.

Using the Singular Value Decomposition (SVD) of matrix A to solve for the least squares solution of vector Q:

$$A = USV^T$$
$$Q = (V_{14}/V_{44}, V_{24}/V_{44}, V_{34}/V_{44}, 1)^T \quad (5)$$

The last column of matrix V transformed into homogeneous coordinates represents the sought solution Q.

D. Inverse Kinematics Calculation

This step aims to drive the model to perform human actions of interest. Specifically, it involves deducing joint movements that can generate a given motion trajectory (Fig.3.a). This problem is formulated as an optimization task, where joint angles are adjusted to minimize the discrepancy between simulated and measured trajectories, as depicted in (6). Due to the highly nonlinear nature of this optimization problem, numerical optimization methods are employed for its solution. For solving this least squares problem, a general quadratic programming solver is utilized with a convergence criterion set at 0.0001 and a maximum iteration limit of 1000 iterations [36].

Describing the motion of Opensim skeletal models involves employing joint angles and translational transformations, denoted as generalized coordinates q. Here, x_i^{exp} represents the three-dimensional coordinates of the marker points obtained through experiments, while $x_i(q)$ signifies the three-dimensional coordinates of simulated marker points at the given q; q_j^{exp} denotes the generalized coordinate values obtained from experiments, and w_i represents the weights assigned to the marker points. Therefore, the problem addressed by the inverse kinematics computation is a weighted least squares problem:

$$\min_{q} \left[\sum_{i \in markers} w_i \left\| x_i^{exp} - x_i(q) \right\|^2 + \sum_{j} \omega_j \left(q_j^{exp} - q_j \right)^2 \right] \quad (6)$$

E. Forward Kinematic Analysis

Forward kinematic analysis is a method used to determine the motion of specific segments within a biomechanical model given certain joint angles. In this analysis, the focus is on calculating the positions, velocities, and accelerations of various segments of the model during motion, starting from an initial state. Initially, the joint angles and translational transformations at the initial moment are provided, determining the model's initial state. Subsequently, based on the joint angles and the model's geometric structure, the position, angular velocity, and acceleration of the segments of interest are solved using the Newton-Euler equations. Over multiple time steps, the calculations are iterated from the initial pose, ultimately providing the kinematic information of the segments for the given joint angles throughout the entire motion.

Taking a simple two-link system as an example, the angular velocity of the next link is the sum of the angular velocity of the previous link and the angular velocity introduced by the joint's rotation. Let θ be the angle of rotation joint, n be the rotation axis, and L be the position vector between the links. The angular velocity and acceleration transmission formula for the link system can be expressed as:

$$w_{k+1} = w_k + \dot{\theta}_{k+1} \cdot n_{k+1}$$
$$\dot{v}_{k+1} = \dot{v}_k + \dot{w}_k \times L_{k+1} + w_k \times (w_k \times L_{k+1}) \quad (7)$$

To calculate acceleration, angular velocity, and other relevant information of various body segments during human motion, Opensim employs a more refined forward kinematic analysis algorithm. By customizing the placement of simulated IMUs, it is possible to obtain three-axis acceleration and angular velocity data for that position during the motion of the skeletal model (Fig.3.b).

F. Fall Detection Model Training

After obtaining the necessary synthetic IMU data, we utilized the generated synthetic dataset for model training. LSTM networks were selected for fall detection due to their capability to model long-term dependencies in sequential data, which is crucial for analyzing IMU outputs. Compared to other Recurrent Neural Network (RNN) models, such as Gated Recurrent Units (GRU), LSTM's intricate architecture enhances its ability to capture extended temporal patterns, leading to superior performance in our evaluations. Moreover, the effectiveness of LSTM networks for fall detection has been validated in related studies [29]. The study also explored alternative models, including GRU, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Convolutional Neural Networks (CNN), among others. Among these models, LSTM demonstrated the highest accuracy and robustness.

LSTM, a variant of RNN, is designed for modeling sequential data and addresses the vanishing gradient problem in traditional RNN. The fundamental structure of LSTM includes the cell state, input gate, forget gate, and output gate [37], [38].

Firstly, the cell state is at the core of the LSTM network and is responsible for carrying long-term information. It can be thought of as a channel that runs through the entire network.
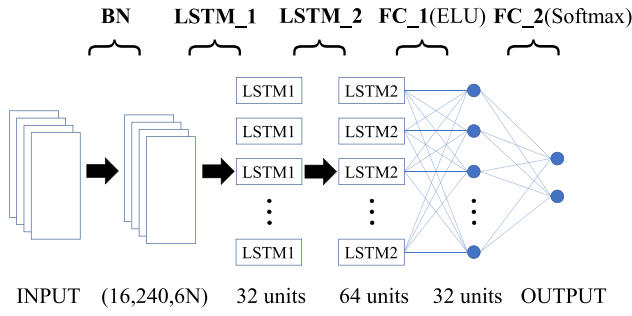
Fig. 4. Structure of the LSTM model.

Fig. 5. Experimental setup, using four cameras to capture motion from different angles while wearing 7 IMU sensors.

At each time step, LSTM updates the cell state using the input data and the output from the previous time step through the input gate and the forget gate.

