Study on Human Motion Forecasting using Self-Attention based Approach

Andi Prademon Yunus
Graduate School of Engineering
Mie University

A dissertation submitted for the degree of
Doctor of Philosophy in Engineering
2023 July

Abstract

Human motion forecasting is a necessary variable to analyze human motion concerning the safety system of the autonomous system that could be used in many applications, such as in auto-driving vehicles, auto-pilot logistics delivery, and gait analysis in the medical field. At the same time, many types of research have been conducted on 2D and 3D human motion prediction for short and long-term goals. In this dissertation, human motion forecasting in the 2D plane has been conducted as a reliable alternative in motion capture of the RGB camera attached to the devices. While for a more precise location in the real-world automation application, 3D human motion forecasting is also necessary since the device could detect the exact location in the 3D plane. The unannotated dataset is used as the samples to conduct the works on 2D human motion forecasting to realize the usability of the task in real-world applications. On the unannotated dataset prediction task, the author proposed the feature extraction by OpenPose as the commonly used pose estimator and then obtained the future prediction movement by the RNN-LSTM or Kalman Filter. As a result, the usability of human motion prediction by applying the RGB camera is confirmed. The prediction results obtained by the Kalman Filter show better performance than the RNN-LSTM based on the correct prediction result within the correct location range.

In contrast, the annotated dataset is used to improve the quality and performance of the prediction results obtained by the models. The author proposed a method, the time series self-attention approach to generate the next future human motion in the short-term of 400 milliseconds and long-term of 1000 milliseconds, resulting that the model could predict human motion with a slight error of 23.51 pixels for short-term prediction and 10.3

pixels for long-term prediction on average compared to the ground truth in the quantitative and qualitative evaluation. Our method outperformed the LSTM and GRU models on the Human3.6M dataset based on the MPJPE and MPJVE metrics. The average loss of correct key points varied based on the tolerance value. Our method performed better within the 50 pixels tolerance. In addition, our method is tested by images without key point annotations using OpenPose as the pose estimation method. As a result, our method could predict well the position of the human but could not predict well for the human body pose. This research is a new baseline for the 2D human motion prediction using the Human3.6M dataset.

Subsequently, studies were carried out to predict human motion in 3D, aiming to improve various applications. Building upon the groundwork established by previous studies, the time series self-attention method was utilized as the model with modifications to accommodate 3D input data. As a result, our approach showed good performance in both short and long-term prediction tasks. It had an average error of $36.4mm$ between the prediction and ground truth in short-term predictions and $73.2mm$ in long-term predictions.

Overall, the studies of human motion forecasting have been conducted based on 2D and 3D input. In this study, we confirmed the realization of our method to predict human motion in the short and long term.

Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have contributed to my journey and supported me along the way.

First and foremost, I am incredibly grateful to my mother and father, my brother Andi Prawirawan, my sisters Anna Syafarina and Adita Agreni Yunus, and my wife Nurul Huda, for their unwavering love, encouragement, and understanding. They have been my pillars of strength, offering guidance and support during both the challenging and joyous moments of my life.

I am sending my greatest acknowledgment to my supervisors, Prof. Naoki Isu, Prof. Tetsushi Wakabayashi, Assistant Prof. Kento Morita, and my special collaborator Nobu C. Shirai. They have been enabling me to this extent, giving me the best advice and support not only as an advisor but also as a family, a friend, and a companion.

Furthermore, To the teacher in Engineering Faculty in general, and in Information Engineering Department especially. Along with the very helpful department, faculty, and international relation office staff that gives me silky smooth problem-solving during my study.

Lastly, I would like to express my appreciation to friends who are more like family in Japan and in Indonesia. As well as I want to acknowledge the supportive community of the Indonesia Student Association or Persatuan Pelajar Indonesia (PPI) in Mie and in Japan.

To everyone who has played a role, big or small, in shaping who I am today, I offer my heartfelt thanks. Your contributions have left an indelible mark on my journey, and I am grateful for the support, guidance, and inspiration you have provided.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Chapter 1

Introduction

1.1 Background

Autonomous system utilization has become more progressive with the advancement of artificial intelligence. Many applications can be operated autonomously, such as self-driving cars and auto-pilot robots. The system's decision-making is taken autonomously by the system. Several things to consider for efficient, effective, and safe decision-making while operating the device. In the matter of the safety of the system, some devices apply depth or distance sensor[1, 2] to keep the device at a safe distance from any object [3, 4]. However, in the case of a moving object like a human, the system might get a blind spot when a human body has not reached the depth sensor measurement area but will be on the track of the device route. With this in mind, the consideration of the behavioral knowledge of the object is necessary to increase the sensitivity and tackle the lack of the blind spot in the current autonomous's safety system. There are several advantages of using the RGB camera for the safety system:

1. Computer-aided system.
2. Wide range measurement.

The other utilities that can benefit from using human motion forecasting are:

1. Computer-aided falling down prevention for the disabled and elderly.
2. Additional input for the human pose estimation training as the biased data.
3. Additional input for the human position tracking research.

1. INTRODUCTION

Human motion forecasting has been retracted attention for advancing methods, strategies, and results. Several types of research, such as using recurrent neural networks, gated recurrent units (GRU), long short-term memory (LSTM), and Transformer networks, are conducted in many ways for this problem. Different approaches with different inputs and expecting different outputs in the process, which the task could be divided by the input of two-dimensional coordinates and three-dimensional coordinates. Both of these works have advantages and disadvantages in the process and precision of prediction. The two-dimensional input contains x and y coordinates in the frame of the image, which makes the input could be from the RGB camera that is commonly applied in many systems. As well as, the process of two-dimensional input is easier to compute, considering the input size is less than the three-dimensional input. However, in the autonomous system, the device might need to consider the z coordinate to measure the real distance in the real world. As for now, the three-dimensional input is still under development, and the input is only given by the RGB camera with a depth sensor to obtain the z coordinate, which makes this input still limited by the cost of the input device. Apart from the input data, human motion forecasting research has been developed along with the human pose estimation problem to support the necessity of the input.

Various Machine Learning (ML) techniques have been used to predict better results judging by the distance of prediction to the ground truth. While conducting the research, the baseline has been set to improve the prediction by one evaluation method. It was started from human motion prediction with recurrent network model[5, 6, 7]. Recent approaches have been followed with more techniques, and the renewal evaluation method to measure the distance in millimeters (mm)[8, 9]. These researches were conducted with the input of 3D input data and expecting the 3D output in the long-short term prediction. While for the 2D input data, some research has been conducted with various inputs of the dataset, approaches, and output[10, 11]. As for this research, the author considers using the most commonly applied dataset for the human motion forecasting and human pose estimation dataset, which is the Human3.6M dataset, to set the baseline of the 2D human motion forecasting and join and improve the 3D human motion forecasting research.

The author conducted several experiments to improve the prediction result with the combination of Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN),

and the Transformer Network model applied in the 2D and 3D input data. Inspired by the natural language processing problem, the attention scheme shows an excellent result in understanding the context and the connection between words in sentences[12]. This attention scheme seems to be applicable in the human motion forecasting problem with some modifications on input data and models. The transformer network is also used in many applications with some modifications based on the problem. For example, the Vision Transformer is developed to do the object detection task[13, 14, 15] as well as in the time-series prediction task[16]. Furthermore, the author proposes several novel methods to improve human motion forecasting task performance. An overview of the contributions of our work is presented in Section 1.2.

1.2 Overview of Human Motion Forecasting

This section provides an overview of the human motion forecasting process. There are four steps to generate the human motion prediction. These include feature preparation, learning or training the model, feature prediction, and feature-to-video interpretation. Each step is briefly introduced below.

1.2.1 Feature Preparation

Given the input generated from the device, like the RGB camera, RGB-D camera, or motion capture by the sensor. This input data must be preprocessed for the feature needed as the learning data in the next step. For example, if the sequences of images containing the human in the frame have been obtained. One needs to extract the human body feature first and get the pose in the format of joint key points. However, it depends on the expected input needed. As for this work, the author separated the research based on the input:

1. 2D input: given the input generated from the motion capture by the distance sensor provided by the dataset for the training and testing process interpolated in the x and y coordinate of the frame image. While the input generated from the pose estimation result is used to evaluate the model on the biased data. Let the input be $X \in \mathbb{R}^{2N}$ consisting of x and y coordinate in the N human body key points.

2. 3D input: given the input generated from the motion capture by the distance sensor provided by the dataset for the training and testing process interpolated in the x, y, and z coordinate. Let the input be $X \in \mathbb{R}^{3N}$ consisting of x, y and z coordinate in the N human body key points.

Data preparation is done by stacking the sequence of frames in the sliding window process. This process becomes the standard method to generate the input feature to the model. Given the coordinate data $\mathbf{X} = [X_1, X_2, \ldots, X_n]$, where X_i is the vector of coordinate data in the frame i with respect of the key points. Further detailed descriptions of the data preparation will be explained in Section 3 and 4.

1.2.2 Model Training

Predicting the sequence of frame vectors is the main task of the model. The model extracted important information from the training data. Thus, later it can generate the prediction from the unobserved data, expecting the next determined number of sequence frames as the output. Several methods to obtain the best model have been applied, which will be further explained in Section 2, 3, and 4.

1.2.3 Feature Prediction

After training the model using the training data, the pattern of the samples has been transformed into the model to recognize. The model is expected to be able to predict the unobserved data based on the pattern that has been trained. The feature predicted by the model will be determined as a good result or not a good result depending on the evaluation method calculated by how far the distance from the prediction to the ground truth is.

1.2.4 Feature-to-video Interpretation

Since the result is in the form of the coordinate features, the visualization of the feature is needed to see how good the prediction looks in the qualitative evaluation and to realize the output in the actual video or any suitable format.

1.3 The Organization of This Dissertation

In this Section, the author describes the overall organization of this dissertation. Chapter 1 introduces the background, main problems, goals, methods in general, and how the author organized the chapters based on the task. Chapter 2 describes the research on 2D human motion forecasting by using unannotated data to realize the usability of human motion prediction in real-world applications. After realizing the usability of the human motion prediction real-world applications, Chapter 3 describes the research on the 2D human motion prediction by using annotated data from the commonly used dataset in human motion research. While human motion forecasting could be used for predicting individuals using 2D inputs like the RGB camera, which also means that it could be applied using the 3D data when the input is a certain coordinate of humans based on the motion capture devices that can measure the location with quite high precision. This brings us to Chapter 4, which describes the research of human motion forecasting using the 3D input by the commonly used annotated dataset. Finally, the author provides the discussion and conclusion in Chapter 5.

1.4 Motivation for the Research

1.4.1 Importance of Human Motion Forecasting Research

Human motion forecasting is the task of predicting the future movements of individuals in a given environment. Why does this task become important? In several cases, this task could greatly improve safety, efficiency, and user experience across various applications.

1. Robotics and autonomous systems: Accurately forecasting human motion enables robots and autonomous systems to interact more safely and efficiently with people in their environment.

2. Healthcare: Human motion forecasting can be used to assist with rehabilitation and to monitor and predict the progression of movement disorders.

3. Transportation: Accurately forecasting human motion can help optimize pedestrian traffic flow and improve safety in public transport systems.

4. Gaming and entertainment: Human motion forecasting can be used to create more realistic and immersive virtual environments.

5. Surveillance and security: Human motion forecasting can be used to detect and respond to potential security threats in real time.

1.4.2 Applications of Human Motion Forecasting

As explained in Section 1.4, the task of human motion forecasting could improve the safety, efficiency, and user experience across various applications. In terms of practical uses of human motion forecasting tasks, for example:

1. Improving the interaction between robots and humans in various settings.

2. Supporting physical therapy and tracking the development of movement disorders.

3. Streamlining pedestrian traffic flow and enhancing safety in public transportation.

4. Creating more believable virtual environments for gaming and entertainment.

5. Detecting and responding to potential security threats in real-time.

6. Analyzing and improving athletic performance in sports.

7. Enhancing the interaction between people and computer systems.

1.5 Scientific Contributions

The author highlighted the scientific contributions of this dissertation as follows:

1. Improved understanding of human movement: The study of human motion prediction has led to a deeper understanding of the underlying patterns and principles of human movement.

2. Improved human-robot interaction: Human motion prediction is crucial for improving human-robot interaction and has led to the development of new safety protocols and interaction methods.

3. Advances in the computer vision and machine learning method: Human motion prediction has pushed the boundaries of machine learning and computer vision by requiring algorithms to make predictions based on complex and dynamic data.

4. Development of the new baseline of the 2D human motion forecasting: 2D human motion forecasting using commonly used annotated datasets and commonly used evaluation metrics could help set the baseline for the related works to follow.

5. Advances in the deep learning application of the attention-based method to understand the human motion prediction task, which also could help the other research to use the same model's structure in another task.

1.6 Summary of the Introduction

In this chapter, the author introduces the basic knowledge needed to understand the human motion forecasting task. The background, main problems, goals, methods, and expected outcomes are explained explicitly. While the following chapters describe more in detail. The author has given examples of the applications, and the importance of the human motion forecasting research described in Section 1.4 as well as the detail of the contributions of our work have been described in Section 1.5.

Chapter 2

RNN-LSTM and Kalman Filter Based Time Series 2D Human Motion Forecasting

2.1 Introduction

While machines were developed to coexist and help the work of a human, a system that considers the behavior of the surroundings for the device is needed. For the example of the system in human interaction, humans constantly interact with their surroundings along with other living things and nonliving things. Many researchers and companies have developed this sensing system for many uses, such as the distancing sensor to measure the distance in the auto-braking system for the car's safety system. However, the system will detect everything at a certain distance as a threat using the distance sensor without considering the object. Compared to a camera, distance sensors are more expensive. And one more reason for users to use the camera is that the development can still go further on distinguishing objects.

