Fall detection.

- synthetic motion generation -

Master thesis

ROB10, Hans Christian Østgård Larsen

Aalborg University
Electronics and IT

Here you can write something about which tools and software you have used for typesetting the document, running simulations and creating figures. If you do not know what to write, either leave this page blank or have a look at the colophon in some of your books.

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Fall detection, synthetic motion generation.

Theme:
Robotics

Project Period:
Spring semester 2023

Project Group:
ROB10

Participant(s):
Hans Christian Ø Larsen

Supervisor(s):
Thomas Moeslund

Copies: 1

Page Numbers: 40

Date of Completion:
June 6, 2023

Abstract:

This report analysis the state of datasets available for training classifiers for discriminating between ADLs and a human fall, with the conclusion that "real" human motion data, is not available for the fall class. However real fall data is available in the form of IMU data, but not visual data. The Human Diffusion Model is there modified to take IMU data as its input, this allows for human motions to be modeled from written text and IMU data, and therefore visual data can be simulated and used in classifier training.

Contents

Preface

I would like to thank Thomas Moeslund, and Eirik from the company ZinTouch.

Aalborg University, June 6, 2023

Hans Christian Østgård Larsen
hlarse16@student.aau.dk

Chapter 1

Introduction

The UN estimates that by the year 2030, those above the age of 60 will outnumber the number of children between 0 and 9. By the year 2050, people over 60 will outnumber those between 10 and 24, the population over 60 is expected to be more than 25 percent of the European population. [17]

Likewise, studies from the 1980s and 1990s have shown that one-third of the elderly fall every year. [6]

In order to allow for the above 60 to stay active, new technologies are being created to assist them.

The company ZinTouch has problems with elderly people falling without triggering their monitoring devices. This project focuses specifically on finding people post fall. This is because the ZinTouch company has problems with elderly people being left for a longer period of time, after a fall.

Once such solution could get a mobile platform for detecting the state of a elderly person, using ZinTouch's armband and a RGB camera on the mobile platform. Therefore in order to create a image classifier which can detect fallen elderly, a fall should be explored, with previous works in with IMU's and cameras. Likewise a solution should support the ZinTouch Armband which provides IMU readings, and distance measurements.

Chapter 2

Problem analysis

Initial the research is focused on exploring what a fall is, what data is available and how can this data be utilized in order to create a classifier.

2.1 Initial Research Question

- What are the risks associated with falls among the elderly? Where do they often fall, what does a fall involve? What data is available and how can this data be utilized?

This initial problem formulation can be broken down into pieces, such as:

- What is a fall?
- What environments causes falls?
- Do datasets exist of elderly people falling?
- What technologies exist to detect falls?
- What are the concerns associated with using tracking devices?

2.2 What is a fall?

Falls are generally divided into four stages [18, pg. 341]:

- Pre-fall phase.
- Critical phase.
- Post-fall phase.

- Recovery phase.

These four phases each represent a different level of stability. The Pre-fall phase is the phase where Activity's of Daily living occurs. This phase is where the person is walking normally and nothing outside the ordinary is happening. The Critical phase is when the person is mid fall, this phase starts when the person starts falling and stops when the person hits the floor, or other wise comes to a rest. The post-fall phase is when the person lies on the floor and awaits recovery or prepares for their own recovery. The recovery phase is then where the person either stands up themselves or they are being picked up by a helper [24, pg. 6892].

Noury et al. [18, pg. 341] also define times for the different stages of a fall. The Critical Phase being between 300 to 500ms, and the Post fall phase being below one hour. Another aspect of the fall is that they expect their to be a free fall period during the critical phase.

Another aspect of falling is known as the long-lie, where a person experience a extended time on the ground, which results in a shorter life expectancy [29].

2.3 Environment

A study from 2014 shows that the falls that people experience are primarily in the home. The study by Carmen A Pfortmueller et al. [20, Table 1] is a review of patients which have been admitted to their emergency room. They show that 35.2 percent of all falls happen in the home, while for those above 75, that number rises to 51.1 percent.

A study by Moreland et al. from 2015 expands on the survey above by including data from 66 hospitals in the USA. They, however, excluded data related to falls that were intentional. The Data collected from the hospitals included only data about the general localisation, such as indoor vs outdoor, or in the home vs in public, the authors have therefore used narrative descriptions to extract room localisation.

Moreland et al. [16] find that 68.2% of falls occur indoors. However, they analyzed the difference between women and men, and find that women fall indoors 71.6% of the time, compared to 61.7% for men [16, Table 1].

They likewise break down if the fall was at home or in public and find that 78.6 % of the falls for men where in the home, and 79.6 % of falls for women [16, Table 2].

Likewise, in their discussion, they talk about the most frequent fall localizations being the bedroom, stairs, and the bathroom [16, Pg. 596].

In order to expand on the dangers of the home environment, the environment can be broken down into specific rooms.

Roya Bamzar [3] conducted an experiment where she attempted to correlate perceived safety with actual safety, they found that apartments with a higher UD

score performed better in preventing falls. As part of their data gathering, they collected the number of falls and amount of time spent in a room[3, Fig. 2]. Their finding is that the most falls occur in the kitchen and the bedroom, with the living room and bathroom following after.

2.4 Previous work

Fall detection system comes in all varieties of sensor. Figure 2 from Singh et al. [24] divides sensor technology into two categories: wearable and ambiance. Wearable sensors are on-person sensors, while ambiance sensors are placed in the environment.

FFor wearable sensors, all studies use data that can be obtained by an IMU, which includes linear accelerometer data and rotational velocity data from the gyroscope.

Tables II and III from Singh et al. [24] likewise show that every approach using wearable sensors has an accuracy above 85%, with most studies lying above 90% accuracy

One notable study is an IMU-based detection system by Mao et al. [15], where they achieve an accuracy of 100% [24, Table II]. They use hand-tuned threshold values on the RMS values for linear acceleration and rotation. As a sub-conclusion, they stated that their solution works best when the sensor is placed on the waist.

While fall detection is an interesting area by itself, there's a very similar area of research where people attempt to classify different activities of daily living (ADL).

One such paper is the machine learning-based detector from Skovbjerg et al. [25], where they achieve an F-measure of 57%. Skovbjerg et al. [25] classifies ADLs by training an SVM to detect the class of the ADL.

Another more modern approach to ADL and fall classification is the approach taken by Shavit et al. [23]. They attempt to use a deep learning approach to the classification problem, where they train a transformer model to detect the class. Shavit et al. [23] achieved a 90.7% accuracy on their test datasets.

Following Figure 2 from Singh et al. [24], ambiance sensors can be broken down into all sensors mounted in the environment around the person.

Table IV from Singh et al. [24] contains papers with RGB camera-based solutions. One notable one is a detection system from Albawendi et al. [1]. They achieve an accuracy of 100% [24, Table IV].

2.5 Datasets

For synthetic datasets, Albawendi et al. [1] uses a dataset called ChangeDetectionNet (SDNET) [7] and various datasets created at the Le2i laboratories in France.

Obtaining datasets with people falling can be a rather difficult task, as many datasets are held by organizations and therefore not public. For example, the dataset used by Skovbjerg et al. [25] is a dataset created by AAU, with 20 healthy people and 25 people with brain injuries.

Another well-known dataset is the FARSEEING dataset [13]. However, like the dataset used by Skovbjerg et al. [25], the FARSEEING dataset is held by the University of Bologna, and it only contains falls collected in the 'real' world

One approach to datasets is to create them, as done in the study by Mao et al. [15]. However, while gathering data in the 'real' world is possible, monitoring the falls of elderly people brings with it its own problems. Creating a location for monitoring the elderly creates problems with costs and various ethics committees, as pointed out by Khan & Hoey [12]. Likewise, Bagala et al. [2] estimate that in order to capture 100 falls of the elderly, the elderly would need to be monitored for 100,000 days.

2.5.1 IMU's

In order to determine the placement of IMUs in existing datasets, Casilaria et al. [4] have provided a summary of the placements on the body. They summarize these localizations in Table 1 [4].

Table 1 [4] describes the placements as chest, waist, waist (belt), pendant, ankle, and wrist. The reason for the duplication of "waist" is that one refers to a sensor mounted on the subject's belt, while the other refers to a handheld device in the pocket, such as a phone.

Skovbjerg et al. [25] also use IMUs strapped to the user's thigh, and like [4], they use an IMU that is strapped close to the center of mass.

Shavit et al. [23] use a sensor placed in the user's pocket, in their case, a smartphone sensor.

The figure shown in Figure 2.1 provides an overview of the placements of IMUs on the human body. The numbers on the figure indicate the bones to which the IMUs are attached. Specifically, IMUs 20 and 21 correspond to the wrists, IMUs 1 and 2 correspond to the waist, IMUs 7 and 8 correspond to the ankles, and IMU 5 corresponds to the inner thigh. Note that the chest IMU is not shown in this figure, but it would be typically attached beneath IMU 12 where the arms are attached.

2.5.2 Synthetic datasets

When considering the availability of datasets, it is often the case that datasets with simulated falls are more readily available. However, synthetic datasets may not accurately represent real falls captured with IMUs.

In a study by Casilari et al. [4], an approach was taken to analytically compare synthetic datasets with real datasets. According to Casilari et al. [4, Page 4.],

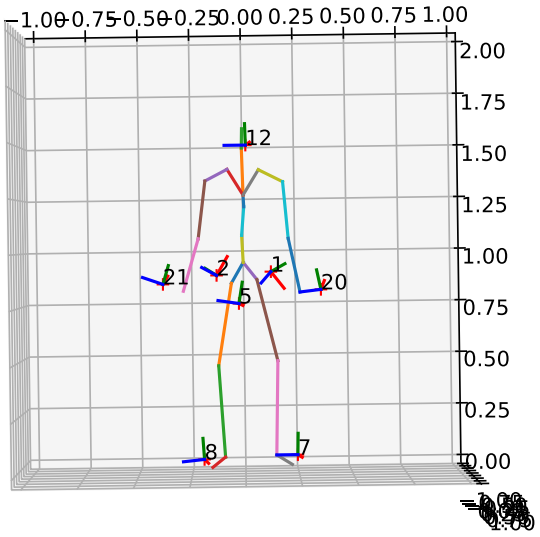

Figure 2.1: Overview of IMU Placements.

real falls may not necessarily have the same free fall phase as simulated falls, and simulated falls often experience higher impact forces. They describe slow falls as 'slumps'.

For their comparison, Casilari et al. [4] used two datasets with real falls. The FFFStudy dataset comprises veterans with multiple sclerosis, and the FARSEEING dataset comprises real-world falls with 300 falls gathered between 2012 and 2015.