Secondly, The input gate decides which information to add to the cell state. It first computes a candidate value based on the input data and the output from the previous time step. Then, it uses logistic regression to determine which parts of the cell state should be updated. The formula for the input gate is as follows:

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \tag{8}$$

Following that, the forget gate decides which information in the cell state to forget. Similar to the input gate, the forget gate employs logistic regression to determine the degree of forgetting for each part. The formula for the forget gate is as follows:

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \tag{9}$$

The cell state is updated by multiplying the candidate value from the input gate and the forgotten parts of the cell state, and then adding new information. The formula for updating the cell state is as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \tag{10}$$

Lastly, the output gate decides which information will be used for the current time step's output. Like the input gate and forget gate, the output gate employs logistic regression to determine which parts of the output to consider. The formula for the output gate is as follows:

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \tag{11}$$

The output state is the final output of the LSTM and is a transformation of the cell state based on the output gate's information and the cell state's transformation. The formula for calculating the output state is as follows:

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{12}$$

Through these gating mechanisms and state updates, LSTM effectively tackles the long-term dependency problem, allowing the network to capture information in sequences better.

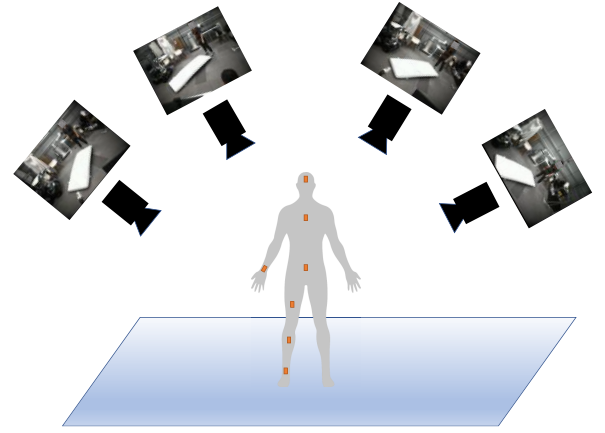In this paper, a LSTM network composed of a Batch Normalization (BN) layer, two LSTM layers, and two fully connected layers is primarily used (Fig.4). The implementation is practically performed using TensorFlow. The training process involves using a cross-entropy loss function and the Adam optimizer with a learning rate of 0.0001. A batch size of 16 and 50 training epochs are employed. The chosen feature is the triaxial acceleration and triaxial angular velocity from IMUs. Each action is represented as a tensor of size $240 \times N \times 6$ (where N represents the number of IMUs). There are two categories, and one-hot encoding is utilized, where the fall action label is encoded as "01" and the non-fall action label as "10".

III. EXPERIMENTS

A. Experimental Setups

The experiments were conducted at the Kaiwu Laboratory [39], Tongji University, Zhangjiang AI Island. Kaiwu lab focuses on human-machine interaction datasets and embodied learning algorithms for intelligent robots. An industrial camera system was utilized, with two cameras positioned on the left front side and two on the right front side of the participant. The cameras on the same side were positioned at different horizontal heights. The Luster Capture software was used to synchronize video recording and decompression. The resulting videos had a resolution of 2048×1536, a frame rate of 30 frames per second, and a duration of at least 8 seconds. Seven Xsens MTw2 IMU sensors were affixed to the participants during the experiment, synchronously collecting experimental IMU data (Fig.5).

B. Experimental Protocol Design

Sixteen fall actions and sixteen non-fall actions were designed (TABLE I). Each action was performed three times, with action videos recorded and experimental IMU data collected. The experimental protocol design drew inspiration from existing research [13], [21], [40], encompassing both common fall actions and daily activities, while also incorporating some potentially confusing movements.

The present study has been granted ethical approval with the approval number TJDXSR055. All human participants

Non-fall	Fall
from standing to lying on the bed	fall forward with arm protection
from lying down to sitting	kneel down
from standing to sitting on the sofa	kneel down and then lying down
from standing to sitting on the chair	falling forward to the right
squatting horse stance	falling forward to the left
walking	fall forwards and quick recover
running	fall forwards and slow recovery
walking backwards	fall backwards and sit on the floor
bending forward at a 90-degree angle	fall backwards and lie on the floor
bending over to pick up an object from the ground	fall backwards and make contact with the floor on the right side
walking, stumbling, and recovering balance	fall backwards and make contact with the floor on the left side
walking with a limp	fall to the right and then lie down
from squatting to standing	fall to the right and then recover
bending over while walking	fall to the left and then lie down
rolling over on the bed	fall to the left side and then recover
from sitting to standing, followed by walking forward	roll from lying down on the bed to the floor

TABLE II
PARAMETER CONFIGURATION DETAILS

Parameter Name	Value
frame_range	[0,240]
frame_rate	30
pose_model	BODY_25B
tracked_keypoint	Head
error_threshold_tracking	20
error_threshold_triangulation	20
likelihood_threshold	0.3
interpolation	cubic
interp_if_gap_smaller_than	20
filter_type	gaussian
sigma_kernel	3

involved in this study provided informed consent, and their privacy and rights were safeguarded.

C. Experimental Parameter Configuration

The primary parameter configuration for Opensim model motion reconstruction based on camera video is listed in TABLE II. The parameter values are set based on the recommended values from Pose2Sim [35] and further adjusted according to the actual performance of the program during execution.

Based on human motion videos recorded by multiple cameras, filtered human keypoint coordinates were obtained. Using these marker points, inverse kinematics calculations were performed in Opensim to derive model movements (Fig.6). Subsequently, simulated IMUs were placed on different body parts of the model, and forward kinematic analysis was conducted to generate acceleration and angular velocity data for the simulated IMUs.