This research aims to develop a system that recognizes the environment's behavior in the next 1-second movement. As for the first step, the system determines the human body as an object. Then, by predicting the human action, which has a problematic pattern to be recognized, the system will understand where the human will move, giving another delay time for the system to do the action. For these reasons, the scope of human motion has been limited by only using simple human motions like hand

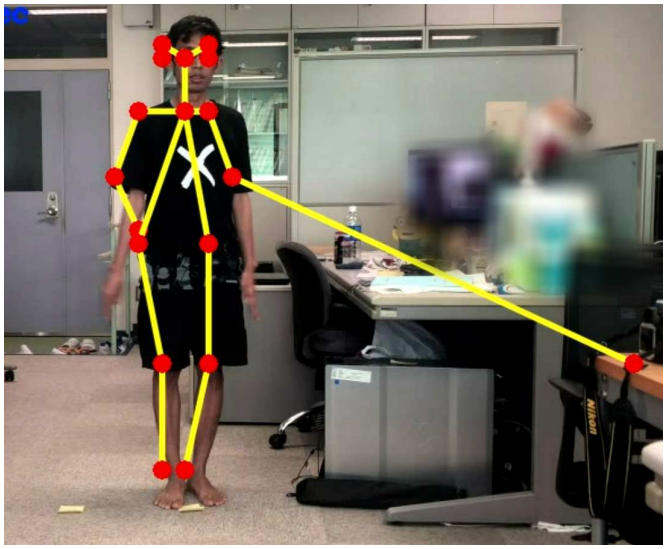


Figure 2.1: Pose Estimation: OpenPose failed to estimate the full body human pose key points.

gestures and walking movements.

For a more reliable dataset, some researchers use the RGB-D camera, like Kinect, to estimate the human body pose [17, 18]. Nonetheless, this paper uses the RGB camera since it has been commonly used recently in any aspect of life. The main objective of this research is to predict the motion of the human pose, focusing on the data obtained from the RGB camera and preprocessed by the pose estimation method. In this research, we rely on OpenPose to estimate the human body pose. However, the data that OpenPose has preprocessed did not consistently generate the precise estimation of human body parts, shown in **Fig. 2.1**. With this in mind, the data needs to be prepared to be the input of the prediction method. We determine the estimation failure by the OpenPose as the unstable data as the challenge in this paper. Related research has been conducted to predict human motion with the RGB camera focusing on sports activities like boxing, karate, or taekwondo. The result shows 0.5 seconds prediction of human movement has been obtained. Nonetheless, the accuracy of the forecast was not found in this paper [19].

Recently, Recurrent Neural Network (RNN) has been used to deal with the specific problem for prediction, inclusively the difficulty of predicting human motion [17, 20, 21]. Because the individual behavior of humans is varied and unique, a short-term and long-

(a) Sample 1

(b) Sample 2

(c) Sample 3

Figure 2.2: Our dataset samples

term prediction method is compulsory to clear up the forecasting problem. Recurrent Neural Network Long Short Term Memory (RNN-LSTM) implements the short-term and long-term prediction method based on its extended memory to store the weight of parameters with reliable certainty of prediction results [22, 23]. While Kalman Filter has been used as the prediction method that is reliable enough based on the result from the time-series data [24], realizing the human motion prediction using RNN-LSTM and Kalman Filter and comparing the result to show the performance on the unstable data like human motion is the main idea of this study. This research has been updated with more data and evaluation methods from the previous experiment [25].

2.2 Preliminaries

In this section, the author describes the tools, terminologies, and methods used in this research.

(a) Sample 1

(b) Sample 2

(c) Sample 3

Figure 2.3: CMU dataset samples

2.2.1 Dataset

COCO dataset key points have been used to generate 18 key points of the human body pose in the frame [26, 27]. The COCO dataset is used as training data in feature extraction to detect the key points of the human body in our dataset and CMU dataset. As the first step of the prediction, simple human motion and gestures are needed, such as hand gestures and simple walking. Our dataset is consisting 30 fps (frame per second) videos with a frame dimension of 960 $\times$ 540 pixels, as shown in **Fig. 2.2**. The CMU dataset has been used as a step forward to a more complex motion as a comparison for our dataset. This CMU dataset consists of 2605 videos with 30 fps and 352 $\times$ 240 pixels frame dimension, as shown in **Fig. 2.3**.

2.2.2 YOLOv3

You only look once (YOLO) is an object detection system targeted for real-time processing. Fast YOLO is the fastest general-purpose object detector in the literature, and

YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection [28].

2.2.3 Pose Estimation: OpenPose

For the 2D real-time multi-person keypoint detection, OpenPose provides 15 or 18, or 25 body/foot key points estimation based on the dataset key points. OpenPose generates 25 joints of the human body with BODY25 joints detection from the RGB image [26]. The features obtained from OpenPose are not as precise as the manually annotated data. With OpenPose as the pose estimation method, one can predict human motion without key point annotations in the frames. Considering the practical applications, a pose estimator such as OpenPose is needed to make it possible for a method to directly predict key points from image data without key point annotations.

2.2.4 Kalman Filter

Kalman Filter is an adequate iterative filter that estimates the internal state of a linear dynamic system from a series of noisy measurements [29]. Kalman Filter has been used in some applications for short-term forecasting [24]. Kalman Filter is based on two primary functions. The first step is the prediction step. The first guess is generated about what we think is valid and the certainty that the estimation is correct. After that, Kalman Filter generates a different estimate with a weighted average calculation. Then, the new guess is generated by the previous guess, which the weighted average has corrected, and these steps are iteratively calculated.

2.2.5 Recurrent Neural Network

Recurrent Neural Network is one of the classes in the neural network where the connections on the units create a structure along with the temporal sequence. RNN has the internal memory to process the series of data inputs. The computing units in the RNN have a time-varying real-valued activation and adjustable weight. RNNs are created by recursively applying the same weights over a graph-like structure [30]. The learned model in RNN has the exact input size since it transitions from one state to another.

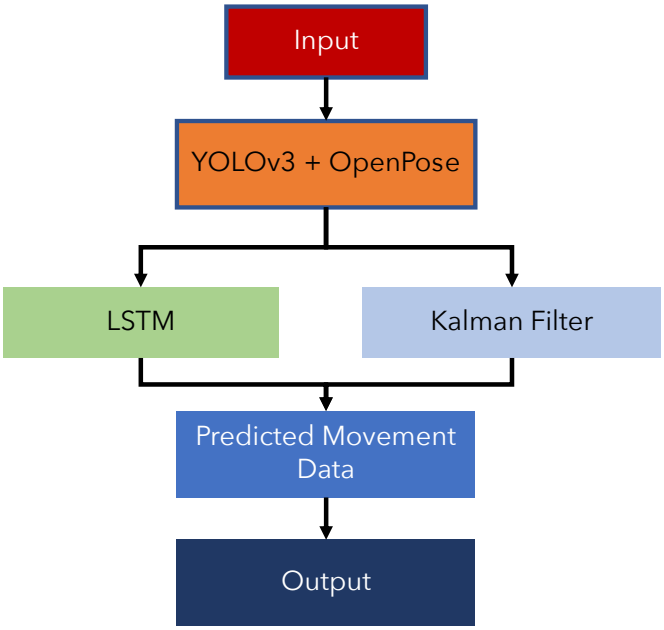


Figure 2.4: Proposed method overview

2.2.6 Long Short-Term Memory

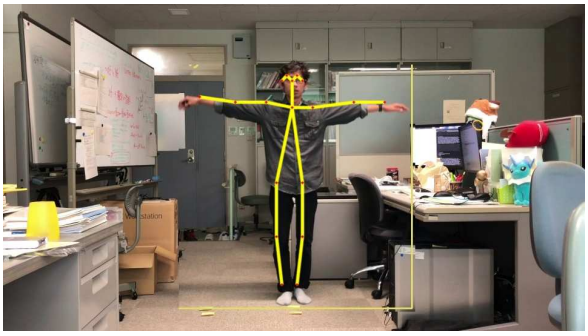
The short-term data could be stored based on the RNN's internal memory that stores the weights and computations of the data. However, RNN cannot keep the series of data in the longer term to be predicted. Here, LSTM performs the role of the extended form of RNN, which contains the extended memory by structure. Hochreiter and Schmidhuber invented LSTM in 1997, which works and can handle signals that mix low and high-frequency components [23].

2.3 Proposed Method

One second of the human motion forecast is the goal of this research. First, the human body pose is defined by parts covering the head, neck, shoulders, elbows, wrists, hips, knees, and ankles as the coordinate data of the features. Then, this coordinate data will be converted to the movement data containing the distance and direction based on the body parts of the frame. Finally, the movement data will be processed using the RNN-LSTM and Kalman Filter to predict.

(a) YOLOv3 is applied on the Our dataset.

(b) OpenPose is applied on the Our dataset after the YOLOv3 crop.

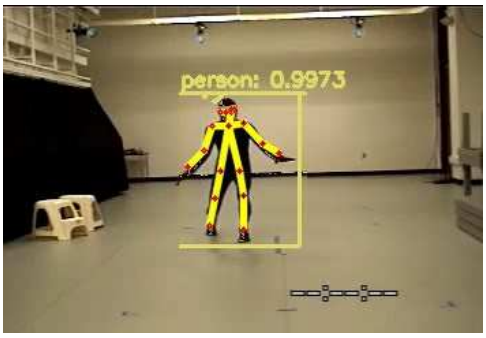


(c) Our dataset sample after YOLOv3 and OpenPose features extraction.

Figure 2.5: Our dataset features extraction process

(a) YOLOv3 is applied on the CMU dataset.

(b) OpenPose is applied on the CMU dataset after the YOLOv3 crop.



(c) CMU dataset sample after YOLOv3 and OpenPose features extraction.

Figure 2.6: CMU dataset features extraction process

2.3.1 Feature Extraction

The feature is one key factor in obtaining the prediction as the data that will be calculated are from this step. The OpenPose from OpenCV implements the state-of-the-art method to estimate the human body pose with an RGB camera. The key points will detail the coordinate data used in the prediction method based on the COCO dataset. Nonetheless, the results obtained by OpenPose are not constantly stable, as shown in **Fig. 2.1**. In this paper, the problem of the estimation failure by OpenPose is solved by narrowing the frame input for OpenPose by using YOLOv3. YOLOv3 detects the object of the human body in the frame, as shown in **Fig. 2.6a** and **2.5a**. With this cropping limitation, the pose estimation is only focused on the human body frame as shown in **Fig. 2.6b** and **2.5b**.

Given the coordinate data x and y based on the result of the pose estimation by OpenPose, the obtained raw x and y coordinate values are not suitable for motion

estimation using our estimation model because their value range depends on the image size. Equations 2.1 and 2.2 convert the obtained coordinate value at i-th frame x_i and y_i to the movement data expression that consists of distance d_i and direction θ_i.

$$d_i = \sqrt{(x_i - x_{i-fs})^2 + (y_i - y_{i-fs})^2} \tag{2.1}$$

$$\theta_i = \arcsin\left(\frac{y_i - y_{i-fs}}{d_i}\right) \tag{2.2}$$

where fs is the constant value of the frame step of 30.

2.3.2 Pose Prediction Using Kalman Filter

Kalman Filter consists of the estimate function and the correction update function. This research defines Kalman Filter in two parts to calculate the movement data distance and direction separately using Equations 2.3 and 2.4.

$$d_i = d_{i-1} + \frac{(\sigma_i \times d_{i-1}) + (\sigma_c \times \delta_i)}{\sigma_i \times \sigma_c} \tag{2.3}$$

$$\theta_i = \theta_{i-1} + \frac{(\sigma_i \times \theta_{i-1}) + (\sigma_c \times \delta_i)}{\sigma_i \times \sigma_c} \tag{2.4}$$

where σ_i is the initial weight and an updated weight of d_i, σ_c refers to the constant noise weight, δ_i is the data obtained by OpenPose.

2.3.3 Pose Prediction Using RNN-LSTM

Pose prediction by Kalman Filter could fail to estimate the sudden move, which is why we propose RNN-LSTM as a comparison to predict human motion. Three stacked hidden layers of RNN-LSTM are used as the learning model to process the input of 14 key points of human body parts. Likewise, other related research used three stacked-layer of RNN-LSTM [17, 18, 19]. The loss function is defined by the Mean Squared Error (MSE) to calculate the loss value in the training process of RNN-LSTM.

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{x} - x_i) \tag{2.5}$$

2.3.4 Evaluation Method

The experiments have been performed by comparing the CMU dataset and our dataset using the Kalman Filter and RNN-LSTM. To evaluate the accuracy of the prediction, the euclidean distance between two nodes from different frames is calculated to compare the lengths from the ground truth to the predicted result [24].

$$E = \sqrt{(x_{i+30} - x_p)^2 + (y_{i+30} - y_p)^2} \tag{2.6}$$

Where i refers to the number of the frame, x_i and y_i are the x and y coordinate data at i-th frame. The x_p and y_p are the coordinates of x and y of the prediction result, calculated by d_i and θ_i, movement data at the i-th frame, as shown in Equations 2.7 and 2.8.

$$x_p = x_i + d_i \tag{2.7}$$

$$y_p = y_i + d_i \tag{2.8}$$

As for the evaluation of the satisfiable prediction, E_p determines the evaluation based on the percentage of the limitation satisfiable range in a frame by:

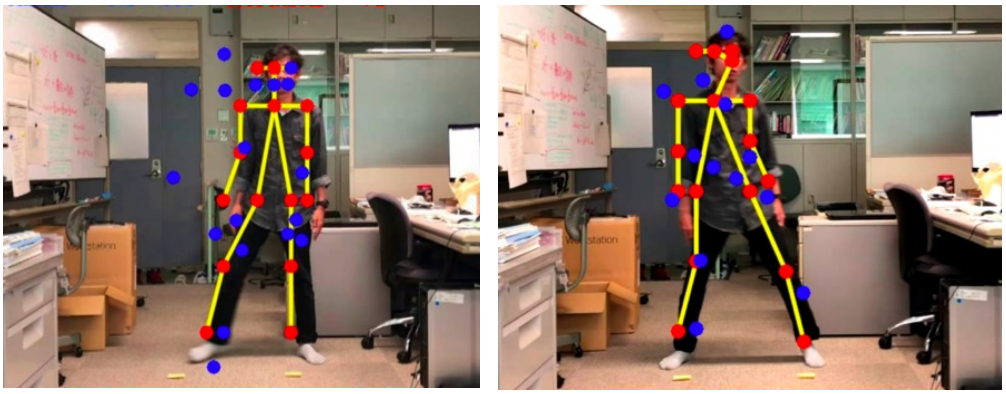
$$E_p = \frac{N_s}{N} \times 100 \tag{2.9}$$

N_s represents the number of the prediction results below the satisfiable range, and N represents the total frame.