In their study, Casilari et al. [4] used IMUs and derived features such as maximum and minimum acceleration from the IMU data. Casilari et al.'s conclusion is that falls generally do not experience forces as large as those in the synthetic datasets. They likewise theorize that the falls experienced by the elderly have features closer to those of ADLs as shown in synthetic datasets.

2.6 Discussion

When discussing the detection algorithms above, it is notable that older algorithms claim an accuracy of 100%, while the more recent study by Skovbjerg et al. [25] only claims an F-measure of 57%. While accuracy and F-measure may not be directly comparable, the fact that the newer algorithm does not achieve an F-measure of 100% indicates that the algorithm is either flawed or the data differs significantly between the different classifiers.

It should be noted that the paper by Skovbjerg et al. [25] attempts to classify different daily activities and is therefore not directly comparable to the other algorithms.

In subsection 2.5, many of the different ways in which these algorithms are

validated are briefly examined. It should be clear from this section that the algorithms do not use a consistent or even comparable dataset, leading to claims such as achieving an accuracy of 100%

Likewise, when exploring the respective papers of the algorithms, it becomes clear that many datasets and algorithms are evaluated using healthy and able-bodied people, presumably students and other personnel found in universities.

Similarly, the older definition of a fall given in section 2.2 attempts to create a definition that is not data-driven. The 300-500ms time of the fall is created as a starting point for defining what a fall is.

The reason for including ADL classifiers is that their accuracy is lower than that of fall detectors, demonstrating that classifying different ADLs may be a harder task than simply identifying a fall.

2.7 Conclusion

When looking at the above classifiers and datasets, it should be clear, that many datasets for fall detection, are either private or requires permission from the authors.

Final problem formulation

When examining the classifiers and datasets discussed above, it becomes clear that many fall detection datasets are either private or require permission from the authors. Another consideration is that IMU data is available for falls, but not necessarily visual data. One approach to training a classifier could therefore be to generate synthetic data. Approaches to generate generating synthetic data, will therefore be explored, in the coming chapter. The research question will be formulated as follows:

Is it possible to generate synthetic data for training classifiers for fall detection?

Are there existing simulation methods for this purpose?

Chapter 3

Synthetic dataset generation

Section 2.5 in the PA talks about synthetic that are generated with real people, who are not the target group for the technology. In this chapter synthetic means data generated from simulated environments, such as NVIDIA's issac sim or a game engine.

A recent development in the area of classifying ADLs' is the ElderSim by Hwang et al. [10]. The ElderSim is a unreal engine 4 project for simulating video clips of elderly people conducting ADLs', it uses pre-recorded motions clips in combination with 4 different virtual environments.

3.0.1 Omniverse Replicator

Recently, NVIDIA has introduced the Omniverse Replicator. The replicator uses NVIDIA's Isaac simulator and allows for machine learning models to be pretrained on simulated data. NVIDIA does not publish numbers related to machine learning on their site, they demonstrate a pipeline for training a image classifier on simulated images. [19]

3.0.2 ElderSim datasets

ElderSim uses four different datasets to both validate and create the synthetic data. This section will go over the datasets and their relevant features, in context of setting and age group. The classes used by the ElderSim team can be viewed in Table 1 [10].

ETRI-Activity3D

ElderSim is a continuation of the ETRI-Activity3D dataset (ETRI), the ETRI dataset consists of 55 action classes with 52 daily activities with 3 activities being related to human-robot interaction. The authors used 100 participants for these activities,

50 of those being elderly persons and 50 of them being young adults. They also vary the distance and perspective in order to simulate human-robot interactions. In all they have 112620 samples, in which they have captured and annotated the skeletons for every sample [11].

They conclude that the actions performed by the elderly and those of healthy adults differ significantly, they show this significance in Table II [11].

NTU dataset

The NTU dataset consists of 60 ADL classes divided into three groups: 40 daily actions, 9 health-related actions, and 11 mutual actions. It includes data from 40 participants aged between 10 and 35. The dataset is collected at the Rapid-Rich Object Search Lab at Nanyang Technological University in Singapore, and only contains samples collected in a laboratory setting [22]. Similar to the ETRI datasets, this one is also collected with the Microsoft Kinect V2 sensor.

Toyota Smarthome

The Toyota datasets is a dataset of 18 elderly individuals which performs 31 kinds of ADL's in a test apartment. The dataset main feature is that the ADL performance is unprompted, and therefore not done on command. The Toyota dataset therefore also contains very unbalanced data and varied duration of these tasks. This dataset like the ETRI dataset is collected in an apartment setting, however unlike the ETRI, this is done with cameras mounted on the ceiling pointed down towards the participants. Unlike the previous two datasets, the Toyota dataset is collected with a Microsoft Kinect v1 sensors [5].

3.0.3 ElderSim

Figure 1 in [10], shows the pipeline that they propose for generating visual samples. They randomize the lighting, viewpoint and character, which allows for them to generate synthetic data. Their generated data is release as the KIST SynADL dataset [10, Page 2].

Table 3 to 10 demonstrates their results of augmenting real data with synthetic data. Section IV subsection 4 [10] is their Cross-Age split, which they train the classifier on only healthy young subjects. Table 6 shows how the algorithm performs on the test part of the ETRI dataset. ETRI_E is the elderly split, and ETRI_Y is the young split of the dataset. Especially the second row is interesting as they train on the young split and validate on the elderly split. Here they show a 2%[10] improvement in accuracy when training with the normal dataset and their simulated one.

Likewise Table 6[10], shows that even if there is a significant difference between elderly performing ADLs' and younger people performing them, training entirely on the younger split does provide a higher accuracy on the elderly split, then the other way around. This is also where the testing of the ElderSim data is flawed, as the numbers in Table 6, are not directly comparable, while it looks like its better to train on younger data then data from the elderly, their test data is not consists between the numbers and they are therefore not comparable.

Their Table 7 and 8[10] is a cross dataset trial, where they examine the performance gains, of training on their synthetic dataset in comparison of validation across datasets.

On Table 7[10] they demonstrate that performance can be gained on the elderly splits from before, when training a algorithm on other datasets and their synthetic data. They gain anywhere from 2% to 16% in performance when training on the synthetic data.

Table 8 [10] describes their algorithm when trained on the NTU dataset and validated on either the TOYS or LIVA dataset. They show that the RGB image approach gains the most when trained with a simulated dataset with varying lighting, which indicates that theres a lack of diverse lighting data in the NTU dataset [10].

Discussion

One interesting note is that the skeletal approach does not improve significantly [10, Table 7] with the simulated skeleton data as training data, which is interesting as the simulated skeleton data, from the ETRI dataset, but reprocessed from different viewing angles.

The conclusion is therefore, that when using RGB, there's a significant amount of accuracy to that can be achieved, by including simulated data in real world data.

While the ElderSim includes a class for fallen, their general approach does not contain many categories with activities on the floor, such as lying down or sleeping, as a result, they classify all people lying down as fallen

In the final problem formulation the question is asked: Is it possible to generate synthetic data for training classifiers for fall detection? and by looking at the results from ElderSim, it seems possible to generate data synthetic data for training a ADL classifier, and while ElderSim does have a fall in their dataset, they do not differentiate between falls and lower activities.

Therefore, motion data for real falls and lower ADLs needs to be explored. Although there do exist datasets with real IMU data, there are no datasets available with 'real' skeletal data for elderly falling. As mentioned in section 2.5, collecting such datasets is outside the scope and time frame of this project. However, the project will explore generating skeletal data for falls and comparing the generated data with the data from real falls and lower ADLs

Chapter 4

Motion generation

An interesting area for dataset generation, is the area of motion synthesis. Using motion synthesis ElderSim and other simulation techniques like it, could be extended with new classes of ADLs' and motions, such as falls.

Datasets of motion

The area of motion synthesis appears to have begun in 2016 with the creation of the KIT dataset [21]. The KIT dataset is a large scale dataset which combines other motion-captured motions with crowd sourced annotations. KIT defines a skeletal structure that allows for the motions to be uniformly represented, independent of the camera setup. Their skeletal structure framework is called the Master Motor Map, which defines how to convert to and from their skeletal structure [21]

In 2019 the AMASS dataset was released, its an extension of the KIT datasets. AMASS aims to make a large-scale dataset available for machine learning tasks. AMASS does this by combining multiple smaller datasets, into one larger datasets, and like with the KIT datasets, it also redefines the skeleton structure. To process motion captures into the SMPL skeletal structure, AMASS extends a library called MoSh, which allows for them to process motion captures into the SMPL skeletal structure. [14]

In 2022 Guo et al. [9] introduced the HumanML3D datasets, this dataset is different from the KIT dataset, in the way that its a combination of several unlabeled datasets. Guo et al. combines 5 unlabeled datasets, resamples them to be in the skeletal form of the SMPL skeleton and then provides labels for them. additionally they scale the sampling frequency to be 20Hz and the maximum snippet length to be 10 seconds. [9, Page 5156]

4.0.1 Motion generation

When it comes to motion generation, there have been several approaches over time, such as the Action2Motion paper in 2020 [8], which was followed by Generating Diverse and Natural 3D Human Motion from Text [9] in 2022, and a variation in 2023 called Human Motion Diffusion [27].

Action2Motion

The Action2Motion papers [8] was the first paper to introduce the parameterized skeletal model, into a deep learned architecture. Specifically they take the joint parameters and train on those instead of the raw localizations of the joints. According to the authors, this approach keeps the joint lengths constant and avoids the issue of limb stretching that previous approaches faced. They likewise utilize a Variational Auto-Encoder(VAE) to model the possible joint motion, while utilizing a RNN model to model the time at the current timestamp.

The Action2Motion paper likewise takes a concept from computer vision called Frechet Inception Distance. The Frechet Inception Distance consists of taking a trained model, running the samples on it, and extracting features from an internal layer of the model. The trained model is run over the entire sample space, and the output activations are modeled using a normal distrubution. Once a normal distrubution is obtained, "for both the original dataset and the synthesized dataset, the Frechet Inception Distance formular is used to calculate the distance between the pair of normal distrubution. The calculate distance is then called the FID score. [8, Page 6]

Generating Diverse and Natural 3D Human Motion from Text

In their paper, Guo et al. [9] add several features to motion generation, features such as positional encoding and the HumanML3D dataset. They also expanded upon the VAE encoder method by training an additional model, which predicts the length of a motion. This approach provides a better representation of when to stop sampling from their VAE encoder by predicting the length of a motion.