D. Model Training and Testing

Based on the experimentally reconstructed 32 actions, a set of simulation configurations was capable of generating 32 IMU data samples. Data augmentation was performed
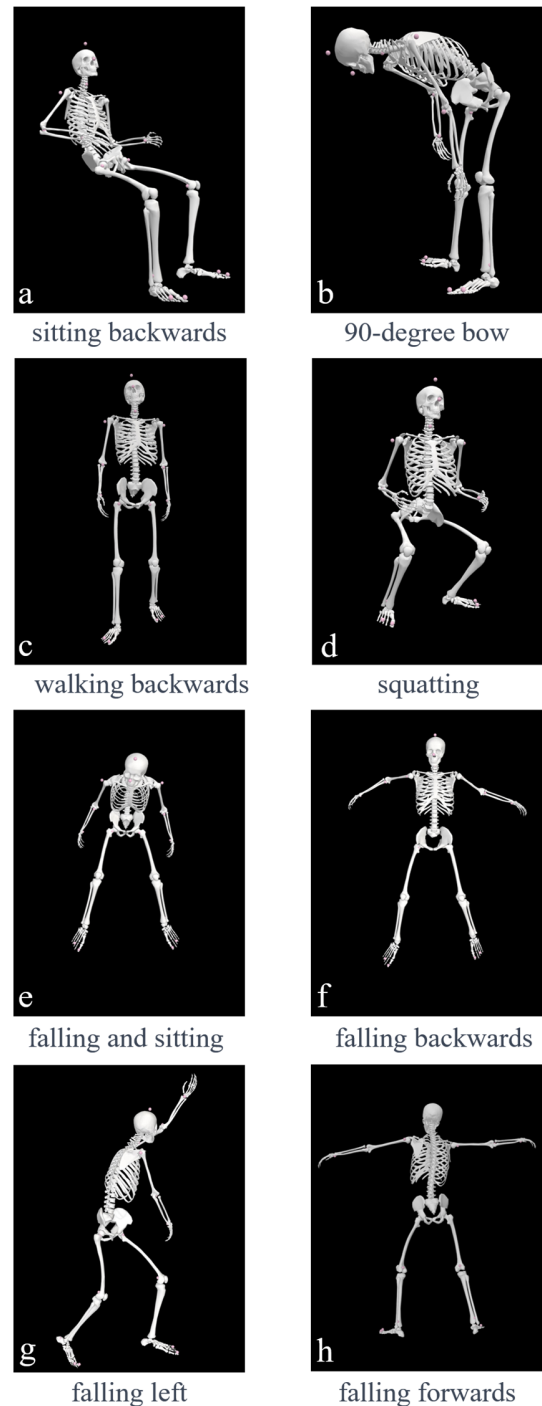

Fig. 6. Examples of action reconstruction. In the figure, panels (a), (b), (c), and (d) depict frontal views of four distinct non-falling actions, while panels (e), (f), (g), and (h) illustrate top views of four different falling actions.

by scaling the simulation model and adding a small random amount to the pose of each simulated IMU. In total, 32×6 IMU data samples were obtained, which constituted the training dataset.

Two test sets were used in this study, both comprising actual fall experiment IMU datasets.

Experimental Dataset A: Seven Xsens MTw2 IMU sensors were worn by participants to collect IMU data (three-axis

acceleration and angular velocity, or other data such as Euler angles) during action execution. The dataset involved six participants performing 16 fall actions and 16 non-fall actions, each repeated 2 or 3 times, resulting in a total of 513 valid samples. IMUs were attached to the head, chest, waist, front of the right thigh, front of the right shank, front of the right ankle, and right wrist.

Public Dataset B: An authentic dataset of actual fall experiments collected by Özdemir and Barshan was used as a test set [14]. This dataset included 17 participants, 20 fall actions and 16 non-fall actions, each repeated 5 or 6 times, totaling 3313 fall experiment data samples. Each sample comprised acceleration and angular velocity data from six Xsens MTw wearable IMU sensors, placed on the neck, chest, waist, right thigh, right shank, and right wrist.

To verify the feasibility of training fall detection models using synthetic IMU data, identical models were trained using both synthetic and experimental data. The performance of these models was tested on the same datasets, and a comparison was made using metrics such as accuracy, precision, recall, and F1-score.

IV. RESULTS

A. Synthetic IMU Data

During the experiments, we synchronized the collection of IMU data and the recording of camera videos. Below are some example plots illustrating the time-aligned comparison between experimental and synthetic sensor samples. For this comparison, we selected IMU data from a participant's experiment and compared it with IMU data synthesized from the corresponding video-reconstructed actions. For non-fall actions, there is a high degree of similarity between the synthetic and experimental data, with their curves closely aligning (Fig.7). For fall actions, the deviation between synthesized IMU data and experimental data is more significant compared to non-fall actions (Fig.8).

The ideal outcome is that the synthetic IMU data closely resembles experimental IMU data. However, due to discrepancies between the simulated IMU poses on the model in the simulation environment and the actual IMU placements on the subjects during the experiment, coupled with the inability of video reconstruction of the model's motion to perfectly replicate real human movements, differences between synthetic and experimental data are expected and can be explained. Further analysis on these discrepancies is provided in Section E of the discussion.

B. Model Test Results

The performance of the model trained using synthetic IMU data from the shank on the two datasets is as follows. Additionally, for comparison with models trained on experimental data, a model was trained using data from one participant on both datasets, and the remaining participants' data were used for testing.