2.4 Experiment Results

Figures 2.7a and **2.7b** shows the prediction results on our dataset. The red nodes are the actual position of the key points, and the blue nodes are the forecast position of the key points. While figure 6 shows the prediction results for the CMU dataset. **Table. 2.1** shows the average evaluation distances by the percentage of successful prediction, error average, and error median for RNN-LSTM and Kalman Filter on our and CMU datasets.

Generally, Kalman Filter shows better results than RNN-LSTM on the predicted key points, with 93.2% of the predictions in the distance range of successful prediction.



(a) Our dataset with RNN-LSTM

(b) Our dataset with Kalman Filter

(c) CMU dataset with RNN-LSTM

(d) CMU dataset with Kalman Filter

Figure 2.7: Prediction results on Our dataset and CMU dataset using RNN-LSTM and Kalman Filter. Red nodes define the current position, and blue nodes define the prediction obtained by the corresponding methods.

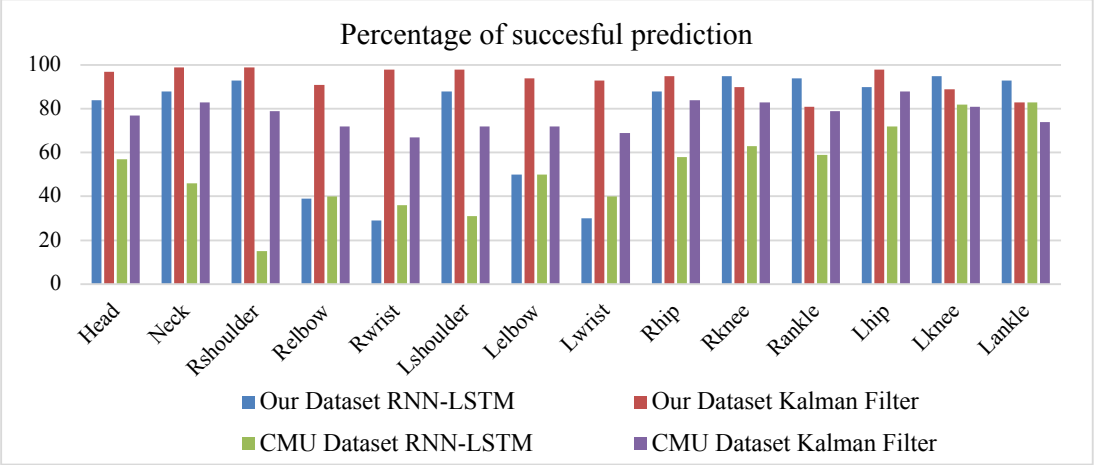


Figure 2.8: Evaluation distance percentage lower than 1.8% of the diagonal frame length in pixels.

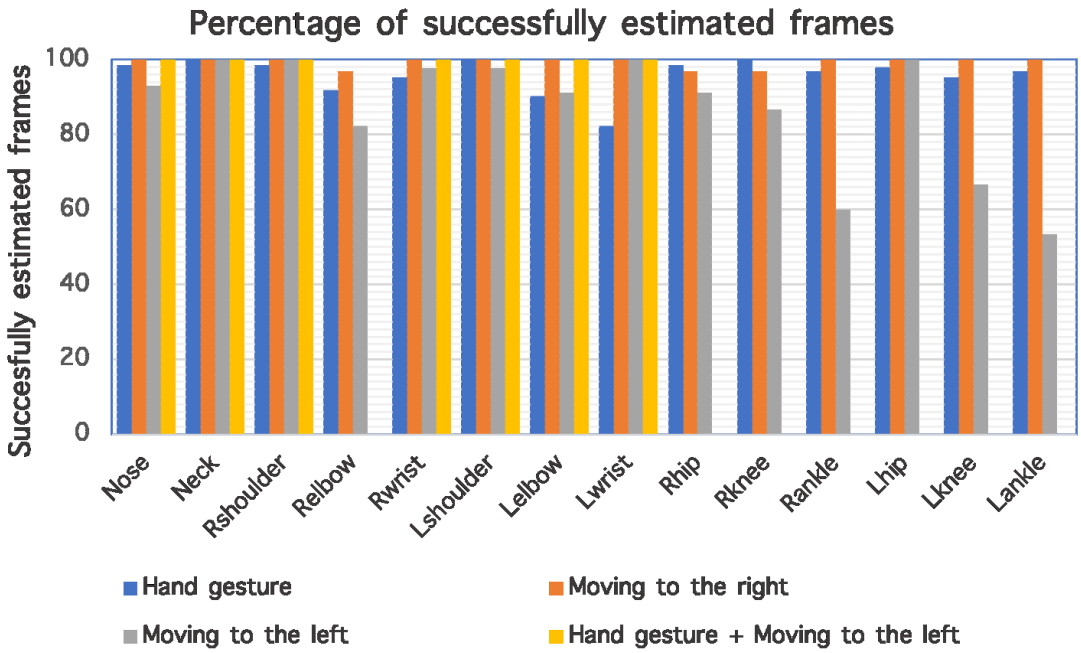


Figure 2.9: Evaluation of predicted results obtained from Kalman Filter prediction result by each node and motions based on the percentage of the value lower than 1.8%.

Table 2.1: Evaluation of the experiment results by RNN-LSTM and Kalman FIlter on Our dataset and CMU dataset.

Method	**Dataset**	**Succesful Prediction (%)**	**Error Average (pixels)**	**Error Median (pixels)**
RNN-LSTM	Ours	75.4 ± 24.9	33.4 ± 9.9	17.6 ± 18.0
	CMU	52.3 ± 18.7	32.4 ± 11.4	7.2 ± 5.4
Kalman Filter	Ours	**93.2 ± 5.6**	7.7 ± 2.3	5.6 ± 1.2
	CMU	77.1 ± 6.0	**5.2 ± 0.8**	**3.8 ± 0.8**

Kalman filter shows a closer error average and error median, which indicates that the results that the Kalman Filter has obtained are generally more comparable to the ground truth but not in the distance range of the successful prediction. However, when we see the result distribution from **Fig. 2.8**, which shows the percentage of successful prediction from accumulative frames in every key point to compare the method and dataset result, RNN-LSTM shows better results on knees and ankles on our dataset. At the same time, the other prediction results are various. For example, RNN-LSTM shows complications in remembering the data on the elbow and wrist key points since these key points are the parts of the human body that move more than others. **Fig. 2.9** shows the evaluation result based on the satisfiable result percentage on all frames based on the movement comparison by Kalman Filter, where the prediction result distribution on the moving to the right side motion is satisfiable with the most negligible value of 97% of the predictions are in the range of the correct prediction. However, in the movement of hand gesture + moving to the left on hips, knees, and ankles nodes, the prediction distribution result shows none of the results are in the range of the correct prediction. While the other motion, the prediction results are varied, with the most negligible value of 80% of the projections in the accurate forecast except for the ankles and knees are varied around 60% of the prediction results are in the range of the correct prediction.

2.5 Summary

We have proposed a system to predict human motion with Kalman Filter and RNN-LSTM using an RGB camera for one second forward. The actions of hand gestures, sideways movement, and simple walking are included in the sample video. Based on the prediction result, most of the predicted key points are close to the ground truth. The validity of the RGB-based method in the simple human motion study has been confirmed. These results concluded that this is an essential step to comprehending a more advanced method for more complex human motion. As for future works, the data has to be normalized since it has spike movement from the feature extraction method, making the prediction method challenging to predict.

Chapter 3

Time Series Self-Attention 2D Human Motion Forecasting

3.1 Introduction

With the development of the autonomous system such as auto-driving cars and auto-pilot robots that can be utilized in many ways [31, 32, 33], the safety system prevents the device or vehicle from crashing unexpectedly into other objects like humans. Such a system in real life is still under development, but some systems using sensors and cameras on devices are implemented on electric vehicles or non-electric vehicles. With more advanced technology, this system can be expanded to the level of knowing the behavior of the environment. Moreover, several applications can be utilized by knowing human motion forecasting. For example, human motion forecasting can help the human motion tracking model for better accuracy [34]. It can be used in the locomotive syndrome disorder evaluation to prevent humans from falling or any self-accident that might happen, as well as gait recognition, to identify patterns during walking [35].

Human motion forecasting has been retracted attention for advancing methods, strategies, and results. Several types of research, such as using recurrent neural networks, gated recurrent units (GRU), long short-term memory (LSTM), and Transformer Networks, are conducted in many ways for this problem [6, 7, 17, 36, 37]. However, the inputs and outputs they generate are based on data obtained from 3D motion capture [8, 9]. This is not directly applicable to the real world when using the 2D input image generated by the pose estimation on an RGB camera. Furthermore, this work is

challenging considering that human behaviors are dynamic based on the person. With this in mind, the scope of our research should be shrinker by uncomplicated motions provided in the dataset.

Inspired by the natural language processing (NLP) problem, the attention scheme has been showing excellent results in understanding the context and connection between words in sentences [38]. This attention scheme seems to be applicable in the sequence-to-sequence time series data, which has a similar conception to the words in natural language processing. When in the NLP task, the word vector is taken as the input on each sentence, as for the time-series data, a single time point data is taken like the word in the NLP task.

In this research, the author proposes the core of behavioral human motion forecasting as a step to realizing and advancing behavior-based knowledge systems for autonomous systems. In order to result in the calculated future movement of the human, the input of image sequences with the human body pose feature is required. The author constructed the human motion forecasting algorithm by using the attention-based method since it shows promising results in many ways of usage and application [12, 13, 14, 16, 38, 39, 40].

The experiment is conducted using the Human3.6M dataset as the primary dataset and the 3DPW dataset as the secondary dataset. In **Fig. 3.1**, samples of the testing sequence "sitting" and "direction" motions have been tested by the model to obtain the 2D visualization for comparison with the ground truth. The Human3.6M dataset provides the human body's key points in every frame used for the proposed model input. In this case, the method does not need to determine the input length since the data has the same size in each frame. The input data is already a numerical value. Unlike the word in the NLP task, the proposed method does not perform the embedding that converts the input data into a numerical feature vector. Instead, the process of positional encoding becomes extensive for understanding the key point connections by frames. Thus, this research proposes positional encoding based on the frame with the transformer encoder-based method to predict human motion.

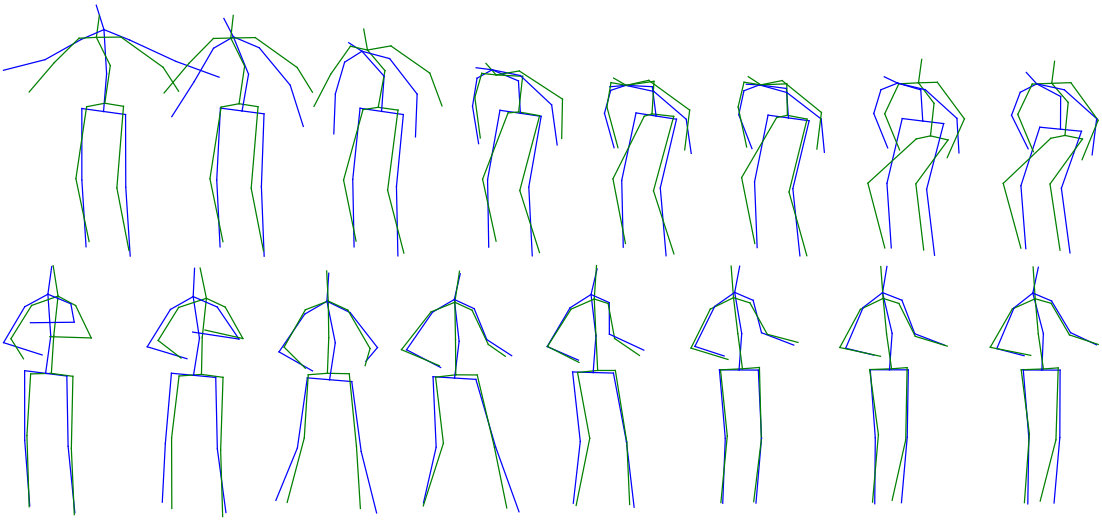


Figure 3.1: Pose prediction sequences of "Sitting" and "Direction" motions with a time-series self-attention network. The blue line is the ground truth, while the green line is the prediction result by the model.

3.2 Related Works

3.2.1 Human Motion Forecasting

3.2.1.1 3D Human Motion Forecasting

As the 3D human motion prediction baseline, the Encoder-Recurrent-Decoder (ERD) model was introduced in 2015 [5]. This research uses the recurrent neural network (RNN) architecture that absorbs nonlinear encoder and decoder networks before and after recurrent layers as the model's input. The Human3.6M dataset [41] is used and expected to obtain the short-term human motion prediction with 80, 160, 240, 320, 400, 480, and 560 milliseconds. As a result, the three layers of LSTM obtained the best result compared to ERD, Conditional Restricted Boltzmann Machines (CRBMs) model, Gaussian Process Dynamic Model (GPDM) model, and the nearest neighbor N-gram model (NGRAM) based on the Euclidean distance loss.

Approaches from different methods and strategies have been broadly applied to solve the problem of human motion prediction. One approach uses the state-of-the-art recurrent neural network (RNN) model [17], which compares several methods as evaluation from other techniques, including LSTM. The goal is to learn time-dependent representations that perform tasks such as short-term motion prediction and long-term

human motion prediction. The seq2seq, the encoding and decoding architecture, is used with the sampling-based loss for the long-term motion prediction [42]. The research also employs the residual network to help the model to incorporate prior knowledge about the statistics of human motion.