Human Motion Diffusion

An interesting development in the area of motion generation is a paper from Tevet et al. [27]. Tevet et al. [27] demonstrate a transformer model for generate human like motion from text. They is archived by encoding a text prompt to a latent space, this latent space is then combined with the time encoding step from the paper by Guo et al. [9] They also introduce the concept of using a diffusion model for denoising motions steps. [27]

4.0.2 Discussion

One interesting aspect of the Guo et al. paper [9] is that it does not draw a direct comparison to the Action2Motion paper, even though it does use its evaluation methods, and references it.

Chapter 5

Proposed system

When looking at the above sections, it should be clear that real falls does not exist as visual data, and only as IMU data.

However many ADL's use similar movement structures to falls and synthesizing human motion is possible though newer machine learning systems.

I therefore propose a pipeline, where a motion system is trained on synthetic falls and then feed real imu fall data, in order to generate the skeletal structure of a real fall.

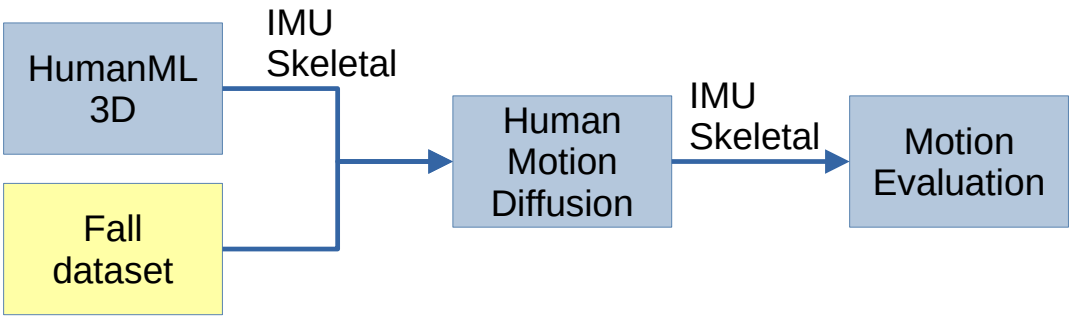

Figure 5.1: Overview of training MDM.

Once the MDM is trained the real IMU data from real falls can be introduced. In Figure 5.2, a dataset can be created by combining synthetic data from ElderSim with real ADL data from an ADL dataset.

5.1 Delimitation

While Completing the entire pipeline shown in Figure 5.2 would be optimal for finishing this problem, in this report, we will focus on modifying MDM to take IMU data, and expanding upon the HumanML3D dataset with synthetic falls.

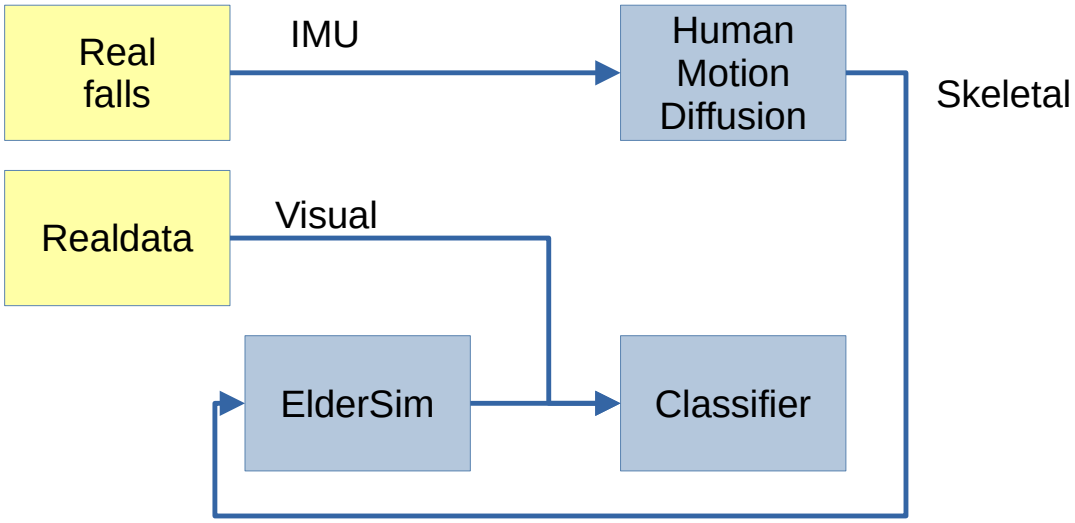

Figure 5.2: Overview of training the classifier.

Likewise while the ETRI and Faarseing datasets would have been perfect candidates for evaluating and training the MDM algorithm, the applications for access to those datasets is currently pending approval.

Instead we're using HumanML3D and some annotated synthetic falls, and as found by the people at ETRI our trained model therefore only knows what movement looks like from young adults. 3.0.2

Chapter 6

Implementation

The first part of the implementation is to prepare the dataset, the HumanML3D dataset requires some setup and download of all the datasets that it requires.

Afterwards the modifications are made to the Human Diffusion Model, which allows for it to take IMU data, as an input.

6.1 HumanML3D

The HumanML3D dataset [9] is a combination of several other datasets, specifically it needs the once listed on table 6.1:

Once these datasets are downloaded and ready, they need to be processed by the HumanML3D scripts. Firstly all the motions are resampled to be at 20FPS, as many of them are captured at a standard 30 and 60FPS.

When the retargeting is done, the HumanML3D goes onto rescaling the datasets. The rescaling happens, based on the size of a example figure. This is such that all movements are conducted by a figure of the same scale. The HumanML3D is likewise cut down to 120 timesteps for each sample.

Once these two steps are completed, the HumanML3D dataset is ready for use. This process takes 30+ hours.

6.1.1 Exploring HumanML3D

To determine if the HumanML3D dataset contains enough ADL-related data for modeling falls and lower activity ADLs accurately, a word search was conducted on the texts. Words commonly associated with lower ADLs and falls were included in the search. Figure 6.1 displays a bar chart showing the frequency of the words 'sleep,' 'yoga,' 'lie,' 'fall,' and 'lying' in the dataset. While these words are not significant by themselves, they indicate a general trend in the dataset. Specifically, more data related to falling and lower activity ADLs might need to be added to

ACCD (ACCD)
HDM05 (MPI_HDM05)
TCDHands (TCD_handMocap)
SFU (SFU)
BMLmovi (BMLmovi)
CMU (CMU)
Mosh (MPI_mosh)
EKUT (EKUT)
KIT (KIT)
Eyes_Janpan_Dataset (Eyes_Janpan_Dataset)
BMLhandball (BMLhandball)F
Transitions (Transitions_mocap)
PosePrior (MPI_Limits)
HumanEva (HumanEva)
SSM (SSM_synced)
DFaust (DFaust_67)
TotalCapture (TotalCapture)
BMLrub (BioMotionLab_NTroje)

Table 6.1: Table over the included datasets in the HumanML3D set

the dataset to improve the MDM model's performance. In Figure 6.1, the words are illustrated as a percentage of the whole dataset. For example, the word 'yoga' appears 36 times in the dataset and represents about 0.2% of the samples, while falls make up more than 1% of the dataset.

Another note is that the word walker does not exist in the HumanML3D dataset, this, along with the source of the datasets included in HumanML3D, suggest that the HumanML3D does not model elderly people well, especially the long drawn falls. The word elder likewise only appears 4 times in the annotated data.

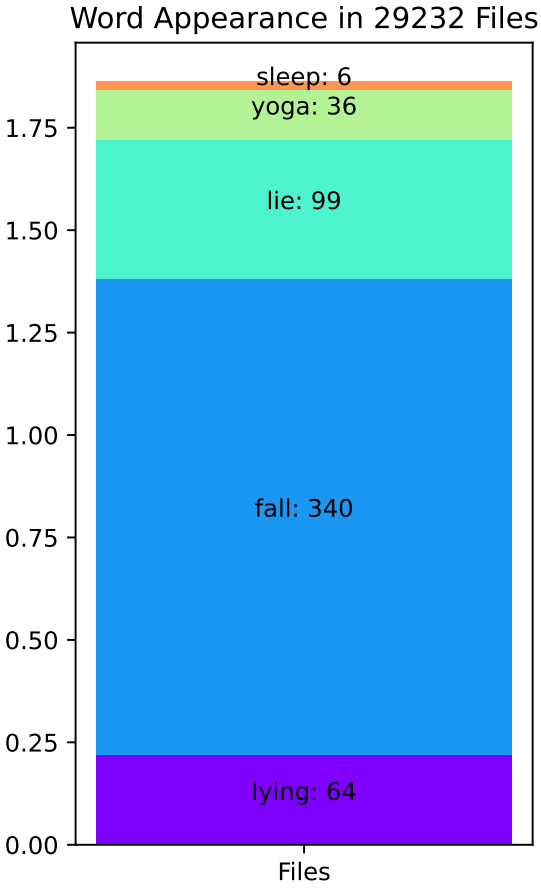

Figure 6.1: Words in the HumanML3D dataset.

6.1.2 Simulated IMU's

In order to get IMU data from the movements of the HumanML3D dataset, a transform can be calculated to the positions described in section IMU's 2.5.1. In order to calculate the transform to each IMU placement, the kinematic chain of the body needs to be followed.

Figure 6.2 details the bone index the bones in the SMPL framework, Figure 6.3 details the placement of the IMU's and the bones which they are attached to.

The SMPL framework for the main body, consist of 21 bones, with a base coordinate system in the tail. In order to get to the transformation matrix of bone 8, the transformation matrixes for bone 2 and 5 first needs to be calculated. The kinematic chain is provided by the SMPL framework, and is as follows $< 0, 2, 5, 8, 11 >$. Likewise the bones do not follow a standard convention and need addition offset parameters, in order to calculated the end position, the SMPL framework likewise provides these.

For formula is simply the standard transformation formula, but with offsets

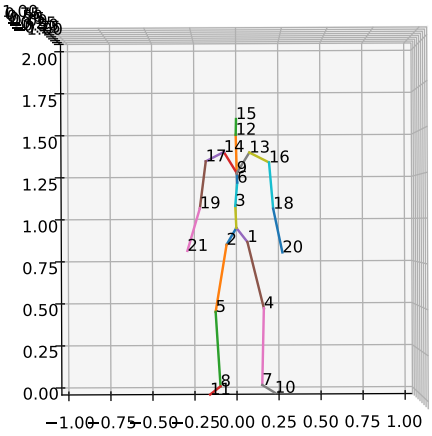

Figure 6.2: Overview of skeleton numbers.