From TABLE III, it is evident that the LSTM model trained using synthetic data achieves an accuracy of 91.99% on Dataset A, which is 4.49% higher than the accuracy of the
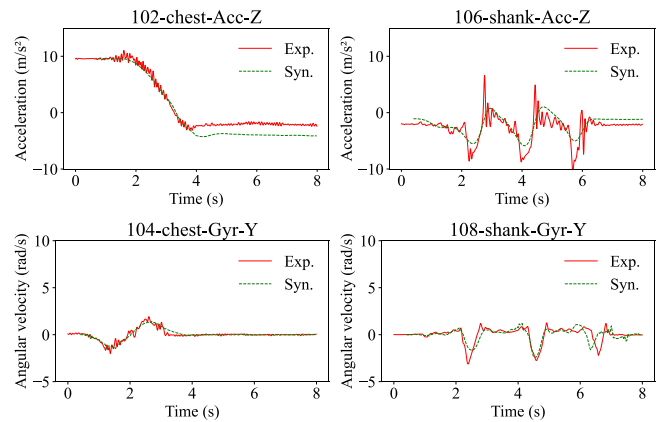

Fig. 7. Non-fall actions comparing experimental and synthetic sensor samples. The captions "Exp." and "Syn." are abbreviations for "experimental" and "synthetic," respectively. The title of the image "A-B-C-D" consists of four parts: A represents the action code, where 102, 106, 104, and 108 correspond to "from lying down to sitting," "walking," "from standing to sitting on the chair," and "walking backward," respectively; B indicates the placement location of the IMU; C denotes the type of IMU data, with "Acc" for acceleration and "Gyr" for angular velocity; D specifies the axis of the data.

TABLE III
PERFORMANCE OF LSTM MODEL TRAINED WITH
LOWER LEG IMU DATA ON DATASET A

	accuracy	precision	recall	F1
synthetic	**91.99%**	88.66%	98.01%	93.10%
experimental	**87.50%**	86.22%	92.80%	89.39%

TABLE IV
PERFORMANCE OF LSTM MODEL TRAINED WITH
LOWER LEG IMU DATA ON DATASET B

	accuracy	precision	recall	F1
synthetic	**86.62%**	89.39%	86.02%	87.64%
experimental	**87.15%**	89.57%	87.40%	88.16%

model trained with experimental data used as comparison. Additionally, the synthetic data-trained model outperforms the experimental data-trained model in the other three performance metrics. However, on Dataset B, which encompasses a more extensive set of samples, the accuracy of the model trained with synthetic data drops to 86.62%, resulting in a 0.53% decrease compared to the experimental data-trained model (TABLE IV).

C. Accuracy for Different Actions

Using synthetic three-axis acceleration data from the shank as features, the LSTM model was tested on the dataset B. The correct classification rates for different actions are calculated and presented in TABLE V. For comparison, the results of the model trained on experimental data are also provided. The results indicate that the four actions with the highest misclassification rates on the dataset B are transitioning from standing to lying on a bed, transitioning from lying down to sitting, falling to the ground on the left side and recovering

TABLE V
ACCURACY FOR DIFFERENT ACTIONS

action	synthetic	experimental
walking forwards	100%	100%
walking backwards	100%	100%
jogging, running	100%	100%
squatting, then standing up	92.05%	100%
bending about 90 degrees	100%	100%
bending to pick up an object on the floor	100%	100%
walking with a limp	100%	100%
stumbling with recovery	100%	100%
bending while walking	100%	100%
coughing or sneezing	100%	100%
sitting onto a chair (hard surface)	100%	100%
sitting onto a sofa (soft surface)	100%	100%
Squatting horse stance	78.41%	95.12%
from vertical to sitting onto a bed	100%	100%
from vertical to lying on the bed	**0%**	**0%**
from lying to sitting	**0%**	**0%**
falling forward to the floor	100%	100%
falling to the floor with arm protection	100%	100%
falling down on the knees	100%	100%
falling down on the knees and then lying	100%	100%
falling on the floor and quick recovery	**58.14%**	**58.23%**
falling on the floor and slow recovery	90.91%	100%
falling forward to the right	100%	100%
falling forward to the left	100%	100%
falling backward, ending sitting	96.25%	88.61%
falling backward, ending lying	96.55%	100%
falling backward to the right	98.88%	98.86%
falling backward to the left	98.05%	100%
falling to the right, ending lying	100%	100%
falling to the right and recovery	96.55%	76.92%
falling to the left, ending lying	73.56%	100%
falling to the left and recovery	**11.63%**	**57.69%**
rolling out of bed and going on the floor	100%	100%
step off the podium	100%	100%
falling on the floor vertically	95.51%	100%
falling down slowly slipping on a wall	96.59%	100%

quickly, and falling and quickly recovering. Similarly, during testing on Dataset A, the results indicate that the three non-fall actions—transitioning from standing to lying on the bed, transitioning from lying to sitting, and rolling over in bed are prone to being misidentified as fall actions. For other actions, the model demonstrates accurate classification in the majority of cases.

D. IMU Placement Positions

Synthetic IMU data was generated from different positions (shank, waist, thigh, chest, head, wrist, ankle). The performance of models trained on these synthetic datasets was compared, and the results were contrasted with models trained on experimental IMU data. The performance comparison was conducted based on accuracy using the dataset A.

Among the seven regions of interest, the models trained on IMU data from the shank and ankle exhibit the highest accuracy, regardless of whether the data is synthetic or experimental, with accuracy levels hovering around 90%.

IMU placement	synthetic	experimental
head	61.84%	66.83%
chest	70.70%	76.92%
waist	70.16%	79.09%
thigh	73.70%	76.92%
shank	**91.99%**	**87.50%**
ankle	**89.39%**	**89.66%**
wrist	65.59%	63.04%

IMU placement	accuracy
shank + tight	86.12%
shank + waist	**89.09%**
shank + chest	87.39%
thigh + waist	74.39%
thigh + chest	75.18%
waist + chest	69.35%

However, the models trained on data from other regions generally achieve accuracy rates ranging from 60% to 70%, with the highest not exceeding 80%. It is evident that in Dataset A, the IMU data from the shank region demonstrates notably superior discriminatory power between fall and non-fall actions compared to other positions. Similar testing was conducted on Dataset B, yielding consistent outcomes (TABLE VI).