More methods were published for this problem as it became popular among autonomous system researchers and developers. Different ways to predict better and to compute reliable standard of error are developed with the following competing results [6, 7, 36, 43]. The mean per joint position error (MPJPE) is a reliable evaluation metric to show the distance between the ground truth and the prediction. One of the most recent human motion forecasting approaches is the Space-Time-Separable Graph Convolutional Network (STS-GCN) [8], which also evaluated on the archive of motion capture as surface shapes (AMASS) dataset [44] and 3D pose in the wild (3DPW) dataset [45]. They also used MPJPE to evaluate the distance between ground truth and the prediction result, commonly used to assess the human pose estimation. Following the success of STS-GCN, A. Bouazizi *et al.* [9] conducted the short and long-term human motion prediction using multilayer perceptron (MLP) architecture solely and achieved the best performance as of now.

As a result, many improvements have been made to this problem with more datasets and cases. Even though 3D human motion forecasting has become an important issue to solve, getting the data of 3D joints in the real world is a different problem. Coming up with the idea of real-time human motion forecasting with an RGB camera, the 2D interpretation of data is unavoidable, which became the reason for conducting the 2D human motion forecasting research. In this research, the author proposed a new baseline for 2D human motion forecasting with Transformer-based architecture.

3.2.1.2 2D Human Motion Forecasting

One of the similar ideas of our research, single human motion forecasting, is conducted by predicting the human motion with 3D poses in the wild (3DPW) dataset [45] for the 3D input dataset and Posetrack [11] for the 2D input dataset. However, due to the lack of the ground truth 2D input dataset, one could not compare using the Posetrack dataset. Therefore, the author considered using the Human3.6M dataset and 3DPW dataset with 2D input as it provides more data and is broadly used in the human motion forecasting problem.

3.2.1.3 Image Synthesis-based Forecasting

Predicting the object not only for humans is also necessary, considering that humans are not the only object that can move in the real world. An approach to predict pixel space for the following sequence of image features has been conducted [10]. It uses the convolutional neural network (CNN) to generate future frames from the input sequences. The model can predict two frames with a structured similarity index to measure the similarity between the prediction and the ground truth images and sharpness evaluation to measure the loss of sharpness between the true frame and the prediction. Compared to our idea, they presented the method for a video not only with a specific object like in our research. However, developing this video prediction for the future image is exciting and promising to be the next step of our research by expanding the object not only for human motion.

3.2.2 Transformer Network for Time Series Problem

Since the attention-based method was introduced, the application of the attention-based method also broadens up not only used in NLP problems but also in time-series data prediction and classification tasks like image and object recognition. It has been proved to give a slight to significant improvement in the results. As for the example usage of the transformer network with time-series data, research to forecast the influenza prevalence case has been conducted [16]. Compared with the other time-series method like LSTM and seq2seq, which uses attention, the transformer model showed performance improvement in Pearson correlation and root mean square error. As a result of this research, it can learn complex dependencies of various lengths from time-series data. Following the success of the transformer network for the time-series problem, the author considered using the transformer network for a more complex time-series problem like human motion forecasting.

3.3 Preliminaries

3.3.1 Dataset

In other related research, the Human3.6M dataset is used in all the studies. This dataset is broadly used in human pose-related research such as the human pose estima-

tion [46, 47]. It was considering the gestures and features of the human body that are nicely provided by the Human3.6M dataset [41]. This dataset contains 17 scenarios by 11 subjects taken from 4 different angles with high-resolution 50 fps video. The Human3.6M dataset provided the scenarios in the studio-like capture with no interaction with another object. 3DPW dataset provided the real-life captured dataset with multiple human features in the frame[45]. As well as the Human3.6M dataset, the 3DPW dataset provided the 2D pose annotation data, which is suitable for this research.

3.3.2 Multi-Head Attention

Multi-head attention is a module for attention mechanisms that run through an attention mechanism several times in parallel [12]. Multiple attention heads allow for attending to parts of the sequence differently.

3.3.3 Transformer Networks

Transformer networks were proposed by Vaswani *et al.* for the machine translation tasks [12]. It became state of the art for the NLP problems with the large scale of usage. The usage of the transformer networks was also broadened to solve the problem of the computer vision tasks by Vision-Transformer, which is a simplified model of transformer networks. In the transformer networks used in the NLP, the attention mechanism tried to compute the relation between words in the sentence to be analyzed. In the case of the vision transformer, it tried to calculate the different parts of the image. By splitting the image into fixed-size patches, linear embedding, adding positional embeddings, and feeding the resulting sequence of vectors to a standard transformer encoder, then for the classification, the standard MLP head is used.

3.4 Proposed Method

In this section, the author describes the detail of the proposed method, which estimates the human body's key points (from now on called key points) in future T_P frames from key points from past T_Q frames by using the time series self-attention model. The overview of the method's flow is described in **Fig. 3.2**.

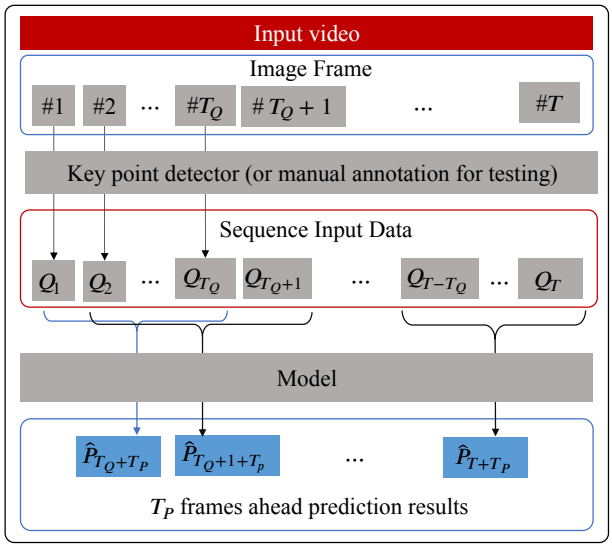


Figure 3.2: Overview of the prediction flow. T_Q number of input frames is defined and predict frame T_P ahead of prediction. While in this research, the T_P is defined as 10 for 400ms and 25 for 1000ms prediction task.

3.4.1 Frame-by-frame 2D Pose Estimation

Requiring the feature of key points from the input video for the processing, the 2D pose estimation is needed to interpret the image into the coordinate location of the key points. In this research, the author used OpenPose as the standard method to estimate the 2D human pose in an image. However, other methods are likewise applicable for this task, such as XNect [48], ViTPose [14], and ViTPose V2 [15].

Most 2D humans pose estimation methods on video images estimate the joint coordinates frame-by-frame, and key points are expected to be the 2D data consisting of x- and y-coordinate in the video frame. The estimated key points in the i-th video frame $\mathbf{Q}_i \in \mathbb{R}^{2N}$ consists of x- and y-coordinates of N points. When the number of given frames is T_Q, $\mathbf{Q} = \{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_{T_Q}\}$ is obtained as the sequence of feature vectors.

Notably, $\mathbf{Q}_i$ is flattened to a 1-dimensional vector, and its components are normalized by scaling between 0 and 1.

3.4.2 Time Series Self-Attention

This section describes the self-attention-based method to forecast key points in future T_P frames from the input sequence $\mathbf{Q}$. Transformer models apply the embedding to

transform the raw data to the 1-D vector. However, the input sequence in this research has the shape of a 1-D vector, which no longer needs the embedding transformation. The positional encoding is applied to the input data's positional index before feeding the input data to the transformer encoder and MLP head.

Considering that the input sequence **Q** is already a vector with numerical numbers, the embedding module is not needed in the network. Positional encoding is added to the input to capture the positional information. However, this positional encoding captures the position based on each time or frame input, as the position of the key points on each frame is fixed. With this in mind, the model only considers the frame position. The transformer encoder network computes the key points inside the frame regarding the frame position. Let t be the desired position in an input frame, $\overrightarrow{F_t} \in \mathbb{R}^d$ be its corresponding encoding, while d is the encoding dimension which in this case $d = 32$.

$$\overrightarrow{F_t}^{(i)} = \begin{cases} \sin(\omega_k \cdot f) & \text{if } i = 2k, \\ \cos(\omega_k \cdot f) & \text{if } i = 2k+1, \end{cases} \tag{3.1}$$

where

$$\omega_k = \frac{1}{10000^{\frac{2k}{d}}} \tag{3.2}$$

and k is indices containing $\{0, 1, \ldots, \frac{d}{2} - 1\}$.

As shown in **Fig. 3.3**, similar to the Transformer Encoder [12], the transformer encoder block in the proposed model includes the Multi-Head Attention, the MLP head block that contains the linear transformation in the fully-connected network, and the normalization layer applied after every block. The MLP head block contains two linear layers with a ReLU activation function after each. The model with a linear layer dimension of 1,024 or 4,096 is set to examine the best model for each motion. The expected output of the MLP head block is $\hat{\mathbf{P}} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_{T_P}\}$, where T_{P} is the length of the expected output dimension.

3.4.3 Loss Metric

Root mean square error (RMSE) is employed to evaluate the distance of estimated key points sequence $\hat{\mathbf{P}} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_{T_P}\}$ from its corresponding ground truth sequence

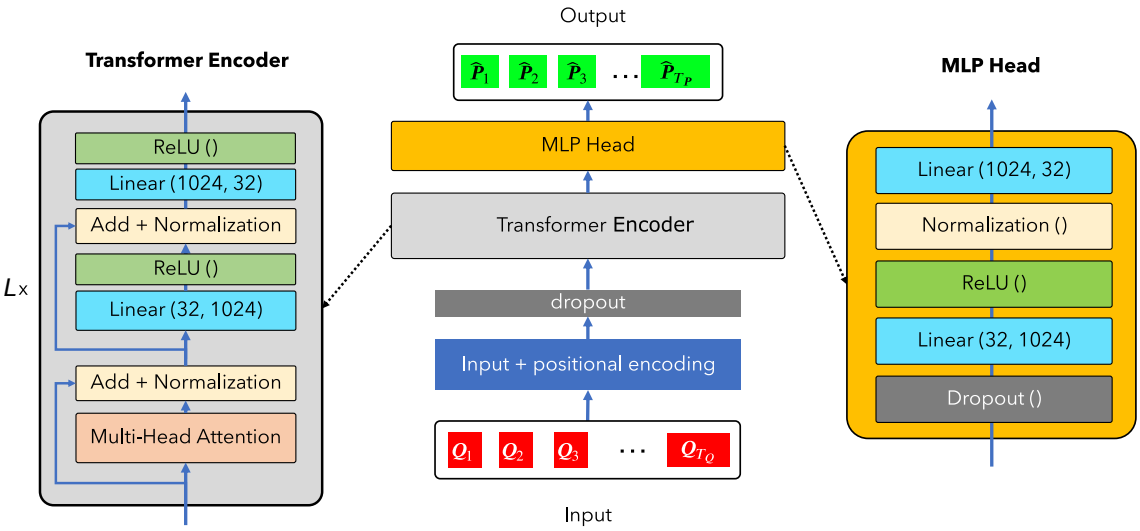


Figure 3.3: Time series self-attention network. L is the number of Transformer Encoder layers.

$\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{T_P}\}$.

$$\mathcal{L}_1 = \sqrt{\frac{1}{T_p N} \sum_{\mathbf{P}_i, \hat{\mathbf{P}}_i} \|\mathbf{P}_i - \hat{\mathbf{P}}_i\|^2} \tag{3.3}$$

Additionally, because key points in the leg and arm have more significant movement than those in the body trunk, additional errors on those points are computed.

$$\mathcal{L}_2 = \frac{1}{T_p \mathcal{N}} \sum_{\mathcal{P}_i, \hat{\mathcal{P}}_i} \|\mathcal{P}_i - \hat{\mathcal{P}}_i\|^2 \tag{3.4}$$

where $\mathcal{P}_i$ and $\hat{\mathcal{P}}_i$ is the set of left and right side of the shoulder, elbow, hand, hip, knee, and ankle key points in $\mathbf{P}_i$ and $\hat{\mathbf{P}}_i$, respectively. $\mathcal{N}$ is the number of key points in $\mathcal{P}_i$. Finally, loss function $\mathcal{L}$ is formulated as the weighted summation of $\mathcal{L}_1$ and $\mathcal{L}_2$.

$$\mathcal{L} = \mathcal{L}_1 + w\mathcal{L}_2 \tag{3.5}$$

where w is a weight parameter set to 4 in the experiment.

3.5 Experiments

3.5.1 Dataset

The Human3.6M dataset has been used with the 2D data pose representation from subjects and motions in this research. The interpretation of 2D data from the Human3.6M dataset is provided by human pose estimation research [49]. The data is divided based on the training and testing. The training data consists of subject numbers 1, 6, 7, 8, 9, and 11, and the testing data consists the subject number 5. This setup is referred to as the related experiment [17]. However, the evaluation methods and metrics apply to this research. As for the 3DPW dataset, the author follows the setting of the training, validation, and testing sets originally from the dataset itself [45].

For the evaluation of robustness against the input sequence **Q**, the testing accuracy is compared to ground truth provided by the Human3.6M dataset and OpenPose estimation. In the ground truth testing, the ground truth key points are used for both the input sequence **Q** and output sequence **P**. In the OpenPose testing, the input sequence **Q** is given by OpenPose. Furthermore, only the ground truth output sequence **P** is used for calculating the evaluation metrics.

3.5.2 Experimental Setup

The author set up the model with several parameters, including the number of transformer encoder layers $L = 6$, dropout value of 0.5, batch size of 64, and dimension on the linear layer of 1024 and 4096 to compare the result repeated in 5,000 epochs.

The proposed model is trained and evaluated using a sliding window strategy in the time axis. As shown in **Fig. 3.4**, one window consists of T_Q input frames and T_P output frames.