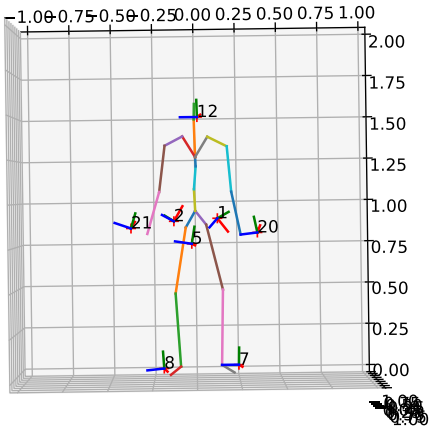

Figure 6.3: IMU Placement.

Bone	Offset	Rotation
7	0, 0, -0.1	0
8	0, 0, 0.1	0
2	0, 0, 0.1	0
1	0, 0.1, 0	0
21	0, 0, 0.1	0
20	0, 0, -0.1	0
12	0.1, 0, 0	0
5	-0.1, 0, 0	0

Table 6.2: Offsets for the IMU's

and bone lengths. Formula: $T^8 = T^0 * \prod_{i=a}^{2,5,8} T^i$ with T^i being $T^i = \begin{pmatrix} R_i & offset_i * length_i \\ 0\ 0 & 0 \quad 1 \end{pmatrix}$

R_i is the current rotation matrix of the bone, and the $offet_i$ is the offset provided by the SMPL framework, and $legnth_i$ being the length of the bone.

Once the transformation matrix of the bone is found, it needs to be propagated in time t, the expression therefore needs to be extended such that it can be done for every timestep, extending the formula: $T_t^8 = T_t^0 * \prod_{i=a}^{2,5,8} T_t^i$. When extending with time only the rotation matrix is timestep dependent, the offset and bone length are not.

The offsets for the IMU's need to be defined, such that the IMU's stay in the same place, relative to the bone, for every timestep.

Table 6.2 shows the offsets used in order to create the transforms for the IMU's, the rotation is 0 as rotation has not been applied.

Once the transform for each IMU is created, the linear acceleration and rotation velocities needs to be calculated. The linear velocity is calculated by taking the difference between the translation at time t and $t-1$ then dividing by the framerate

of 20FPS. The acceleration is then calculated by taking the difference between the velocity at step t and $t-1$, and dividing by the framerate which gives the linear acceleration, as the acceleration is calculated in

$$velocity[t] = \frac{pos[t] - pos[t-1]}{\Delta t} \tag{6.1}$$

$$acceleration[t] = \frac{velocity[t] - velocity[t-1]}{\Delta t} + [0.0, 0.0, -9.82] \tag{6.2}$$

Equation 6.2 and Equation 6.1 are used for all IMU's in order to generate their respective acceleration, one note because it needs a velocity in order to calculate the acceleration, timestep 0 and 1 are going to be wrong for the acceleration.

Additionally the acceleration needs to be transformed into frame of the last IMU, such that the IMU readings appear with the correct rotation, this is done by multiplying the acceleration with the inverse transformation matrix.

$$acceleration[t] = T_t'^{imu_8} * acceleration[t] \tag{6.3}$$

Once these steps are completed, they are repeated for all IMU's.

For the rotatinal velocity equation 6.4 is used, again for every IMU.

$$velocity[t] = \frac{rotation[t] - rotation[t-1]}{\Delta t} \tag{6.4}$$

Once completed it should generate a matrix of the size [timesteps, 8, 6] corresponding to the 8 IMU's and the 6 variables that describe localization and rotation.

6.2 Human Motion Diffusion

The Human Motion Diffusion (MDM) model introduces two new technologies for motion generation: the diffusion method and the transformer. This paper will first describe these technologies, followed by an explanation of the proposed model.

6.2.1 Technologies Used in MDM

In order to modify the MDM route to accept IMU data, the transformer and diffusion architectures used by it, first needs to be understood. In this section, we will therefore research the fundamentals behind the diffusion and transformer techniques to prepare for modifying MDM later.

Transformer

The Transformer was introduced by the paper "Attention is All You Need" by Vaswani et al., in 2017. [28] The transformer model is mainly composed of an encoder and a decoder, each made up of multiple identical layers.

Figure 1 from their paper [28] illustrates the transformer architecture, with the left part being the encoder and the right part being the decoder. One of the main features of the transform is the self attention, which is the core innovation of the transformer. It enables the model to weigh and incorporate information from different positions of the sequence into its representation of a single element in that sequence. Specifically in MDM, the input embeddings are the output of the MLP layer connected to CLIP on figure 6.4. CLIP being a pretrained word encoder. The transformer scales well as transformer are designed to handle sequences of data, such as time-series data or natural language, and they're especially known for their effectiveness in natural language processing tasks. Unlike models like RNNs or LSTMs, transformer do not process the sequence data in a linear manner, which allows them to efficiently handle longer sequences, with MDM the transformer will be used with the previous joint positions. The transformer will be used with the previous joint positions as input and will output new joint positions.

Transformer have significantly pushed the performance boundary in a variety of tasks, leading to substantial improvements in the state-of-the-art. [28]

Diffusion

In machine learning diffusion is the process of learning the gradient instead of the data itself, by taking the data, adding gaussian noise to it. It is believed that the model can learn the inverse of the noise and generate a gradient which lead the data back to the original form. [26]

$$q(x_t^{1:N}|x_{t-1}^{1:N}) = \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1}^{1:N}, (1-\alpha_t)I\right) \tag{6.5}$$

Tevet et la. [27] uses equation 6.5 in order to add gaussian noise to a sample, t in this case is the amount of noise steps that have been added, and α_t is a hyper parameter for tunning the strength of the noise.

The likewise use a sampler that allows for them to predict the signal itself. [27]

6.2.2 Modifying MDM

The Human Motion Diffusion models consist of two steps, firstly it generates 512 latent vectors, it adds a timestamp to this vector, and then runs a diffusion step for 1000 samples. [27]

Figure 6.4 demonstrates the initial initialization of the MDM model, in this state it takes in a piece of text and uses that text to generate a vector of 512 values,

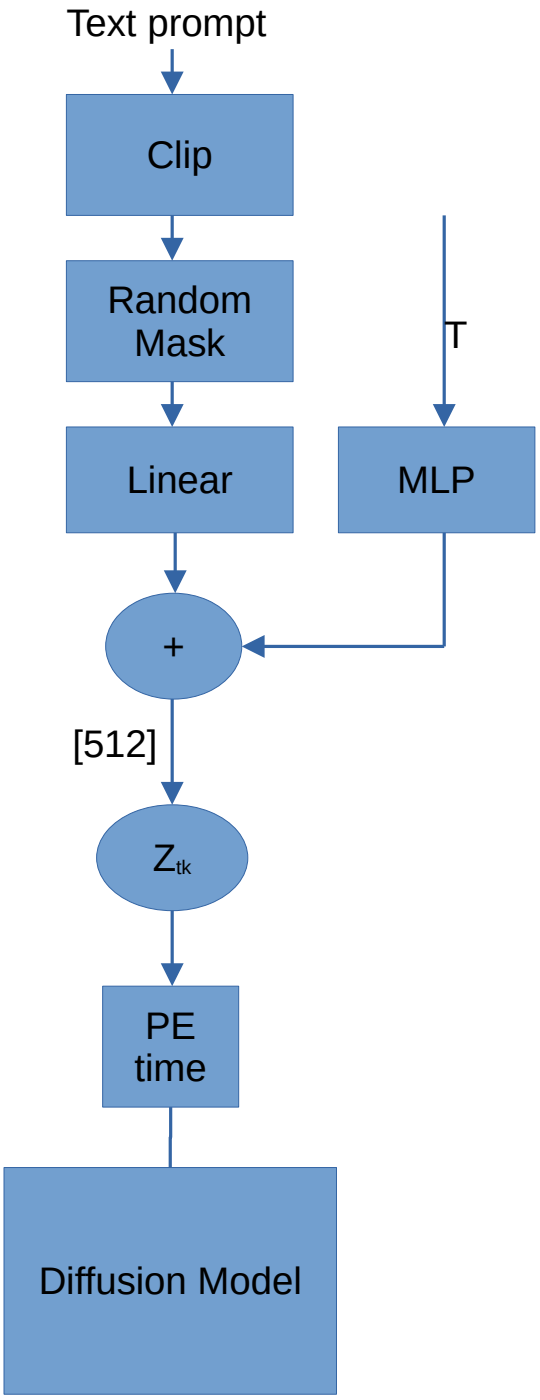

Figure 6.4: The Human Diffusion Model

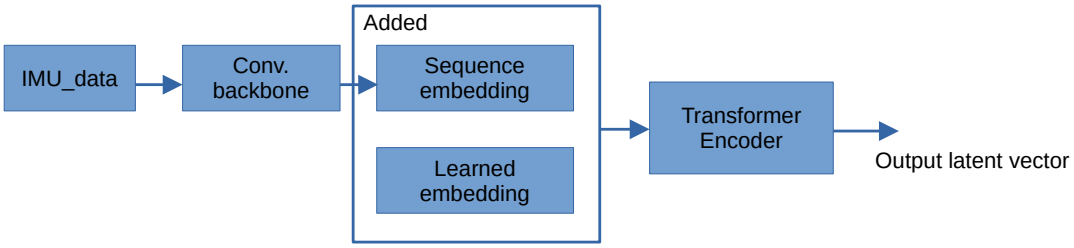

Figure 6.5: HAR classification model.

these values are used to generate a latent vector, which is then used as input for the diffusion step and describes the motion. In the paper by Tevet et al. [27] they describe how these 512 values can be generated from anything, the model is therefore modified, that these 512 values can be generate from IMU data. Tevet et al. [27] call these 512 values a latent vector.

When modifying the model of Tevet et al. [27], a proven model from the previous work section 2.4 can be used. The model by Shavit et al. [23], has proven itself capable of detection the motion and classifying it, it is therefore expected that the same classification can be used when initialization the diffusion model with the latent vector. The HAR models is shown on figure 6.5. Unlike the MDM model, the HAR model learns the position embedding directly from the data, Therefore, it is debatable whether the positional embeddings from figure 6.4 should replace the Learning Embedding in figure 6.5, since the HAR model learns the position embedding directly from the data.

The proposed modified architecture can be seen on figure 6.6, here the output from the HAR classifier is added, directly onto the latent vector generated from the text.

Another possibility is to simply remove the text latent generation at all, this will also tell us if its possible to extract motion from IMU data in general.