E. Combination of Multiple IMUs

Combining the synthetic IMU data from multiple body locations within the same experimental trial, an LSTM model was trained and tested on Dataset B. For the scenario involving two IMUs, we considered four candidate body locations (chest, waist, thigh, shank) and explored all possible six combinations.

Among all combinations, the training data combination of waist and shank achieves the highest model performance, reaching 89.09%. In fact, models trained on data combinations that include the lower leg segment all attain high accuracy rates (exceeding 85%). Conversely, the accuracy rates of the three combinations lacking lower leg data fall below 80% (TABLE VII). In the case of the combination of lower leg and waist data, its accuracy is enhanced compared to using only lower leg or waist data individually.

V. DISCUSSION
A. Performance of Models Trained With Synthetic Data

The experimental results demonstrate that LSTM models trained using the synthetic IMU dataset generated by Opensim exhibit good performance on both public datasets, with

comparability to models trained using experimental IMU data of the same sample size. On dataset A, the model trained with synthetic data achieves higher accuracy compared to the model trained with experimental IMU data, while the opposite trend is observed on dataset B. The differences in performance among models trained with synthetic IMU data indicate the presence of variations, yet the disparities from models trained with experimental data remain within an acceptable range. Despite the discernible differences between synthetic IMU data and authentic IMU data at the signal level, the utilization of techniques such as human model scaling and randomized IMU pose offsets during the simulation process introduces data augmentation. Consequently, the occurrence of superior generalization performance to unknown subjects in models trained on the synthetic dataset, as compared to models trained solely on data from a single participant (as observed on dataset A), is also logically explicable. It's important to note that experimental IMU data collection is limited, while synthetic IMU data can be generated indefinitely. Based on this fact, this study further explores the impact of IMU placement positions on fall detection using arbitrarily generated synthetic IMU data. Different combinations of IMU data from various placements were also explored to enhance the accuracy of the model.

B. IMU Placement for Fall Detection

The results of testing LSTM models trained with both synthetic and experimental IMU data indicate that the fall detection model trained with data from the shank (lower leg) IMU achieves the highest accuracy. This holds true for both synthetic and experimental IMU datasets. Considering that the test dataset encompasses a wide range of actions, including common activities and highly confusing actions, and the model used is an effective LSTM model, these results shed lights on that placing the IMU on the shank region provides superior performance for detecting fall actions. However, it is important to note that when selecting the actual placement of IMUs in practical applications, consideration should also be given to the convenience of device wearing. For some elderly individuals, the ankle may not be an ideal placement for IMU devices.

C. Actions Susceptible to Misclassification

The testing results on dataset B reveal that the model trained using synthetic data correctly classifies the majority of both fall and non-fall actions. However, there are some actions that are particularly prone to misclassification, resulting in high false positive rates.

Among the 16 non-fall actions in dataset B, the actions of transitioning from standing to lying on a bed and from lying to sitting have the highest misclassification rates. These actions involve changes between standing/sitting and lying positions, which have similar movement patterns to fall actions, differing only in the intensity of the motion, making misclassification understandable.

Among the 20 fall actions, the actions of falling to the left side and then recovering and falling and quickly recovering have the highest misclassification rates. These actions involve

falling and recovering, and their movement patterns are similar to actions where the person stumbles but doesn't fall. Therefore, they are susceptible to being misclassified as non-fall actions. Correctly classifying these highly confusing actions might require the use of alternative models or feature extraction methods. The models trained with experimental IMU data also show high misclassification rates for these actions, which indirectly validates the reliability of synthetic IMU data.

D. The Combination of Multiple IMUs

Experimental results on Dataset B indicate that when using two IMUs, the best performance is achieved by combining the IMU data from the shank and waist. This performance also surpasses models trained solely on shank or waist IMU data. This observation suggests that the combination of IMU data from different body locations may enhance the representation of falling actions. The placement of an IMU at the waist captures the motion of the body's center of mass, providing insights into body posture and inclination. Moreover, such positioning usually minimizes user discomfort and interference with natural movements. It is noteworthy that IMU placement at the waist has been commonly adopted in a substantial number of fall detection studies [41]. Conversely, an IMU attached to the lower leg can track motion at the extremity of the body, aiding in detecting localized fall actions or gait anomalies. While utilizing multiple IMUs may enhance fall detection, it's important to consider the inconvenience of wearing numerous IMU devices in daily life. Additionally, this study's model is relatively simplistic and might not fully extract features from multiple IMUs. Exploring methods for the model to better capture the optimal representation of fall actions from multiple IMUs remains an area of investigation.