In this research, the input data is determined to be 25 frames. The ground truth data is the subsequent 10 frames for the short-term prediction or 25 frames for the long-term prediction consecutively with one frame shifted.

The optimal batch size is 64, and the optimal number of transformer encoder layers is $L = 6$ for our environment. The model took more memory and made the training process heavier if the batch size exceeded 64.

The training process is done by specific single motions one by one with a learning rate of 0.001 with the Adam optimizer. Then, the sequence data with 25 frames is

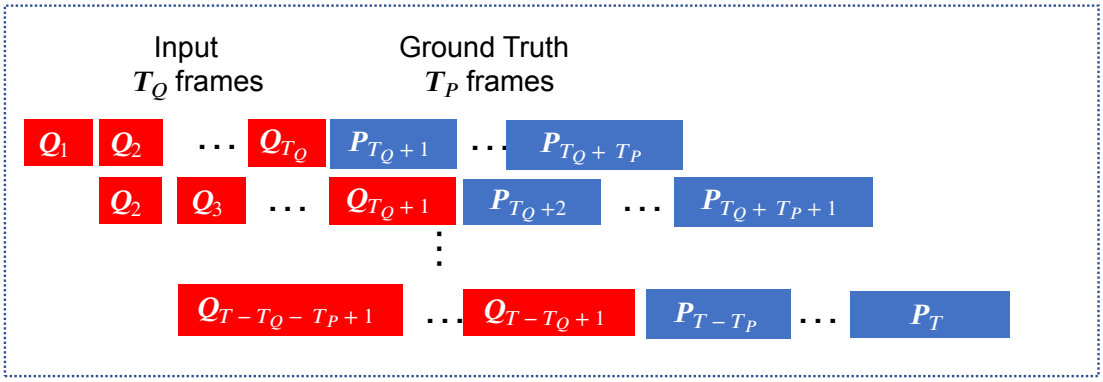


Figure 3.4: Sliding window on the input and ground truth data as the expected output. Given the source input for the method with the shape of T_Q frames and the target expected output with the shape of T_P frames. One shifting scheme as the input is used to keep the input length sufficient for the model.

defined as the input, expecting 10 frames of 400 milliseconds or 25 frames of one second in the video. The following experiments use the PyTorch environment on an NVIDIA GeForce RTX 3090 GPU.

3.5.3 Evaluation Metrics

Recent works in human pose forecasting and estimation have standardized the evaluation metrics by calculating the mean per-joint position error (MPJPE) [5, 6, 7, 8, 9, 17, 36, 43], mean per-joint velocity error (MPJVE) [49, 50] and mean per-joint localization error (MPJLE) [41]. MPJPE is calculated by computing the squared Euclidean distance between the ground truth and the prediction with respect to the treated joints. The evaluation metric of MPJPE is defined by:

$$\text{MPJPE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{P}_i - \hat{\mathbf{P}}_i\|, \tag{3.6}$$

where $\mathbf{P}_i$ and $\hat{\mathbf{P}}_i$ is the ground truth and predicted coordinates in the frame i in the N frames respectively.

The MPJVE is calculated by computing the L2-norm of motion velocity, which is the one-frame difference of coordinates between the prediction and the ground truth. The evaluation metric of MPJVE is defined by:

$$\text{MPJVE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{V}_i - \hat{\mathbf{V}}_i\|, \tag{3.7}$$

where $\mathbf{V}_i$ and $\hat{\mathbf{V}}_i$ are velocity from frame $i-1$ to time i, and is defined by:

$$\mathbf{V}_i = \mathbf{P}_i - \mathbf{P}_{i-1}. \tag{3.8}$$

If MPJPE is defined to calculate the distance from the prediction to the ground truth and MPJVE is to define the mean differences in each frame movement. MPJLE defines the localization of the correct key point within the tolerance value. The evaluation metric of MPJLE is defined by:

$$\text{MPJLE} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\|\mathbf{P}_i - \hat{\mathbf{P}}_i\| \geq t}, \tag{3.9}$$

where $\mathbb{1}$ is a binary step function that gives 0 if the distance value is within t and gives 1 otherwise. At the same time, t is the integral tolerance value in an interval. In this case, the author defined $t = [0, 200]$. One can obtain an estimate of the average error. In the same way, the mean average precision measure the performance of a classifier [41].

Qualitatively, the skeleton data is shown to visualize the difference between the prediction and the ground truth. Additionally, to know the computational time of the proposed model on each frame to produce the prediction value, the average time taken by motion is calculated. This is necessary considering predicting human activity for the role in real life.

3.6 Results

As described in Section 3.5.3 for the 2D human motion forecasting task, the prediction result is quantitatively evaluated by the evaluation methods to show that the prediction is correct or sufficient based on the distance of the prediction result and the corresponding ground truth data. As for the comparison in the 2D human motion prediction, the state-of-the-art method, such as LSTM and GRU, is used to compare the validity of our method. Qualitatively, the prediction result could be seen in frames in the video comparing the ground truth and prediction key point skeleton movement.

3.6.1 Model Training

In this section, the author describes the evaluation of the model in the training phase. To show the model validity of learning the certain problem in the data, evaluation

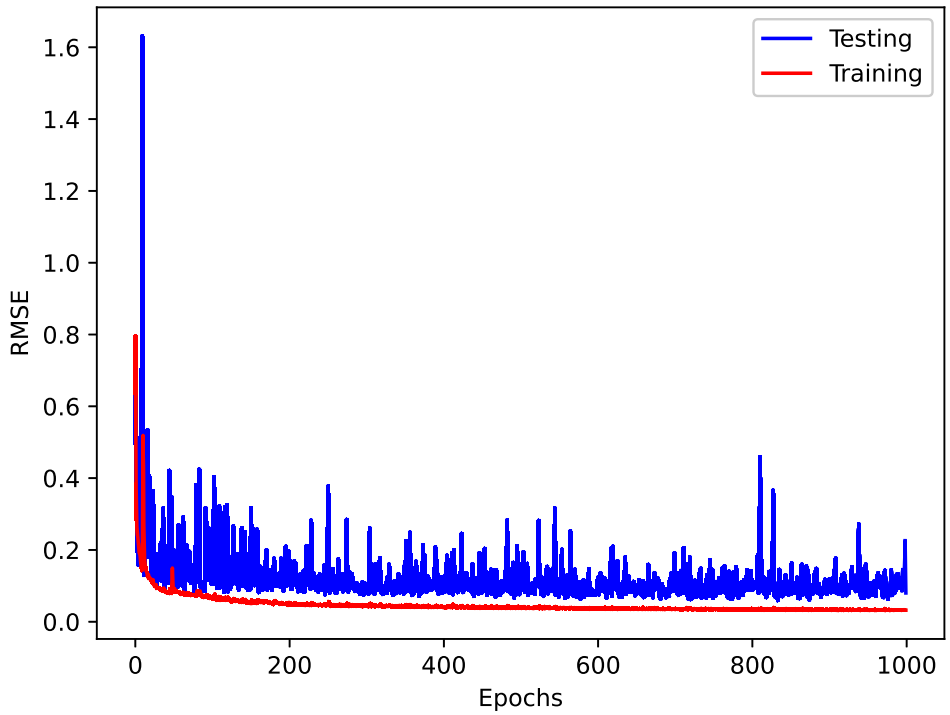


Figure 3.5: Our model is trained for 1000 epochs using Walking motion data for the long-term prediction task.

over the training phase is needed. **Fig. 3.5** shows the training phase on the Walking motion using the Human3.6M dataset. The RMSE with additional weight is computed to evaluate the distance from the prediction results to the corresponding ground truth.

3.6.2 Quantitative Evalulation

The results are separated regarding the dataset and evaluation metric for quantitative evaluation.

3.6.2.1 Human3.6M Dataset

Comparison based on MPJPE. In this part, the prediction results are evaluated based on the MPJPE score to see the distances from the prediction to the ground truth. **Table. 3.1** shows the evaluation result based on the MPJPE for the Hu-

Table 3.1: MPJPE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data for the short-term prediction task.

Motion	400 msec			
	Ours (1024)	Ours (4096)	LSTM	GRU
Walking	**13.32**	49.19	14.38	13.90
Eating	**8.51**	12.57	13.14	14.48
Smoking	9.91	**7.51**	16.47	17.40
Discussion	11.20	**8.55**	20.62	20.75
Direction	28.09	**7.63**	21.55	21.77
Greeting	15.26	**14.38**	36.67	35.42
Phoning	60.26	**11.20**	19.00	18.89
Waiting	37.08	70.17	26.98	**25.84**
Walking Dog	**10.97**	29.93	41.19	39.60
Walking Together	**9.09**	32.48	45.42	46.32
Posing	47.98	**8.19**	30.76	27.86
Sitting	**9.99**	19.28	18.91	21.74
Sitting Down	**8.37**	23.09	27.95	28.84
Taking Photo	59.11	119.23	36.58	**34.52**
Average	**23.51**	29.53	26.40	26.24

man3.6M dataset comparing the time series self-attention method (hereinafter called Ours), RNN-LSTM, and RNN-GRU in the short-term prediction task. Our method is performed with two linear dimension values, 1024 and 4096, to see the difference in which dimension gives the best results.

The best result can be seen in the bold highlighted value. Our method with 1024 linear dimensions obtained the best result compared to the other method in Walking, Eating, Walking with Dog, Walking Together, Sitting, and Sitting Down motions. This means 6 over 14 motions, or 42.8 percent of the motions with a range of the MPJPE score around 8 to 14 pixels, within the best-predicted motions with our method with 1024. At the same time, Our method with a linear dimension of 4096 obtained the best result on Smoking, Discussion, Direction, Greeting, Phoning, and Posing motions. Similar to the linear 1024, this model also obtained 6 over 14 motions or 42.8 percent of the motions with a range of error of around 7 to 15 pixels. However, our method

Table 3.2: MPJPE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data for the long-term prediction task.

Motion	1000 msec			
	Ours (1024)	Ours (4096)	LSTM	GRU
Walking	14.43	**12.03**	12.11	12.74
Eating	**9.11**	10.11	14.09	13.02
Smoking	**7.11**	8.27	15.21	16.53
Discussion	**8.34**	14.30	21.21	19.48
Direction	**5.76**	8.17	20.35	20.56
Greeting	12.90	**10.67**	33.92	32.56
Phoning	**9.11**	10.85	18.41	16.93
Waiting	**6.37**	7.15	24.46	22.37
Walking Dog	**10.98**	13.1	39.99	38.82
Walking Together	**8.15**	8.20	40.82	29.62
Posing	15.46	**8.41**	26.38	25.96
Sitting	**9.61**	17.05	20.73	19.48
Sitting Down	9.18	**7.59**	31.25	25.74
Taking Photo	17.69	**12.53**	40.70	29.50
Average	**10.30**	10.60	25.69	23.09

failed to predict the Waiting and Taking Photo motions with quite a significant MPJPE score of around 37 to 120 pixels. Instead of our model, GRU could predict the motion better with an MPJPE score of 25.84 pixels on Waiting and 34.52 pixels on Taking Photo motion. Indeed 25 to 35 pixels is quite a significant score compared to the other best results. With regard to this MPJPE score, Waiting and Taking Photo motions are considered the most challenging case, as these motions are aperiodic motions which means that these motions are not recurring at regular intervals.

In contrast, Our model with 1024 linear dimensions obtained bad results on Direction, Phoning, and Posing motions compared to the other models with a range of 28 to 61 pixels. On the other hand, our model with 4096 linear dimensions obtained bad results on Walking, Walking Dog, and Walking Together compared to the other models with a range of 29 to 50 pixels.

According to these results, our model with a linear dimension of 1024 could not

predict the motions with not so much movement but good with the recurrent motions. In contrast, our model with a linear dimension of 4096 obtained bad results on the motions with a lot of movement but could predict well on the motions with not so much movement. This means that more dimension linear gives the effect of more possibilities and a better understanding of passively moving actions, but many possibilities value also reduces the context understanding of the actively moving actions. On average, our model, with 1024 linear dimensions, obtained the best result compared to other models with 23.51 pixels by MPJPE. However, the average values were not very different from the other models due to some significant errors.

Table. 3.2 shows the MPJPE evaluation result of the long-term human motion prediction task on the Human3.6M dataset. While on the short-term prediction showing, the prediction results varied based on the motions. Long-term prediction results show consistency over all motions in the range of 5 to 18 pixels by our models. On average, our model with a linear dimension of 1024 obtained the best result of 10.30 pixels by MPJPE. Followed by a very small difference in our model with a linear dimension of 4096 by 10.60 pixels on average of MPJPE. The best prediction results were obtained on the Direction motion with 5.76 pixels by MPJPE score. There is not so much difference in our model with a linear dimension of 1024 and 4096 in the long-term prediction task, which indicates the model could predict well in any case of the motions. Our model outperformed the RNN-based method with quite a significant MPJPE score. The motion-wise comparison shows our model with a linear dimension of 1024 is a bit bigger than the other models, while the other prediction results on other motions are smaller than the RNN-based models.

Given a clean annotation over the key points of the human body poses might be unrealistic in the real world yet. Due to this reason, the pose estimation key point detection is used to extract the human body pose features as the testing data for our trained model. In this case, OpenPose is used to generate the human body pose features as it is currently one of the most used pose estimation methods. The features with the noise of incorrect estimation could be one barrier for the model to predict future human motion. With this in mind, our model could be evaluated with the noisy data obtained from the real-time human pose estimation.

Table. 3.3 shows the MPJPE score evaluation on the data generated by OpenPose as the pose estimation on the Human3.6M dataset. For the short-term prediction task,

Table 3.3: MPJPE of 2D joint positions in pixel on the Human3.6M dataset using human pose estimation by OpenPose as the testing data.