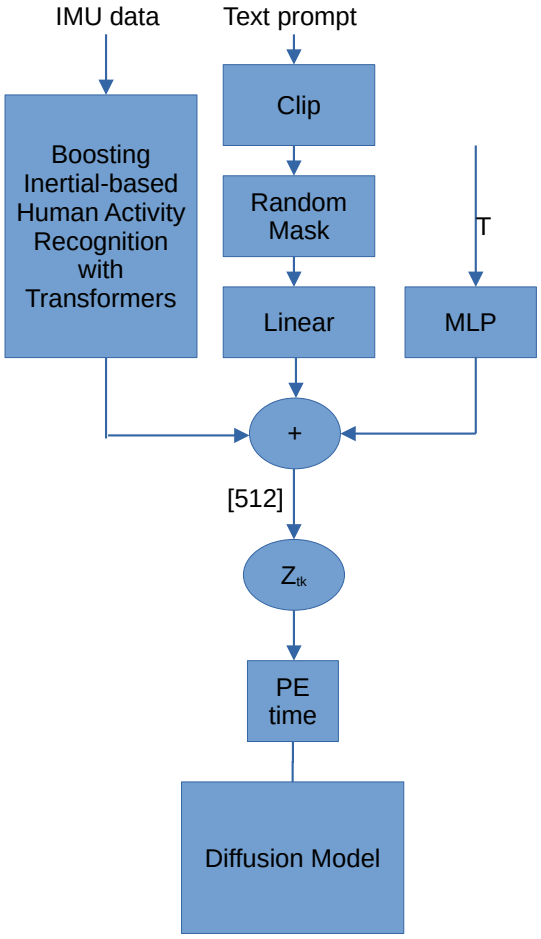

Figure 6.6: Modified MDM architecture, which takes IMU data in order to generate its latent vector.

Chapter 7

Training the model

When training the MDM paper by Tevet et al. [27] proposes a batch size of 64, a latent dimension of 512 and between 750,000 and 2 million training steps. Similarly the project has 8 IMU's available for test, and the model only takes a single IMU as input, thus, the model can be trained separately for each of the 8 available IMUs, using one IMU as input per training run. However, training the model takes approximately 3 days, and with the time constraint on this project, only 3 different locations for the IMU's have been trained. Specifically, the model has been trained for three different IMU locations: the wrist, the waist, and an ankle.

Tevet et al. [27] introduces modifications to 3 losses, the position loss, the loss for distance between the foot and the floor, and a loss for the velocity of the joints.

$$L_{\text{pos}} = \frac{1}{N}\sum_{i=1}^{N}\left\|FK(x_0^i) - FK(\hat{x}_0^i)\right\|_2^2 \tag{7.1}$$

$$L_{\text{foot}} = \frac{1}{N-1}\sum_{i=1}^{N-1}\|(FK(\hat{x}_{i0+1}) - FK(\hat{x}_{i0}))\cdot f_i\|_2^2 \tag{7.2}$$

$$L_{\text{vel}} = \frac{1}{N-1}\sum_{i=1}^{N-1}\left\|(x_0^{i+1} - x_0^i) - (\hat{x}_0^{i+1} - \hat{x}_0^i)\right\|_2^2 \tag{7.3}$$

Equation 7.1, 7.2 and 7.3 are from the paper by Tevet et al. [27]. They combine these losses into the total loss called Loss/loss on Figure 7.1. Specifically these allow for them to avoid foot sliding, with equation 7.2, and model proper human motion velocities with loss 7.3.

Discussion

As loss 7.3 operates on velocities and therefore force the model to move 'realistically', it is hoped that this approach results in a realistic model of acceleration, in addition to realistic movement.

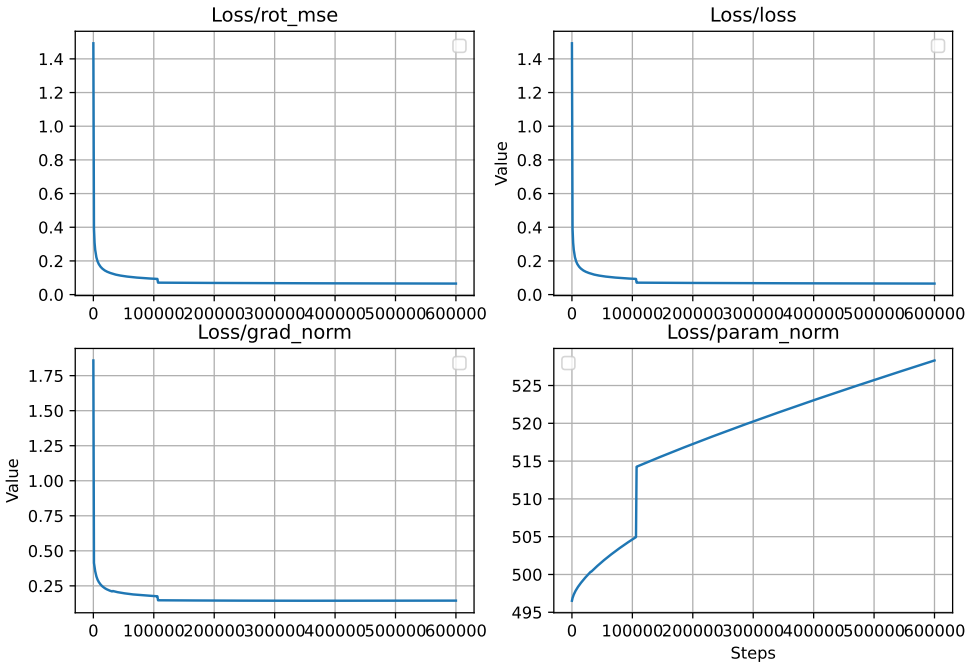

Figure 7.1: Training losses for the IMU on the waist.

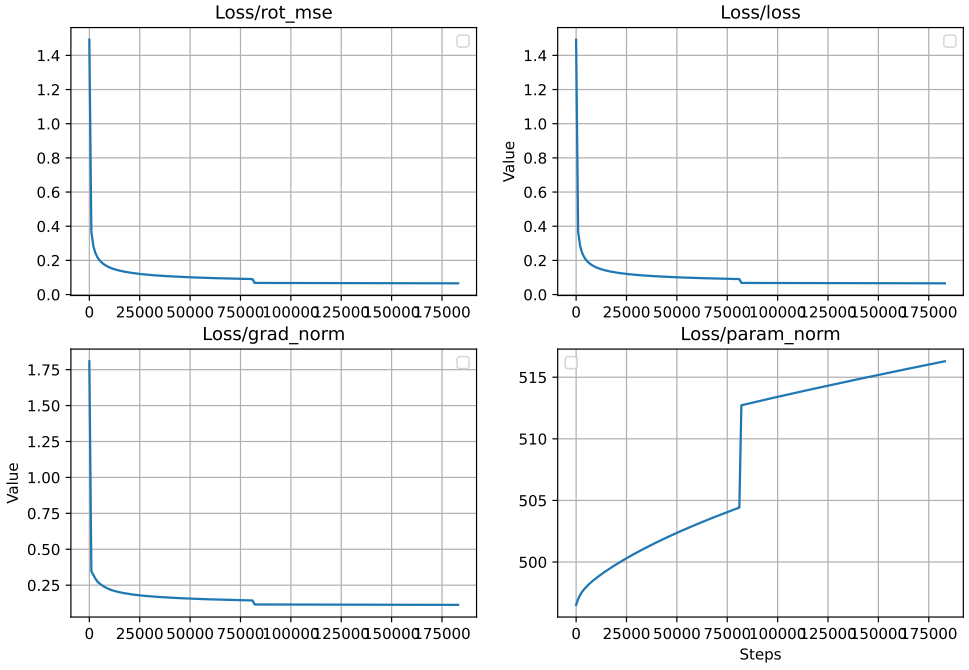

Figure 7.2: Training losses for the IMU on the ankle.

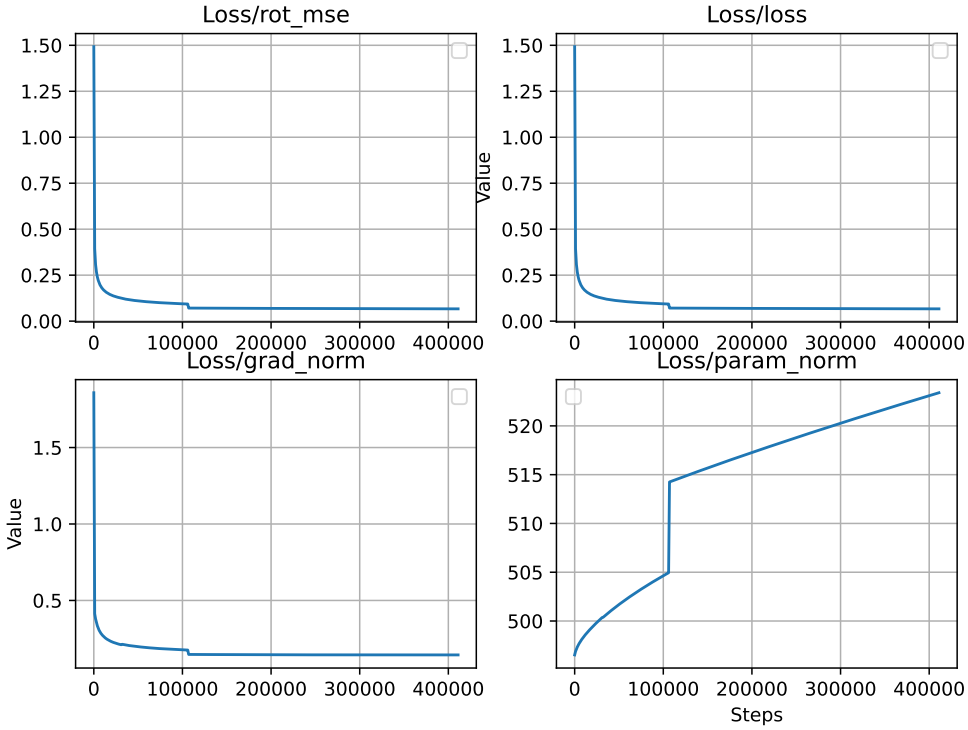

Figure 7.3: Training losses for the IMU on the wrist.

On Figures 7.1, 7.2 and 7.3 it can be seen that after about 100K steps the model stops decreasing in loss, and thus, it can be inferred that the model has reached optimization after approximately 100K steps. The rot_mse which stands for root mean square error loss, and is tied to the 0'th joint of the model, representing the model's predictions for the physical locations of the skeleton.

When using the diffusion model and learning the gradient of the data instead of the data itself, the grad_norm is a loss, which indicates if the models has learned the required data distribution. Similar to the error loss, the grad_norm loss indicates that the model has successfully learned to model the data.

The jump at around 100K steps is where the model was saved and restarted, it occurred with a reload of the model. Later the waist model was trained for another 1M iterations, the loss did not decrease and the param_norm loss kept going up indefinitely.