E. Quality of Synthetic Data

To assess the quality of synthetic IMU data generated by Opensim from a signal perspective, a comparison was made between synthetic IMU data and experimental IMU data collected during the same experiments (Fig.7). For accelerometer data, it was observed that experimental IMU data exhibited a significant amount of noise and spikes, while synthetic IMU accelerometer data appeared smoother. This difference in noise levels could be attributed to the use of Gaussian filtering during the action reconstruction process and the inherent filtering effect of Opensim's inverse kinematics calculation, acting akin to a human motion filter. Furthermore, the discrepancy between synthetic and experimental IMU data can be partially attributed to the fact that experimental IMUs can experience shifts on the body's surface due to clothing movement or muscle contractions, whereas simulated IMUs are anchored to specific body parts and do not undergo relative motion. Regarding synthetic IMU angular velocity data, it exhibited higher-frequency variations compared to accelerometer data. In fall actions, the deviation between synthesized and experimental IMU data is more pronounced due to reconstruction difficulties from self-occlusion, resulting in less smooth movements (Fig.8). Additionally, Opensim's
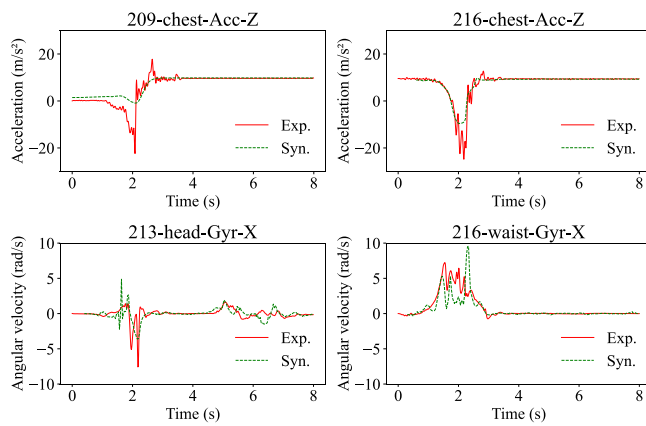
Fig. 8. Fall actions comparing experimental and synthetic sensor samples. The image caption and title convey the same meaning as Fig. 7, where 209, 216, and 213 respectively represent "fall backwards and lie on the floor," "roll from lying down on the bed to the floor," and "fall to the right and then recover.".

less effective synthesis of impact signals leads to smaller amplitude in synthesized IMU acceleration compared to actual falls. Nevertheless, despite these differences, synthesized data effectively tracks the changing trends of experimental data. This largely preserves the IMU signal patterns corresponding to different actions, providing interpretability for the trained neural network model to accurately distinguish between fall and non-fall scenarios. The quality of generated synthetic IMU data plays a pivotal role in training the model, and there is still room for improvement in the algorithm used to generate high-quality synthetic IMU data.

High-quality synthetic data could facilitate the extension of the experimental approach of this study to encompass a broader range of actions, particularly more intense movements like jumping. This expansion further enhances the applicability prospects of the proposed simulation method, such as in the realms of safeguarding children's physical activities [42] and facilitating the rehabilitation of foot drop patients [43].

F. Methodological Limitations

Based on our research, the methodology we proposed primarily exhibits limitations in two aspects. On one hand, the method requires videos recorded by multiple cameras to detect key points of the human body. Therefore, basic conditions for multi-camera recording, especially camera calibration and synchronization, are essential during action recording. It is crucial to position the cameras reasonably to ensure comprehensive recording of human motion from multiple perspectives. This constraint also implies that movements should be confined to a relatively small range. Additionally, common issues such as lighting conditions and occlusion need to be addressed. On the other hand, concerning the reconstructed human actions, due to the weak high-frequency signals of IMU acceleration data synthesized by Opensim (as detailed in Section E of the discussion), challenges may arise when studying transient changes in acceleration during motion. To a certain extent, this method is more suitable for exploring patterns and trends in the acceleration and angular velocity

changes during the action process. It can be anticipated that breakthroughs in these two aspects will facilitate the broader application of our methodology beyond fall detection.

G. Further Experimental Details

Two experimental considerations worth noting are the number of cameras and the selection of the simulated skeletal model. In our study, we opted for four cameras to fulfill the reconstruction requirements for fall actions. While theoretically at least two cameras are required to achieve this functionality, in practical situations, a single camera may not capture all key points of the human body entirely, primarily due to self-occlusion. Therefore, three or more cameras are needed to ensure the feasibility of reconstructing actions in the recorded videos. However, the increased number of cameras not only incurs additional costs but also adds complexity to the calibration and synchronization processes. By strategically planning the positions of each camera, we employed four cameras to meet the reconstruction requirements for fall actions.

In the synthesis of IMU data using the human body skeleton model in Opensim, we primarily utilized a standard human body skeleton model. Although we initially attempted to simulate different subjects by using various skeleton models, the identical execution of actions led to similar trends in the synthesized IMU acceleration and angular velocity signals, only varying in amplitude (attributed to model height and bone lengths). As a result, we opted for the approach of introducing random deviations in the placement poses for simulating IMU data as a means of data augmentation.

H. Comparison With Previous Methods

The method proposed in this paper introduces several innovative approaches compared to previous data augmentation methods. Firstly, our approach excels in synthesizing any required IMU data by reconstructing actions in a simulation environment. This unique capability enables the synthesis of IMU data for any body part in any pose, a feat unattainable with traditional data augmentation methods, such as rotation transformations applied to experimentally collected IMU data. For example, in practical experiments where the IMU is placed on the right leg of a subject, our method can synthesize IMU data for the left leg, even though it is not physically present but can be simulated in the environment. Secondly, in contrast to prior fall detection research that employed Opensim to generate simulation data, the IMU data we synthesize is more comprehensive, encompassing three-axis acceleration and three-axis angular velocity. Unlike previous studies, which typically generated only limited information, such as the acceleration of the model's center of mass, our method fully exploits the potential of simulation tools. Lastly, it is crucial to emphasize that the IMU data synthesized by our method adheres to the forward dynamics of human motion, providing a solid theoretical foundation. This synthesized data closely simulates IMU data obtained from experiments, enhancing the robustness and applicability of our proposed methodology.

I. Using Public Datasets

The principles behind vision-based action reconstruction and motion capture-based action reconstruction are fundamentally the same. Both methods employ least-squares algorithms to optimize the model's motion to best fit the simulated markers' positions to the actual markers' positions. There is a wealth of publicly available motion capture datasets, such as the CMU Motion Capture dataset, that can be used for this purpose. Utilizing these datasets involves performing inverse kinematics calculations.