Forecasting ($msec$)	400		1000	
Model	**Ours (1024)**	Ours (4096)	**Ours (1024)**	Ours (4096)
Walking	**29.10**	52.38	**25.82**	36.00
Eating	**22.51**	26.73	24.69	**16.79**
Smoking	50.50	**38.53**	21.12	**19.15**
Discussion	26.75	**21.02**	**21.66**	25.94
Direction	26.24	**19.40**	**17.43**	19.53
Greeting	29.11	**25.09**	23.18	**16.80**
Phoning	61.17	**24.84**	22.47	**21.39**
Waiting	**35.05**	70.17	**23.96**	25.34
Walking Dog	**25.70**	48.98	**23.27**	83.43
Walking Together	48.62	**39.19**	**23.01**	23.51
Posing	58.16	**23.84**	27.80	**23.91**
Sitting	**17.27**	18.91	22.29	**20.87**
Sitting Down	**22.97**	29.18	21.05	**17.28**
Taking Photo	**78.84**	119.23	23.80	**22.73**
Average	**37.71**	39.90	**22.97**	26.62

our model with 1024 linear dimensions obtained the best result on average, with a small difference over our model with a linear dimension of 4096. As the comparison regarding the motions, our model with a linear dimension of 1024 obtained quite consistent MPJPE scores with a range of 17 to 35 pixels on most of the motions. Expect the Smoking, Phoning, Walking Together, Posing, and Taking Photo motions. Comparing the prediction results on the ground truth annotation testing, the Phoning, Posing, and Taking Photo motions also obtained bad results. The difference between the result of the ground truth annotation testing and the OpenPose extracted features testing is 14.49 pixels, which indicates the prediction result by using the OpenPose generated features is still reliable to get the future of human body motions.

Similarly, our model with a linear dimension of 4096 obtained almost the same pattern as the evaluation result on the ground truth annotation data testing with only 10.29 pixels differences between the evaluation result of the OpenPose generated features testing.

Furthermore, on the long-term prediction task, our models predict better regarding the MPJPE evaluation score. The results of both models consistently predict human motion in a range of 16 to 36 pixels of MPJPE score. However, our model with a linear dimension of 4096 failed to predict future human motion with the MPJPE score of 83.43 pixels. As a result, our model with 1024 and 4096 linear dimensions can predict future human motion on average. Even though the MPJPE score is quite big, the prediction result is still reliable for predicting the human location movement, while on the other hand, the model could not predict to visualize the human body pose well.

Additionally, **Fig. 3.6** shows the MPJPE evaluation on the key points with respect to the motions. The ankle's key points in the Walking motion obtained the highest MPJPE score compared to the difference with another key point in the other motions. Since the ankles are the most moving key points in the Walking motion, followed by the hands and elbows. This could be explained more clearly in the qualitative evaluation in Section 3.6.3.

For more details, **Fig. 3.7** shows the comparison between the testing result on the data obtained by OpenPose and the ground truth testing in each motion.

Figure 3.8 shows the comparison of the MPJPE distance trajectories on the OpenPose testing and ground truth testing by each frame in the "Walking" motion.

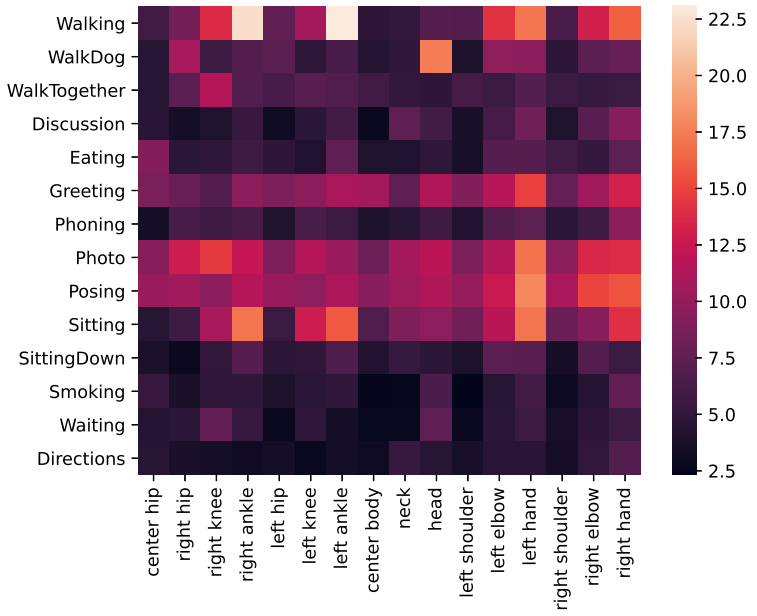


Figure 3.6: MPJPE-based evaluation on the key points and motions for the long-term prediction task using our model with a linear dimension of 1024. The heatmap color contains the MPJPE score based on the color scale on the right side.

Comparison based on MPJVE. In this part, the evaluation based on MPJVE is described in detail to compare the smoothness of the movement prediction results obtained by the methods.

Table. 3.4 shows the score of MPJVE evaluation obtained by our model with a linear dimension of 1024 and 4096, RNN-LSTM, and RNN-GRU for the short-term prediction task. Based on the motions, empirically, the motion with more movement will obtain more MPJVE scores due to the changes in the movement over the frames. For example, comparing the Eating and Walking Dog motion will obtain a very different MPJVE score since most of the Eating motion movements stayed on the same spot while the only movement was at the hands and some gestures of the torso. On average, our model with a linear dimension of 4096 obtained the best MPJVE score with 1.22 pixels over other methods. Our model with a linear dimension of 1024 could predict the best based on the MPJPE score, but based on the MPJVE, the prediction result is not as smooth as the other method obtained. This indicates the results obtained by this model are quite spiky, with 1.99 pixels average movement for 1 frame. At the same time, the worst result was obtained by our model with a linear dimension of 1024 on the Walking

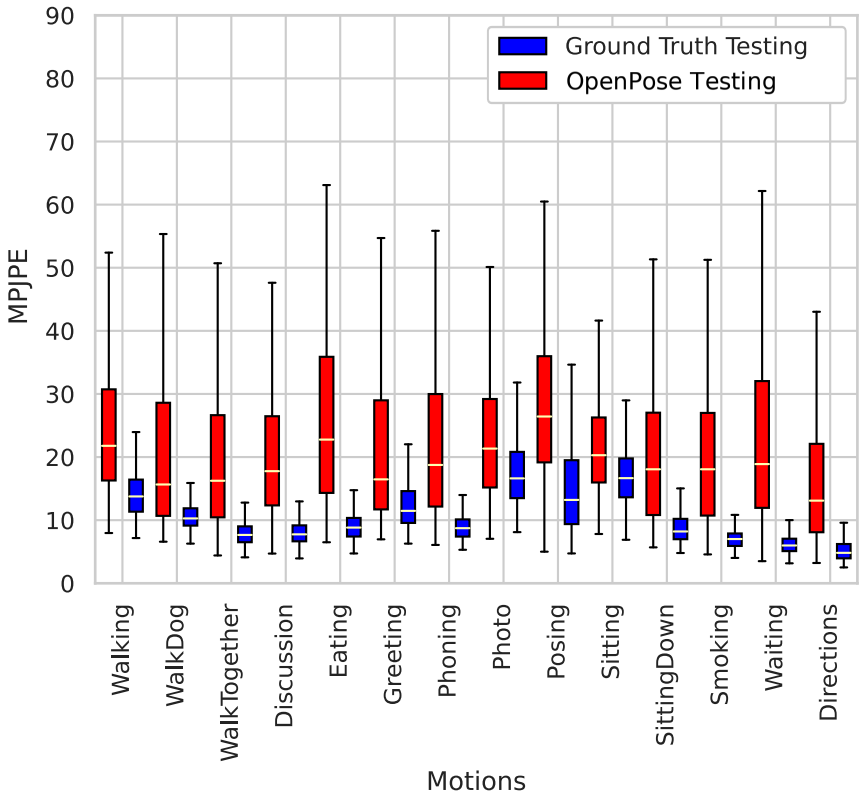


Figure 3.7: Comparison of MPJPE distance for each motion on data obtained by OpenPose and data from the dataset for testing.

Table 3.4: MPJVE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data.

Motion	400 msec			
	Ours (1024)	**Ours (4096)**	LSTM	GRU
Walking	**1.05**	1.67	1.15	1.23
Eating	**0.70**	0.73	0.97	1.00
Smoking	2.7	1.45	**0.86**	0.89
Discussion	1.34	1.28	**1.27**	1.33
Direction	1.34	**0.69**	1.15	1.23
Greeting	1.33	**1.30**	2.77	2.91
Phoning	2.38	**0.83**	1.11	1.10
Waiting	2.59	**0.93**	1.60	1.62
Walking Dog	4.81	2.79	2.51	**2.38**
Walking Together	1.28	**1.25**	2.36	2.35
Posing	1.64	**0.84**	1.58	1.60
Sitting	0.85	**0.9**	1.18	1.44
Sitting Down	2.69	1.64	1.30	**1.21**
Taking Photo	3.15	**0.73**	1.31	1.24
Average	1.99	**1.22**	1.51	1.54

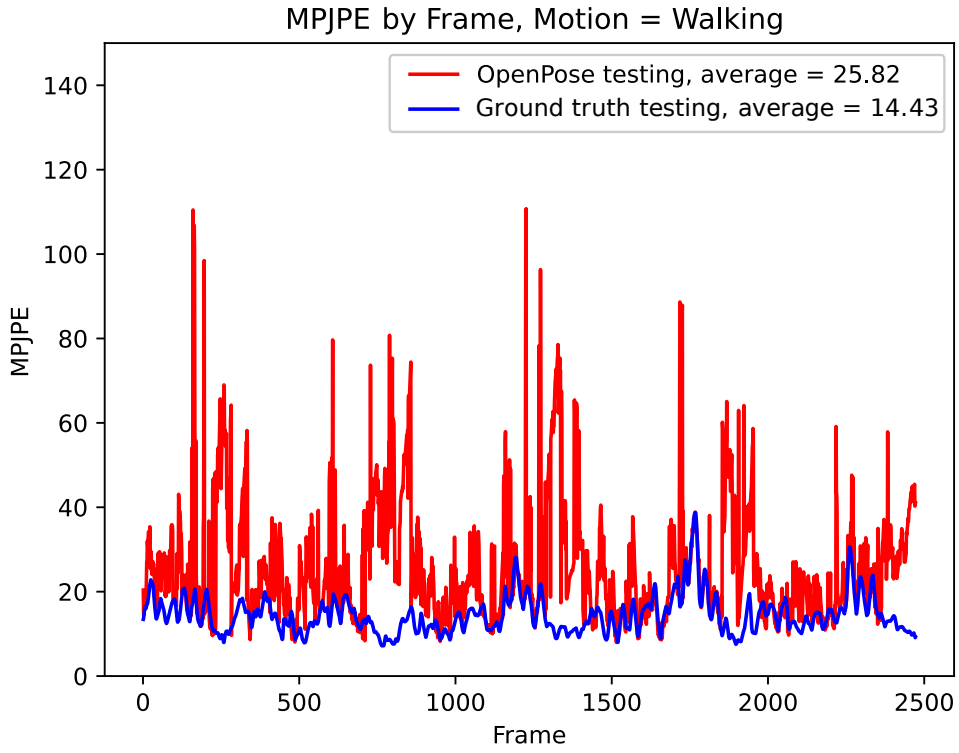


Figure 3.8: Comparison of MPJPE in long-term prediction by frame on OpenPose testing and ground truth testing with linear 1024 model.

Dog motion with 4.81 pixels average movement for 1 frame step. On the other hand, the prediction results obtained by the RNN-based models are quite consistent with the range of the MPJVE score of 0.8 to 2.5 pixels.

Table. 3.5 shows the score of MPJVE evaluation obtained by our model with a linear dimension of 1024 and 4096, RNN-LSTM, and RNN-GRU for the long-term prediction task. On average, both of our models outperformed the RNN-based method. Our model achieved the best results when using a linear dimension of 4096, and it obtained the highest MPJVE score for 11 out of 14 motions. Additionally, our model performed well for the remaining 3 out of 14 motions using a linear dimension of 1024. All the models demonstrated exceptional performance in the long-term prediction task, with an MPJPE score range of 0.6 to 2.39 pixels for a 1-frame step. Overall, our approach showed strong performance in both short-term and long-term prediction

Table 3.5: MPJVE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data.

Motion	1000 msec			
	Ours (1024)	**Ours (4096)**	LSTM	GRU
Walking	1.11	**1**	1.07	1.22
Eating	0.71	**0.66**	1.04	1.10
Smoking	0.66	**0.62**	0.84	0.88
Discussion	**1.16**	1.36	1.31	1.24
Direction	**0.72**	0.77	1.15	1.16
Greeting	1.45	**1.26**	2.39	2.67
Phoning	0.87	**0.80**	1.11	1.14
Waiting	0.98	**0.85**	1.45	1.62
Walking Dog	1.28	**1.1**	2.16	2.37
Walking Together	0.72	**0.67**	2.00	2.03
Posing	1.31	**0.91**	1.58	1.67
Sitting	1.07	**0.9**	1.41	1.29
Sitting Down	0.97	**0.76**	1.61	1.53
Taking Photo	**0.88**	0.83	1.40	1.33
Average	0.99	**0.89**	1.47	1.52

tasks. In the short-term prediction task, using a larger linear dimension resulted in better performance. In contrast, for the long-term prediction task, there was only a minimal difference in performance between using a linear dimension of 1024 and 4096, with an MPJPE score difference of only 0.1 pixels.

During the evaluation, models were compared to ground truth annotation data in testing. Additionally, models were tested on non-annotated data that had been processed with the OpenPose feature extraction method as shown in the **Table. 3.6**. The evaluation results using non-annotated data for testing yielded significantly higher average movement over 1 frame step than when ground truth annotated data was used. The average movement when using non-annotated data was 5 to 9 pixels, whereas it was only between 0.8 to 2 pixels when using annotated data. However, it's important to note that comparing the results on average may not provide an accurate representation of the performance as the different motions have distinct characteristics. For instance,

Table 3.6: MPJVE of 2D joint positions on the Human3.6M dataset using human pose estimation by OpenPose as the testing data.