Chapter 8

Testing

When testing the parameters from the paper by Tevet et al. [27] can help determine whether the model is capable of accurately modeling human motion. Looking at table 8.1 it can be seen that the results of the model with the IMU around the waist is significantly worse than that of the dataset. It however archives a better FID score then the original model.

Method	R Precision ↑	FID ↓	MultiModality ↑	Diversity →
Real	$0.798^{0.001}$	$0.002^{0.000}$	$-.--$	$9.481^{0.050}$
MDM encoder	$0.608^{0.005}$	$0.767^{0.085}$	$2.927^{0.125}$	$9.176^{0.070}$
IMU ankle	$0.619^{0.006}$	$0.520^{0.044}$	$5.525^{0.034}$	$9.498^{0.066}$
IMU waist	$0.642^{0.005}$	$0.487^{0.052}$	$5.326^{0.010}$	$9.566^{0.101}$
IMU wrist	$0.637^{0.007}$	$0.316^{0.031}$	$5.392^{0.032}$	$9.544^{0.096}$

Table 8.1: Results of the model compared to the real dataset. [27]

One thing to note however, is that FID and MultiModality, measures the distribution of the data, and therefore give a general impression of data, however as the model is being provided with a IMU reading for a single sample, these readings could potentially be explained by the fact, that the data generated by the model is not as variable anymore. This can likewise be observed when compared to the original model from the paper. An observation is likewise it appears that changing the placement of the IMU does not significantly improve the precision of the model, therefore it is possible that the IMUs are not being utilized by the final model.

8.1 Outside comparison

By using the classifier trained by Skovbjerg et al. [25] in combination with the simulated IMU's, Figure 8.1 can be generated. By taking windows of 1 second,

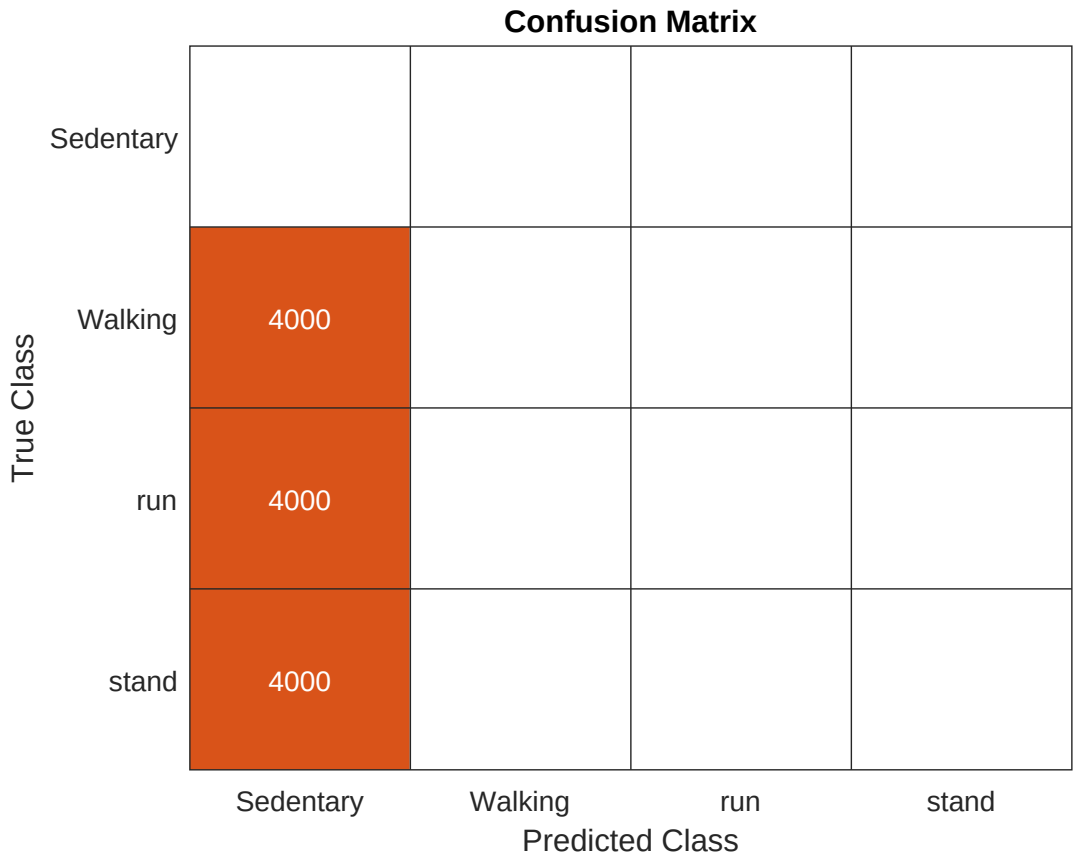

Figure 8.1: The classifier from Skovbjerg et al. [25] on the simulated IMU's.

which in the case of MDM is 20 samples, the classifier from Skovbjerg et al. [25] can be used, however the sample frequency of Skovbjerg et al. [25] is 100hz, the simulated motion is therefore resampled to 100hz, by using bicubic interpolation.

The results on figure 8.1 demonstrate, that the model trained by Skovbjerg et al. [25] can not classify the data from the generated IMU's. By looking further at the data and plotting the data captured by Skovbjerg et al. [25] on Figure 8.2 as normal curves compared to the simulated data.

Figure 8.2 illustrates the normal distribution of the RMS value of the x axis of the linear accelerometer, it can be seen that the normal distribution of the simulated data, is orders larger then the measurement obtained by Skovbjerg et al. [25].

Figure 8.3 likewise shows that the peak values lies a order of magnitude above those captured by Skovbjerg et al.

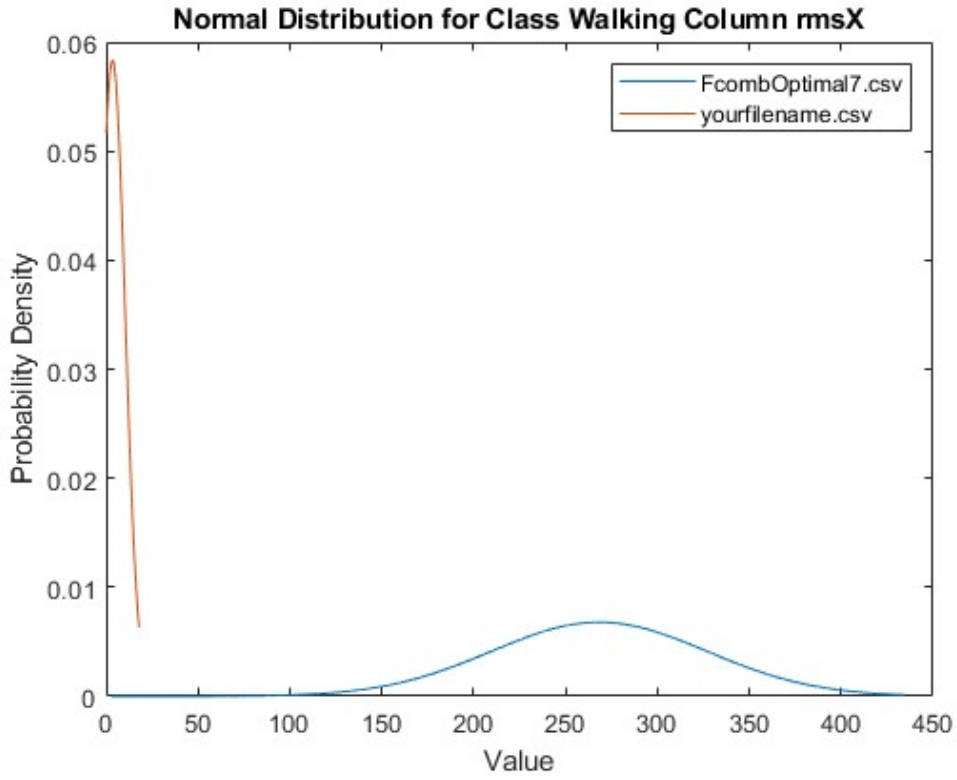

Figure 8.2: Rms values of the X axis, from Skovbjerg et al. [25] vs simulated. FcombOptimal7.csv is Skovbjerg et al. and yourfilename.csv is the MDM simulated data.

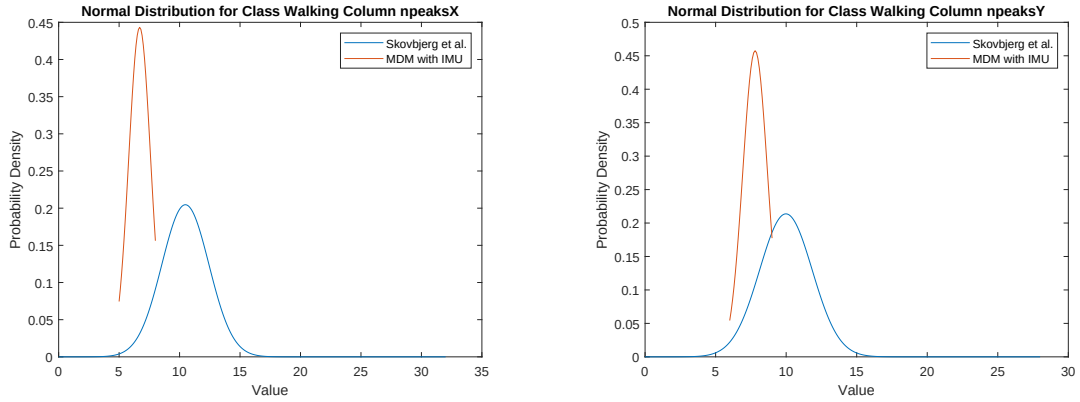

Figure 8.3: Peak values of x and z axis, from Skovbjerg et al. [25] vs MDM simulated.

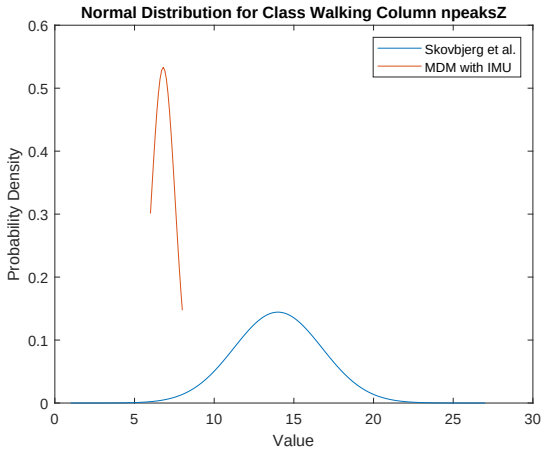

Figure 8.4: Peak values of the y axis, from Skovbjerg et al. [25] vs MDM simulated.