From publicly available datasets, various common actions such as walking and running can be extracted. If custom actions are required, however, conducting action experiments is still necessary. Moreover, since most existing fall detection research relies on video imagery or wearable IMUs, a substantial number of fall action videos might potentially be employed to reconstruct falling actions. This approach could further reduce the need for extensive actual fall experiments and contribute to expanding the training dataset with various human motion patterns.

VI. Conclusion

In this study, a cross-modal data synthesis approach, from visual to IMU data, was employed. This method enabled the generation of a substantial volume of synthetic IMU data with only a limited number of actual fall experiments. Importantly, it was demonstrated that models trained on synthetic IMU data exhibited performance comparable to models trained on experimental datasets. This approach effectively reduces the count of actual fall experiments and mitigates the demands on experimental apparatus, thereby facilitating research in machine learning based fall detection algorithms. This study further investigates the optimization of fall detection based on simulation by exploring the placement of IMUs. The results indicate that the lower leg region is a recommended IMU attachment site, as models trained on data from this position demonstrate the best performance on the test dataset. Finally, the fusion of IMU data from different regions is discussed in terms of its impact on model performance. The findings suggest that when utilizing two IMUs, the placement configuration involving the lower leg and waist regions enhances the ability to distinguish fall actions from non-fall actions. This research provides novel venue for generating low cost synthetic data in scenario where acquiring data for machine learning is challenging and paves the way for customizing machine learning configurations.

Acknowledgment

References

[1] World Health Organization. *Falls*. Accessed: Aug. 18, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[2] P. Vallabh and R. Malekian, "Fall detection monitoring systems: A comprehensive review," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 6, pp. 1809–1833, Nov. 2018, doi: 10.1007/s12652-017-0592-3.

[3] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, "Latest research trends in fall detection and prevention using machine learning: A systematic review," *Sensors*, vol. 21, no. 15, p. 5134, Jul. 2021, doi: 10.3390/s21155134.

[4] Md. M. Islam et al., "Deep learning based systems developed for fall detection: A review," *IEEE Access*, vol. 8, pp. 166117–166137, 2020, doi: 10.1109/ACCESS.2020.3021943.

[5] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Budapest, Hungary, Jul. 2019, pp. 601–604, doi: 10.1109/TSP.2019.8768883.

[6] L. Liu, M. Popescu, M. Skubic, and M. Rantz, "An automatic fall detection framework using data fusion of Doppler radar and motion sensor network," in *Proc. 36th Annu. Int. IEEE Eng. Med. Biol. Soc. Conf.*, Chicago, IL, USA, Aug. 2014, pp. 5940–5943, doi: 10.1109/EMBC.2014.6944981.

[7] H. Jian and H. Chen, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine," *China Commun.*, vol. 12, no. 4, pp. 23–31, Apr. 2015, doi: 10.1109/CC.2015.7114066.

[8] J. Xu, Z. He, and Y. Zhang, "CNN-LSTM combined network for IoT enabled fall detection applications," *J. Phys., Conf.*, vol. 1267, no. 1, Jul. 2019, Art. no. 012044, doi: 10.1088/1742-6596/1267/1/012044.

[9] A. Lisowska, G. Wheeler, V. C. Inza, and I. Poole, "An evaluation of supervised, novelty-based and hybrid approaches to fall detection using silmee accelerometer data," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 402–408, doi: 10.1109/ICCVW.2015.60.

[10] P. Vallabh, R. Malekian, N. Ye, and D. C. Bogatinoska, "Fall detection using machine learning algorithms," in *Proc. 24th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Split, Croatia, Sep. 2016, pp. 1–9, doi: 10.1109/SOFTCOM.2016.7772142.

[11] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors*, vol. 17, no. 12, p. 198, Jan. 2017, doi: 10.3390/s17010198.

[12] F. Bagala et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS ONE*, vol. 7, no. 5, May 2012, Art. no. e37062, doi: 10.1371/journal.pone.0037062.

[13] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, "Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: A survey," in *Wireless Sensor Networks: Application-Centric Design*, Y. K. Tan, Ed. Rijeka, Croatia: InTech, 2010, doi: 10.5772/13802.

[14] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, Jun. 2014, doi: 10.3390/s140610691.

[15] A. Wertner, P. Czech, and V. Pammer-Schindler, "An open labelled dataset for mobile phone sensing based fall detection," in *Proc. 12th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Coimbra, Portugal, 2015, pp. 277–278, doi: 10.4108/eai.22-7-2015.2260160.

[16] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers Robot. AI*, vol. 7, p. 71, Jun. 2020, doi: 10.3389/frobt.2020.00071.

[17] L. Tong, Q. Song, Y. Ge, and M. Liu, "HMM-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1849–1856, May 2013, doi: 10.1109/JSEN.2013.2245231.

[18] B. T. Nukala et al., "A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network," in *Proc. IEEE Healthcare Innov. Conf. (HIC)*, Seattle, WA, USA, Oct. 2014, pp. 219–222, doi: 10.1109/HIC.2014.7038914.

[19] O. Ojetola, E. Gaura, and J. Brusey, "Data set for fall events and daily activities from inertial sensors," in *Proc. 6th ACM Multimedia Syst. Conf.*, Portland, Oregon, Mar. 2015, pp. 243–248, doi: 10.1145/2713168.2713198.

[20] L. Martínez-Villase nor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núnez-Martínez, and C. Penafort-Asturiano, "UP-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019, doi: 10.3390/s19091988.