Forecasting ($m$sec)	400		1000	
Model	Ours (1024)	**Ours (4096)**	**Ours (1024)**	Ours (4096)
Walking	**7.92**	9.79	**6.89**	7.82
Eating	4.87	**3.63**	**4.85**	4.92
Smoking	14.61	**5.96**	5.78	**5.59**
Discussion	7.11	**6.96**	7.50	**6.13**
Direction	5.20	**4.82**	5.29	**4.84**
Greeting	8.12	**8.07**	**1.45**	7.34
Phoning	7.35	**5.03**	5.44	**5.02**
Waiting	7.36	**0.93**	7.43	**6.03**
Walking Dog	**8.19**	11.42	**8.29**	9.50
Walking Together	24.50	**11.87**	**9.29**	9.30
Posing	6.31	**6.20**	**4.59**	6.96
Sitting	3.74	**3.44**	3.34	**2.84**
Sitting Down	**5.72**	5.78	5.86	**5.33**
Taking Photo	11.98	**0.73**	**4.34**	5.05
Average	8.79	**6.05**	**5.74**	6.19

in the case of the "Walking Together" motion, the MPJVE score is relatively higher than other motions because the subject in this motion has more movement than the other motions. This also suggests that the more movement occurs within a frame, the harder it is for the model to generate a smooth movement prediction over multiple frames.

Comparison based on MPJLE. In this part, the author describes the evaluation based on the MPJLE metric to show the localization of the prediction results by the Threshold at the tolerance t as shown on **Table. 3.7**. The tolerance t is an interval from 0 to 200 pixels. When the prediction result is above the threshold of tolerance, the result is considered an error. The performance of the methods is evaluated at different thresholds (5, 10, 20, 50, 75, 100, 150, and 200). The best performance for each threshold is highlighted in bold. Overall, it appears that the "Ours (4096)" method performs the best, having the lowest MPJLE among all methods at most thresholds.

Table 3.7: MPJLE of 2D joint positions on the Human3.6M dataset in long-term prediction task.

Methods	Threshold@t							
	5	**10**	**20**	**50**	**75**	**100**	**150**	**200**
LSTM	1.000	1.000	1.000	0.884	0.654	0.446	0.225	0.121
GRU	1.000	1.000	1.000	0.854	0.627	0.435	**0.193**	**0.092**
Ours (1024)	1.000	1.000	**0.980**	0.662	0.505	0.387	0.226	0.150
Ours (4096)	1.000	1.000	1.000	**0.635**	**0.457**	**0.373**	0.218	0.141

Although the performance of our model with a linear dimension of 1024 is similar to that of our model with 4096 linear dimensions, at a threshold of 20 pixels, our model with a linear dimension of 1024 only slightly outperforms the latter, with a difference of just 0.02%.

3.6.2.2 3DPW dataset

In this section, the author describes the result of evaluating the human motion prediction task using the 3DPW dataset by time series self-attention, RNN-LSTM, and RNN-GRU. To evaluate the effectiveness of the methods when using different datasets, an evaluation was performed using the 3DPW dataset and based on the MPJPE metric as shown on **Table. 3.8**. While the Human3.6M dataset collected data at the indoor studio, the 3DPW dataset collected the outdoor data with unscripted motions. Hence, overall the MPJPE scores using the 3DPW dataset are very high. This means that the models could not predict future human motion. However, based on the evaluation result of MPJPE scores, the 1-layer RNN-GRU obtained the best average prediction result with 236.31 pixels. Our model could only be performed better than the RNN-based method on the left ankle key point with a very small difference compared to the RNN-GRU.

3.6.2.3 Computational Time

Additionally, the author acknowledged the average computation time by the model to predict a frame of motion as an important note, considering that in the real world, one needs optimal computation to get a real-time prediction. As shown on **Table 3.9**, the

Table 3.8: MPJPE of 2D joint positions in pixel on the 3DPW dataset using the real position data as the testing data. The author compared our method with LSTM and GRU models for Layer $L = 1$, 2, and 3 with the 3DPW dataset.

Keypoints	Our Method			LSTM			**GRU**		
	$L = 1$	$L = 2$	$L = 3$	$L = 1$	$L = 2$	$L = 3$	**L=1**	$L = 2$	$L = 3$
Head	300.66	304.49	305.94	291.18	292.15	287.65	287.09	304.28	**283.75**
Neck	210.09	211.55	243.07	198.10	216.20	215.61	**193.15**	210.02	205.77
Right Shoulder	230.72	219.34	254.70	208.71	223.54	224.08	**200.17**	218.20	216.35
Right Elbow	272.43	258.64	289.76	243.50	260.86	259.78	**236.62**	242.08	251.56
Right Arm	284.46	285.97	304.73	268.98	280.76	298.59	268.57	**260.71**	277.89
Left Shoulder	225.60	226.63	252.11	209.95	228.16	232.55	**206.32**	222.86	216.89
Left Elbow	262.93	251.46	279.75	252.70	260.94	275.28	**244.77**	259.65	249.14
Left Arm	278.76	269.65	290.99	278.02	276.12	302.90	270.06	278.03	**268.75**
Right Hip	200.78	207.60	238.80	189.49	207.88	207.93	**187.28**	206.86	203.76
Right Knee	201.87	212.06	246.68	205.82	213.77	215.71	**198.18**	210.54	214.54
Right Ankle	232.04	243.93	269.89	237.51	245.93	244.32	**227.35**	239.76	244.73
Left Hip	212.96	210.56	238.46	**191.73**	213.81	213.65	193.32	211.36	206.62
Left Knee	215.32	218.37	243.52	208.82	230.06	220.52	**207.50**	222.62	216.19
Left Ankle	**237.89**	245.12	272.82	244.08	266.75	252.06	239.56	257.20	246.73
Right Eye	297.97	315.54	316.55	316.22	305.46	**292.54**	317.87	335.00	311.33
Left Eye	328.79	347.23	318.94	310.22	**289.83**	309.99	303.15	319.58	303.48
Average	249.58	251.76	272.92	240.94	250.77	254.32	**236.31**	249.92	244.84

Table 3.9: The average computation time in seconds for a frame to be predicted in the testing by the model.

Motion	Processing time (*sec*)
Walking	2.44
Walking Dog	2.42
Walking Together	2.42
Discussion	2.43
Eating	2.42
Greeting	2.42
Phoning	2.42
Taking Photo	2.42
Posing	2.42
Sitting	2.42
Sitting Down	2.43
Smoking	2.43
Waiting	2.43
Direction	2.43
Average	2.43

model could generate the 400 milliseconds prediction around 2.4 seconds for a frame in general. On average, the models with linear dimensions of 1024 and 4096 are not significantly different, with only 0.03 seconds differences. The average computation time principally depends on the process and the power of the GPU, which is only comparable when using the same GPU with no other process running since it could affect the time taken by the GPU to generate the prediction.

3.6.3 Qualitative Evalulation

In this section, the author describes the evaluation results in the qualitative-based comparison. **Fig. 3.9** shows the short-term prediction task results on Walking, Eating, Smoking, and Discussion motion. At the same time, the **Fig. 3.10** shows the long-term prediction task results on the same motions. The motions of the qualitative-based evaluation are shown with respect to the 5 frame steps to show the differences in moving motions. The blue line refers to the ground truth based on the annotated data from

the corresponding frames, and the green line refers to the prediction results obtained by our models with a linear dimension of 1024. As shown in the figures, some poses failed to be precisely predicted. **Fig. 3.9a** shows quite off from the correct pose in the 5 sequences poses. However, the location of the human is still correct regardless of the pose. At the same time, the 2 last poses seem to follow the pose. **Fig. 3.9c** shows the pose with a large error in the pose prediction. Our model failed to predict the pose well but could predict the location of the human regardless of the pose. This qualitative evaluation is in line with the quantitative evaluation on **Table. 3.1**.

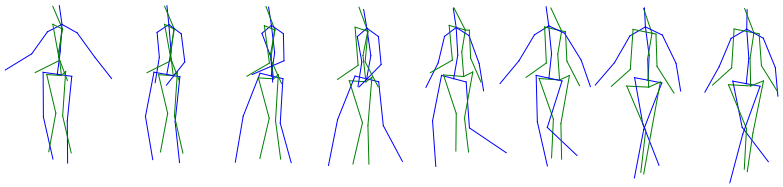
The long-term prediction task performed better in most motions except for the walking motions compared to the short-term prediction task. This could be seen based on the qualitative evaluation results on the **Fig. 3.10** which is in line with the quantitative evaluation result on the **Table. 3.2**.

Fig. 3.11 shows a good prediction result based on the qualitative comparison of the corresponding ground truth. Meanwhile, **Fig. 3.12** shows the bad prediction result. On the good prediction results, our model could predict the pose very well, including the hands and legs movements which are considered the most challenging key points to predict since it moves much more than the other key points.

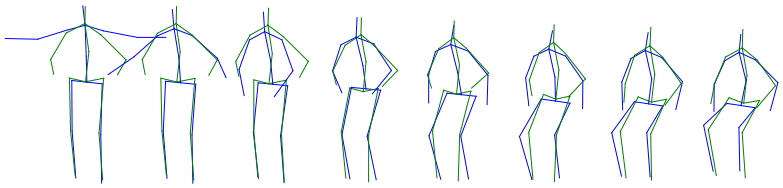
3.7 Summary

This research is conducted to set the baseline of 2D human motion forecasting, which is applicable to most systems that use RGB cameras. In this case, one can also see it as a reliable alternative to the 3D-based data of human motion forecasting. The author proposed the time-series self-attention as a method to predict human motion for the short and long term. This study compared the time-series self-attention method with the LSTM and GRU models.

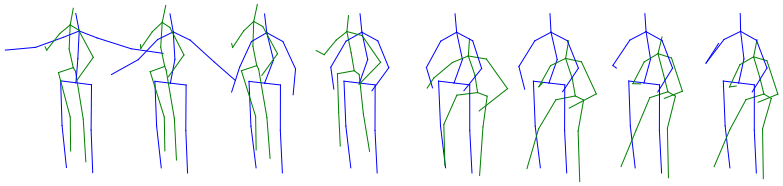
The author evaluates the models based on the MPJPE to measure the error from the prediction to the ground truth, MPJVE to evaluate the movement of every frame in pixels, and MPJLE to calculate the average correct key points in the threshold tolerance value. This study also compared the result when the data obtained by the pose estimation method is used. In this case, the author uses OpenPose as a standard method. In addition, the average computation time of our method is calculated to see



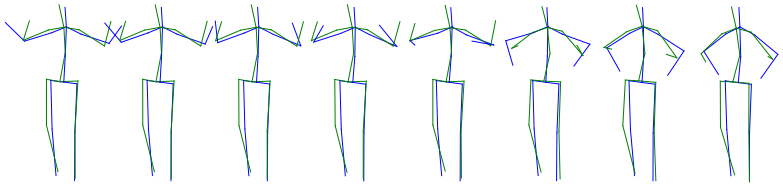
(a) Walking



(b) Eating

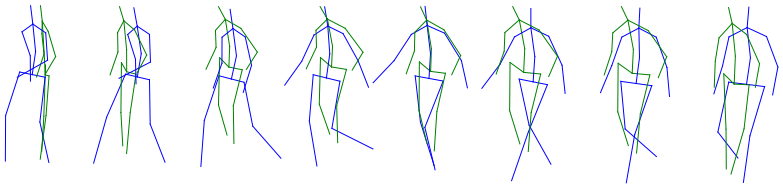


(c) Smoking

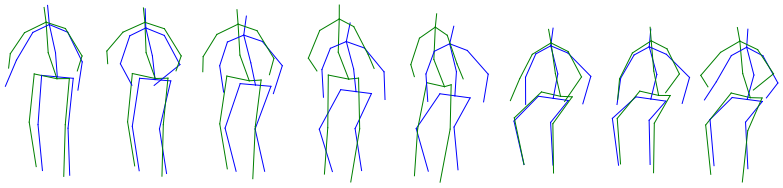


(d) Discussion

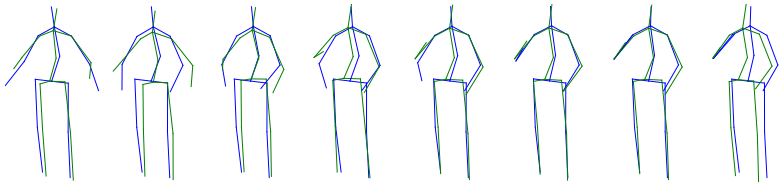
Figure 3.9: Short-term prediction result by our model using 1024 linear dimension in Human3.6M dataset.



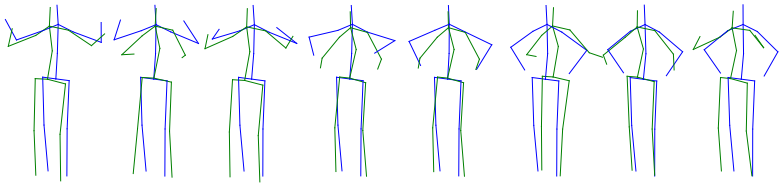
(a) Walking



(b) Eating



(c) Smoking



(d) Discussion

Figure 3.10: Long-term prediction result by our model using 1024 linear dimension in Human3.6M dataset.