8.2 Sequence comparison

This section is a quantitative analysis of the produced skeleton, where the module with the IMU on the waist, will be used to generate a sequence, which will be compared to the input sequence from the validation set.

The Figures on 8.5 illustrates the generated motion for the sample with the prompt: a person walks in a forward motion. The prompt is taken from the validation set, as this allows for a IMU data to be compared to the generated sample.

Figure 8.6 and Figure 8.7 illustrates the IMU data, before and after the MDM model simulates the skeleton.

In Figure 8.5, the starting pose and the end pose of the skeleton are shown, with the line in the middle representing the trajectory of the 0th bone. The 0'th being the base localizations of the skeleton.

Figure 8.6 and Figure 8.7 are therefore the double differentiated of Figure 8.5,

with the Blue line being the left of Figure 8.5 and the orange line being the right of 8.5.

It is worth noting that, while the magnitude of the acceleration, produce by the modified MDM model, are smaller, it did place the peaks at the same locations as the input data, presumably these peaks occur when a foot is placed on the ground and the model has therefore, successfully modeled the persons walking cadence.

When looking at Figure 8.5 and Figure 8.6 it should be clear that the movement of the person is modeled incorrectly, the input sequence has more variance in the persons walking height, as displayed by the peaks on Figure 8.5, then it does on MDM generated person. This can likewise be observed between the Figures on 8.5 where its possible to observe a flatter line in the right Figure then on the left Figure.

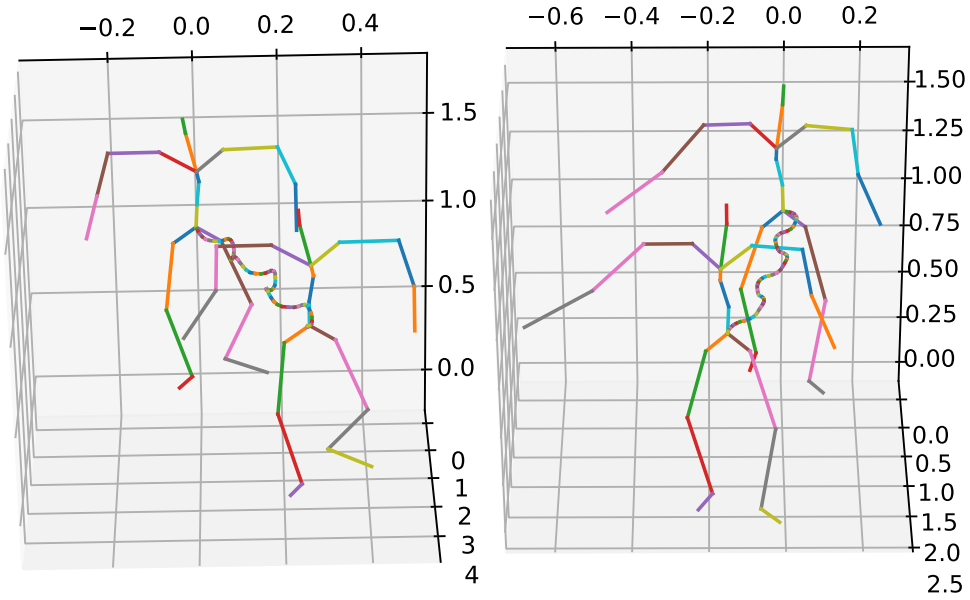

Figure 8.5: Validation sequence on the left, generated motion on the right.

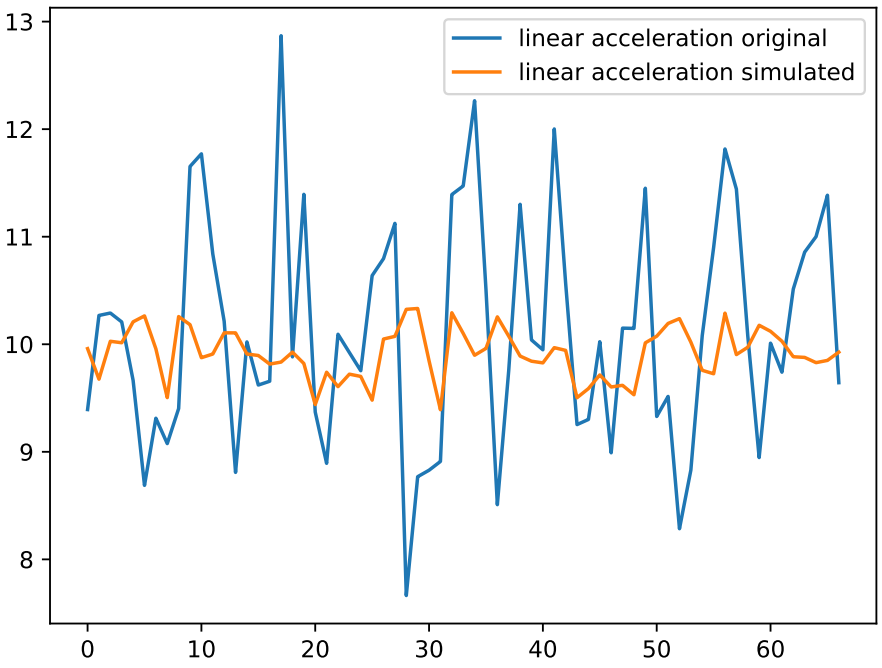

Figure 8.6: IMU linear acceleration magnitude for the sequence on Figure 8.5. the x axis is frame count and the y axis is acceleration in m/s^2.

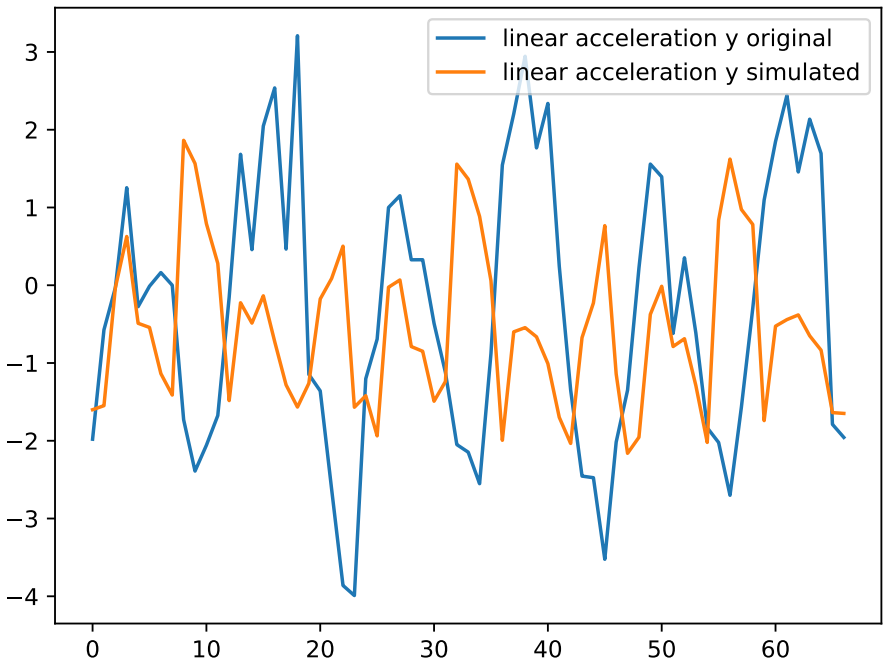

Figure 8.7: IMU linear acceleration y for the sequence on Figure 8.5. the x axis is the frame count and the y axis is the acceleration in m/s^2.

FFT Analysis

One interesting note from Figure 8.7 is that the peaks on the data, are noticeably reduced in amplitude. Performing a Fast Fourier Transform (FFT) will help determine if this is the case.

Figure 8.8 shows the FFT run on the data from Figure 8.6, and it is evident that all magnitudes in the simulated data are lower than those in the input data.

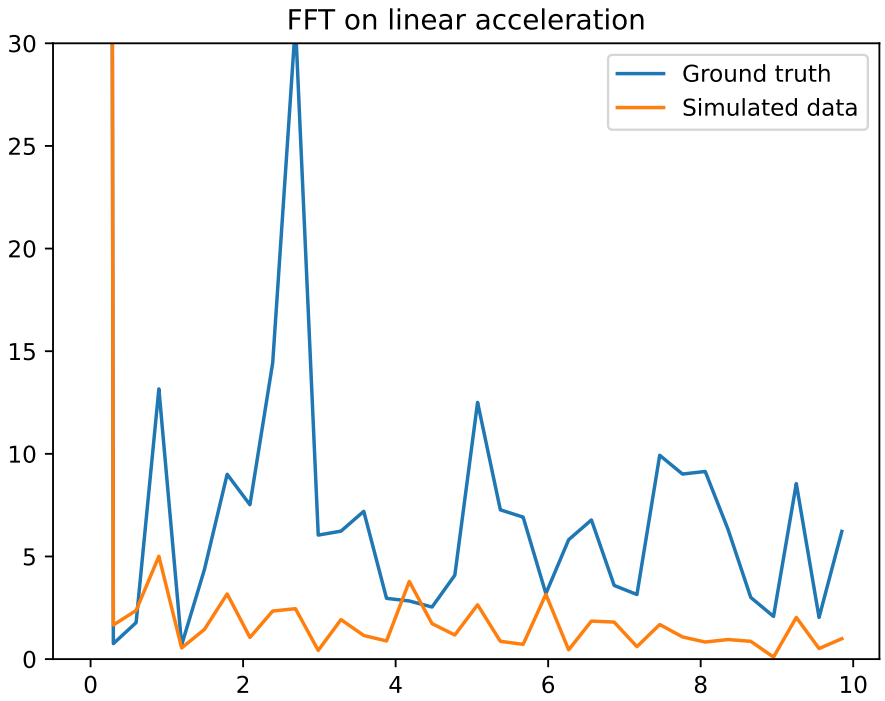

Figure 8.8: A FFT of the linear acceleration on Figure 8.5

Figure 8.8 shows that the motions modeled by the modified MDM are not 'realistic' and appear to have been smoothed over. The data displayed in Figure 8.8 represents the simulated linear acceleration of the generated motion, the same FFT can however be run on the movement of the 0'th bone, the one displayed on Figure 8.5.

Figure 8.9 clearly shows that, based on the IMU data, the modified MDM can not accurately model human walking motions; rather, it provides a smoothed interpretation of walking. The frequency present at 2Hz in normal human motion is lacking in the generated motion, this could correspond to the lack of traveled motion, on Figure 8.5 its observable that the validation travels 4 meters while the generated motion only does 2.5 meters.