[21] J. Yan, X. Wang, J. Shi, and S. Hu, "Skeleton-based fall detection with multiple inertial sensors using spatial-temporal graph convolutional networks," *Sensors*, vol. 23, no. 4, p. 2153, Feb. 2023, doi: 10.3390/s23042153.

[22] T. Theodoridis, V. Solachidis, N. Vretos, and P. Daras, "Human fall detection from acceleration measurements using a recurrent neural network," in *Precision Medicine Powered by pHealth and Connected Health*, vol. 66, N. Maglaveras, I. Chouvarda, and P. De Carvalho, Eds. Singapore: Springer, 2018, pp. 145–149, doi: 10.1007/978-981-10-7419-6_25.

[23] H. Yhdego et al., "Towards musculoskeletal simulation-aware fall injury mitigation: Transfer learning with deep CNN for fall detection," in *Proc. Spring Simul. Conf. (SpringSim)*, Tucson, AZ, USA, Apr. 2019, pp. 1–12, doi: 10.23919/SpringSim.2019.8732857.

[24] A. Butt, S. Narejo, M. R. Anjum, M. U. Yonus, M. Memon, and A. A. Samejo, "Fall detection using LSTM and transfer learning," *Wireless Pers. Commun.*, vol. 126, no. 2, pp. 1733–1750, Sep. 2022, doi: 10.1007/s11277-022-09819-3.

[25] G. Mastorakis, X. Hildenbrand, and K. Grand, "Customisable fall detection: A hybrid approach using physics based simulation and machine learning," *IEEE Trans. Biomed. Eng.*, vol. 54, pp. 1940–1950, 2007.

[26] E. Recinos, J. Abella, S. Riyaz, and E. Demircan, "Real-time vertical ground reaction force estimation in a unified simulation framework using inertial measurement unit sensors," *Robotics*, vol. 9, no. 4, p. 88, Oct. 2020, doi: 10.3390/robotics9040088.

[27] Y. Jiang, P. Malliaras, B. Chen, and D. Kulić, "Model-based data augmentation for user-independent fatigue estimation," *Comput. Biol. Med.*, vol. 137, Oct. 2021, Art. no. 104839, doi: 10.1016/j.compbiomed.2021.104839.

[28] L. Uhlenberg, A. Derungs, and O. Amft, "Co-simulation of human digital twins and wearable inertial sensors to analyse gait event estimation," *Frontiers Bioeng. Biotechnol.*, vol. 11, Apr. 2023, Art. no. 1104000, doi: 10.3389/fbioe.2023.1104000.

[29] J. P. Matos-Carvalho, S. D. Correia, and S. Tomic, "Sensitivity analysis of LSTM networks for fall detection wearable sensors," in *Proc. 6th Conf. Cloud Internet Things (CIoT)*, Lisbon, Portugal, Mar. 2023, pp. 112–118, doi: 10.1109/CIoT57267.2023.10084906.

[30] A. R. Inturi, V. M. Manikandan, and V. Garrapally, "A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network," *Arabian J. Sci. Eng.*, vol. 48, no. 2, pp. 1143–1155, Feb. 2023, doi: 10.1007/s13369-022-06684-x.

[31] M. Amsaprabhaa, "Multimodal spatiotemporal skeletal kinematic gait feature fusion for vision-based fall detection," *Exp. Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118681, doi: 10.1016/j.eswa.2022.118681.

[32] S. L. Delp et al., "OpenSim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 1940–1950, Nov. 2007, doi: 10.1109/TBME.2007.901024.

[33] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1302–1310, doi: 10.1109/CVPR.2017.143.

[34] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.

[35] D. Pagnon, M. Domalain, and L. Reveret, "Pose2Sim: An end-to-end workflow for 3D markerless sports kinematics—Part 1: Robustness," *Sensors*, vol. 21, no. 19, p. 6530, Sep. 2021, doi: 10.3390/s21196530.

[36] OpenSim. (8443). *How Inverse Kinematics Works*. Accessed: Aug. 18, 2023. [Online]. Available: https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+Inverse+Kinematics+Works

[37] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[38] C. Olah. (2015). *Understanding LSTM Networks*. Accessed: Aug. 18, 2023. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[39] Tongji University. *Intelligent Robotics and Computational Perception Lab*. Accessed: Aug. 18, 2023. [Online]. Available: https://robot.tongji.edu.cn/info/1297/2097.htm

[40] E. Boutellaa, O. Kerdjidj, and K. Ghanem, "Covariance matrix based fall detection from multiple wearable sensors," *J. Biomed. Informat.*, vol. 94, Jun. 2019, Art. no. 103189, doi: 10.1016/j.jbi.2019.103189.

[41] X. Chen, S. Jiang, and B. Lo, "Subject-independent slow fall detection with wearable sensors via deep learning," in *Proc. IEEE SENSORS*, Rotterdam, The Netherlands, Oct. 2020, pp. 1–4, doi: 10.1109/SENSORS47125.2020.9278625.

[42] P. Kang, J. Li, S. Jiang, and P. B. Shull, "A visual variability and visuo-tactile coordination inspired child adaptation mechanism for wearable age group recognition and activity recognition," *Adv. Intell. Syst.*, vol. 5, no. 1, Jan. 2023, Art. no. 2200236, doi: 10.1002/aisy.202200236.

[43] P. Kang, S. Jiang, P. B. Shull, and B. Lo, "Feasibility validation on healthy adults of a novel active vibrational sensing based ankle band for ankle flexion angle estimation," *IEEE Open J. Eng. Med. Biol.*, vol. 2, pp. 314–319, 2021, doi: 10.1109/OJEMB.2021.3130206.