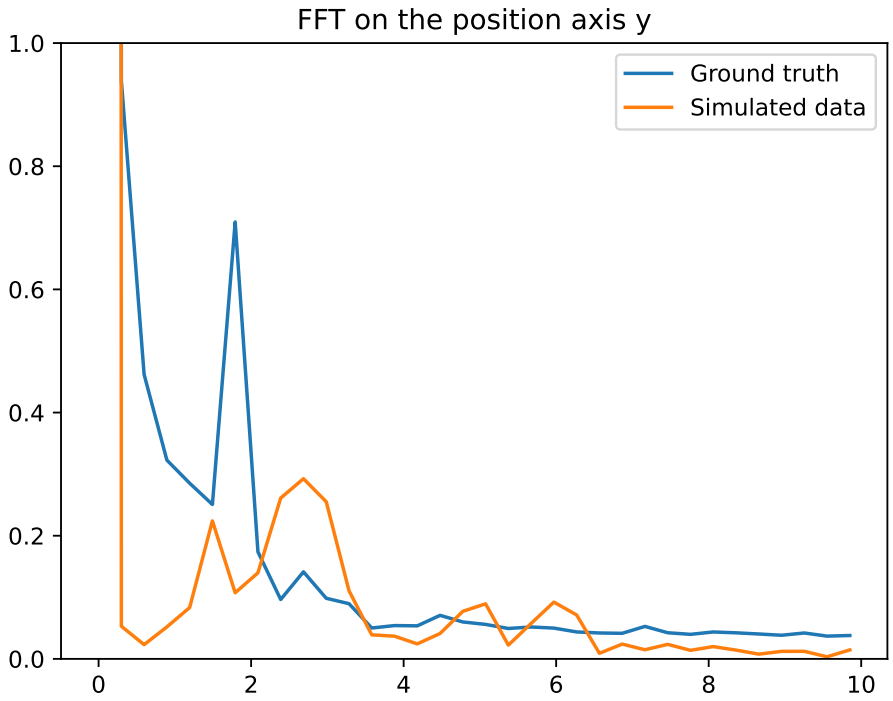

Figure 8.9: A FFT y axis position of bone 0 on Figure 8.5.

Chapter 9

Conclusion

The results are inconclusive, while some movements were generated, the results delivered in the Results chapter, does not show any significant deviation from the original model. This does not appear to be a training issue, as the model seems to perform comparably to the original model.

The chapter also demonstrates that the simulated data differs from the data captured by Skovbjerg et al., in that the data generated from the simulated IMUs is significantly larger in magnitude, However the testing section in 8.2 indicates that the motion lies an order of magnitude below the original motion in terms of its FFT.

Likewise, the training in the Training chapter suggests that the IMU input had minimal impact on the training of the model, However table 8.1 actually shows that, by modifying MDM with IMU data, the generated output improves, specifically the R Precision and FID does improve for all models.

When answering the Final Problem Formulation, Is it possible to generate synthetic data for training classifiers for fall detection?

The high-quality data in section 8.2 shows that MDM is unable to generate human-accurate data, it does generate something that looks by human motion according to the metrics given by Guy Tevet et al. [27], it however lacks the impacts and forces required for real looking motion.

The problem with unrealistic movements seems to be a know case, as a newer version, complete with physics simulation, has already been made. [30]

9.1 Future work

Future work could be in creating a loss function that brings the out motions closer to the input motions, specifically by bringing the acceleration in the outputted motion, closer to those of the original motion. Potentially though the use of a loss that targets the high frequencies in the output motions directly.

Bibliography

[1] Suad Albawendi et al. "Video Based Fall Detection with Enhanced Motion History Images PETRA '16". In: (). DOI: 10.1145/2910674.2935832. URL: http://dx.doi.org/10.1145/2910674.2935832.

[2] Fabio Bagalà et al. "Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls". In: *PLOS ONE* 7.5 (May 2012), e37062. ISSN: 1932-6203. DOI: 10.1371/JOURNAL.PONE.0037062. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0037062.

[3] Roya Bamzar. "Assessing the quality of the indoor environment of senior housing for a better mobility: a Swedish case study". In: 34 (2019), pp. 23–60. DOI: 10.1007/s10901-018-9623-4. URL: https://doi.org/10.1007/s10901-018-9623-4.

[4] Eduardo Casilari and Carlos A. Silva. "An analytical comparison of datasets of Real-World and simulated falls intended for the evaluation of wearable fall alerting systems". In: *Measurement: Journal of the International Measurement Confederation* 202 (Oct. 2022). ISSN: 02632241. DOI: 10.1016/J.MEASUREMENT.2022.111843.

[5] Srijan Das et al. "Toyota smarthome: Real-world activities of daily living". In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-October (Oct. 2019), pp. 833–842. ISSN: 15505499. DOI: 10.1109/ICCV.2019.00092.

[6] Lesley D. Gillespie et al. *Interventions for preventing falls in older people living in the community*. Sept. 2012. DOI: 10.1002/14651858.CD007146.pub3.

[7] Nil Goyette et al. "changedetection.net: A new change detection benchmark dataset". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2012), pp. 1–8. ISSN: 21607508. DOI: 10.1109/CVPRW.2012.6238919.

[8] Chuan Guo et al. "Action2Motion: Conditioned Generation of 3D Human Motions". In: (July 2020). DOI: 10.1145/3394171.3413635. URL: http://arxiv.org/abs/2007.15240http://dx.doi.org/10.1145/3394171.3413635.

[9] Chuan Guo et al. "Generating Diverse and Natural 3D Human Motions from Text". In: (). URL: https://ericguo5513.github.io/text-to-motion.

[10] Hochul Hwang et al. "ElderSim: A Synthetic Data Generation Platform for Human Action Recognition in Eldercare Applications". In: (). DOI: 10.1109/ACCESS.2021.3051842. URL: https://ai4robot.github.io/ElderSim..

[11] Jinhyeok Jang et al. "ETRI-Activity3D: A Large-Scale RGB-D Dataset for Robots to Recognize Daily Activities of the Elderly". eng. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway: IEEE, 2020, pp. 10990–10997. ISBN: 9781728162126. DOI: 10.1109/IROS45743.2020.9341160.

[12] Shehroz S. Khan and Jesse Hoey. "Review of fall detection techniques: A data availability perspective". In: *Medical Engineering and Physics* 39 (Jan. 2017), pp. 12–22. ISSN: 18734030. DOI: 10.1016/J.MEDENGPHY.2016.10.014.

[13] Jochen Klenk et al. "The FARSEEING real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls". In: *European Review of Aging and Physical Activity* 13.1 (Oct. 2016), pp. 1–7. ISSN: 18137253. DOI: 10.1186/S11556-016-0168-9/FIGURES/3. URL: https://eurapa.biomedcentral.com/articles/10.1186/s11556-016-0168-9.

[14] Naureen Mahmood et al. "AMASS: Archive of Motion Capture as Surface Shapes". In: (). URL: https://amass.is.tue.mpg.de/..

[15] Aihua Mao et al. "Highly Portable, Sensor-Based System for Human Fall Monitoring". In: (2017). DOI: 10.3390/s17092096. URL: www.mdpi.com/journal/sensors.

[16] Briana L. Moreland et al. "A Descriptive Analysis of Location of Older Adult Falls That Resulted in Emergency Department Visits in the United States, 2015". In: *American Journal of Lifestyle Medicine* 15.6 (Nov. 2021), pp. 590–597. ISSN: 1559-8276. DOI: 10.1177/1559827620942187. URL: http://journals.sagepub.com/doi/10.1177/1559827620942187.

[17] United Nations. *World Population Ageing 2015: Highlights*. United Nations, 2016. URL: https://www.un-ilibrary.org/content/books/9789210576123.

[18] N. Noury et al. "A proposal for the classification and evaluation of fall detectors". In: *IRBM* 29.6 (Dec. 2008), pp. 340–349. ISSN: 1959-0318. DOI: 10.1016/J.IRBM.2008.08.002.

[19] NVIDIA. *NVIDIA Omniverse Replicator Generates Synthetic Training Data for Robots*. Website. 2021. URL: https://developer.nvidia.com/blog/bootstrapping-object-detection-model-training-with-3d-synthetic-data/.

[20] Carmen A Pfortmueller et al. “Fall-Related Emergency Department Admission: Fall Environment and Settings and Related Injury Patterns in 6357 Patients with Special Emphasis on the Elderly”. In: (2014). DOI: 10.1155/2014/256519. URL: http://dx.doi.org/10.1155/2014/256519.

[21] Matthias Plappert, Christian Mandery, and Tamim Asfour. “The KIT Motion-Language Dataset”. In: *Article originally appeared in Big Data* 4.4 (2016), pp. 236–252. DOI: 10.1089/big.2016.0028. URL: http://dx.doi.org/10.1089/big.2016.0028..

[22] Amir Shahroudy et al. “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis”. eng. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-. IEEE, 2016, pp. 1010–1019. ISBN: 9781467388511. DOI: 10.1109/CVPR.2016.115.

[23] Yoli Shavit and Itzik Klein. “Boosting Inertial-Based Human Activity Recognition with Transformers”. In: *IEEE Access* 9 (2021), pp. 53540–53547. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3070646.

[24] Anuradha Singh et al. “Sensor Technologies for Fall Detection Systems: A Review”. In: *IEEE SENSORS JOURNAL* 20.13 (2020). DOI: 10.1109/JSEN.2020.2976554. URL: https://www.ieee.org/publications/rights/index.html.

[25] Frederik Skovbjerg et al. “Monitoring Physical Behavior in Rehabilitation Using a Machine Learning–Based Algorithm for Thigh-Mounted Accelerometers: Development and Validation Study”. In: (2022). DOI: 10.2196/38512.

[26] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: (Mar. 2015). URL: http://arxiv.org/abs/1503.03585.

[27] Guy Tevet et al. “Human Motion Diffusion Model”. In: (Sept. 2022). URL: http://arxiv.org/abs/2209.14916.

[28] Ashish Vaswani et al. “Attention Is All You Need”. In: (June 2017). URL: http://arxiv.org/abs/1706.03762.

[29] Deidre Wild, U. S.L. Nayak, and B. Isaacs. “How dangerous are falls in old people at home?” In: *Br Med J (Clin Res Ed)* 282.6260 (Jan. 1981), pp. 266–268. ISSN: 0267-0623. DOI: 10.1136/BMJ.282.6260.266. URL: https://www.bmj.com/content/282/6260/266https://www.bmj.com/content/282/6260/266.abstract.

[30] Ye Yuan et al. “PhysDiff: Physics-Guided Human Motion Diffusion Model”. In: *arXiv preprint arXiv:2212.02500* (2022).