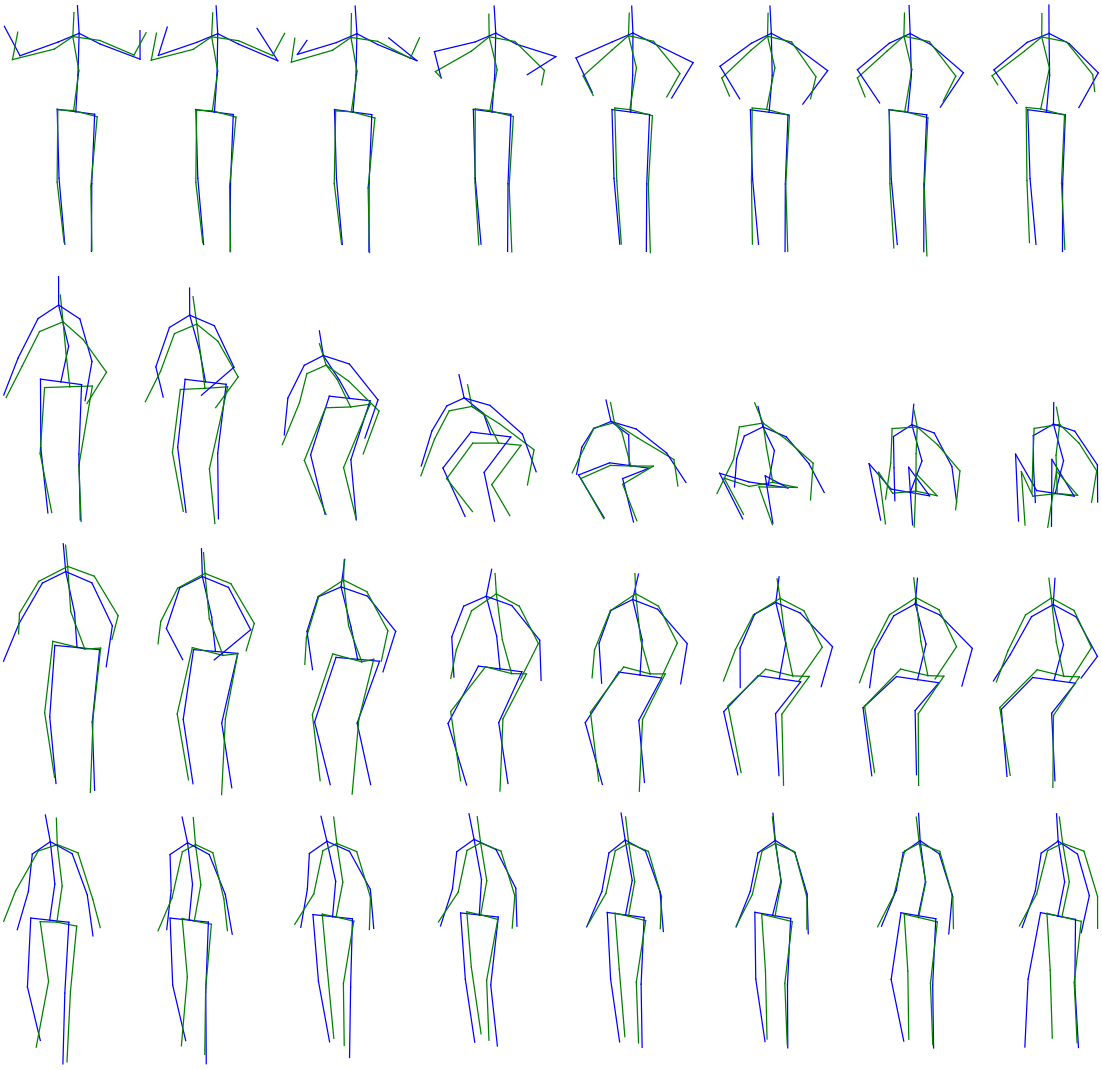


Figure 3.11: Good prediction results obtained by our model with a linear dimension of 1024.

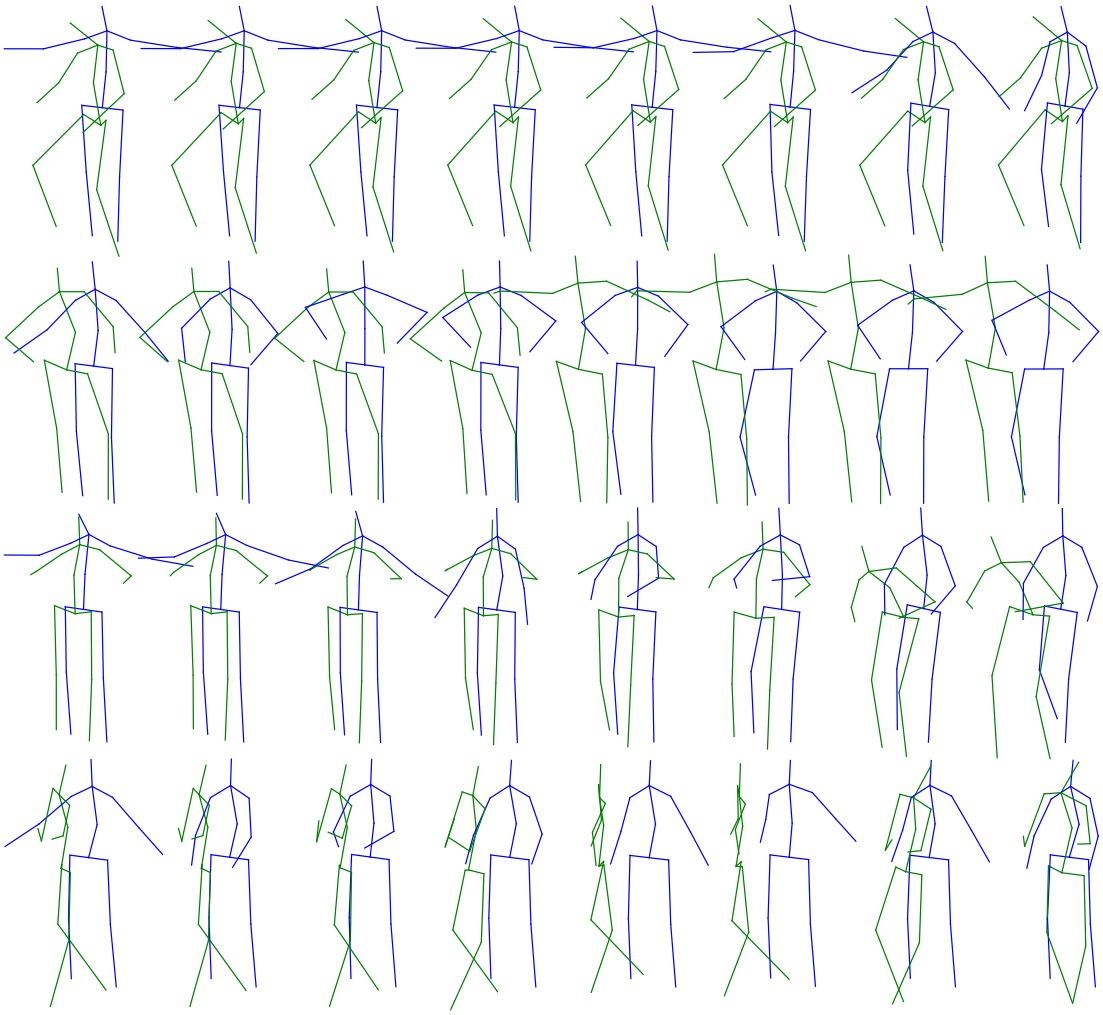


Figure 3.12: Bad prediction results obtained by our model with a linear dimension of 1024.

the real-time usability. As a result, our method could predict short-term and long-term human motion well. Based on the MPJPE, the author found out that the model predicted better on the long-term prediction task than in the short-term. Meanwhile, on the 3DPW dataset, the prediction results for the long-term prediction from our method, LSTM, and GRU models obtained a considerably significant MPJPE metric. The average computation time is below our expectation to be applicable in the real-time system. Although the average computation time relies on the computation device, this issue is still can be solved by reducing the linear dimension on the MLP Head, reducing the number of heads in the self-attention layer, reducing the number of the transformer encoder layer, and also changing the type of data input to Float16. However, this issue needs further research to do to evaluate the result.

While the author strongly believes that the outcome of the method is giving an impact as the baseline for future work in this related field of study. Thus, future relevant work could advance the result based on the evaluation metrics and the average computational time cost.

Chapter 4

Temporal-Spatial Time Series Self-Attention for 2D and 3D Human Motion Forecasting

4.1 Introduction

In this section, the author describes the detail of human motion forecasting using 3D data. In the same way, that human motion forecasting using 2D, using the 3D features has its advantages. The devices to capture the 3D features are broadly developed and produced in the market. For example, the light detection and ranging (LiDAR) camera is developed to capture the depth parameter in a frame. This feature has unlocked the ability to get the third parameter of location over the 2D image frame. Additionally, with the depth parameter, the object detection method could work more precisely to localize the object over the 3D plane recognition. Regarding that, the research on human motion forecasting using 3D data is applicable in the real world, unlocking the application of automation over devices. Adding one more parameter in motion prediction also gives the model another weight to calculate, but also gives extra information to predict better. While the 2D human prediction is evaluated in pixels, the 3D data gives the ability to transform the data into real metrics like millimeters to evaluate and process the visualization in the 3D plane.

Several research addressed the model to forecast based on the temporal dimension, with the Recurrent Neural Network[5], LSTM[5, 51], and Triangular Prism RNN

method [7]. While some other research modeled the forecasting using the spatial and temporal dimensions such as the STS-GCN[8] and MotionMixer[9]. However, understanding the motion based on spatial and temporal dimensions is needed to improve the prediction. Understanding over relation and connection of the spatial and temporal dimensions is necessary. In this research, we improved the method used in the Section. 3 by applying Multi-Layer Perceptron (MLP) for temporal dimension computation. We applied this method for the 2D and 3D human motion forecasting tasks. This research is conducted to provide the feasibility of human motion forecasting in real-world applications using 2D and 3D input.

- We propose a novel self-attention architecture for human motion forecasting tasks.
- We propose a feasibility study on the usability of human motion forecasting applications using unannotated data.
- We provide the standard evaluation metric and compare previous related works in human motion forecasting.
- Our code available at: https://github.com/AndiDemon/HumMovForecasting.

4.2 Related Works

In this section, the author describes the previous related research in which those similarities are found and compared with regard to the dataset, methods, and evaluation metrics.

Based on the baseline, the research on 3D human motion forecasting has been settled on the results over certain evaluation metrics. Started from [5] that aims to recognize and predict the human body pose in videos and motion capture by the encoder recurrent decoder (ERD) model using the Human3.6M dataset. As a result, the prediction of human motion obtained the short-term prediction for 400ms and the long-term prediction for 1000ms. They provided a comparison with state-of-the-art methods such as RNN-LSTM-3LR, CRBM, 6GRAM, and GDPM in walking motion by Mean Angle Error (MAE). Following the research to predict human motion, several works have been done. One research is conducted by using the Structural-RNN model [6]. As a result, the proposed method using S-RNN successfully improved the performance in

the MAE metric on the Eating, Walking, Smoking, and Discussion actions. While more works have been published with better performance based on the MAE [9], the other metric to evaluate the model based on the distance from the prediction result to the ground truth has been introduced [8] and followed by [9]. Furthermore, not only using the Human3.6M dataset, the comparison over another dataset such as AMASS [44] is conducted to validate the various samples and actions. Comparing the models on different datasets and more actions done, this research broadly improved the validation of real-life applications.

Hence, this research is still ongoing, and more state-of-the-art methods, such as the Transformers model [12], have been introduced. The author follows the previous related research to develop the best performance on the MPJPE, MAE, and additionally MPJVE, which is just as important as the other metric to show the smoothness over the changes of the frames. Since the author has researched 2D human motion forecasting, the time-series self-attention is modified to process the 3D data.

4.3 Feature Preparation

In this section, the author describes the data preparation and pre-processing of the 3D features. Similarly, the process is almost the same after the 3D data transformation, given the data with 33 key points all over the human body, including the fingers. At the same time, the features of fingers are not necessary for this research since they will not give more information about human motions. Hence, the key points regarding the fingers are ignored, which remains the 22 key points left. Let the sequence data $\mathcal{X} = \{X_1, X_2, \ldots, X_T\} \in \mathbb{R}^{3\times N\times T}$, where T is defined as the number of frames for N key points. Setting up the sliding window of $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{T_Q}\}$ input, and $\mathcal{P} = \{P_{T_Q+1}, P_{T_Q+2}, \ldots, P_{T_Q+T_P}\}$ expected output. Aiming to predict the N key points for the next $T_Q + T_P$ future frames $\hat{P} = \{\hat{P}_{T_Q+1}, \hat{P}_{T_Q+2}, \ldots, \hat{P}_{T_Q+T_P}\}$ respectively to the frames. As for the subject on the Human3.6M dataset, the training data includes subject numbers 1, 6, 7, 8, and 9. Subject number 11 is used as the validation data, and subject number 5 is used as the testing data. This setting is used in several related research to keep the comparison between the methods in line [8, 9].

4.4 Proposed Method

4.4.1 Temporal-Spatial Time Series Self-Attention

In this section, the author describes the detail of the prediction method model. As described in the 4.1, the architecture of the model is quite different when using 2D features. Given the 3D features in the shape of $(batch\_size, input_window_frames, \mathbb{Q})$ as the input, it is processed with positional encoding and dropout layers. Then, the CNN block layer with SE Block and the Add layer. After that, this step is repeated by L times. Following the CNN block step, Transformer Encoder with SE Block and Add layer is applied repeatedly by M times. Finally, the tensors are finalized by the MLP Head layer to obtain the prediction.

4.4.2 Loss Metric

In this section, the author describes the detail of the method to evaluate the method in the training and testing phase. These evaluation metrics are used as the loss function and to evaluate the method regarding the prediction results.

Following the metric evaluation protocol from previous related research. MPJPE is employed to evaluate the Euclidean distance between the forecasting result to the ground truth using the cartesian coordinate[8, 9]. At the same time, MAE is used to evaluate based on the Eular-angle representation[8, 9]. MPJPE is defined as:

$$E_{\text{MPJPE}} = \frac{1}{N}\|\mathbf{P}_{T_p} - \hat{\mathbf{P}}_{T_p}\| \tag{4.1}$$

where N is the total number of frames in the testing data, $\hat{P}_{T_P}$ determines the prediction on the number T_p frame of the output window, and the P_{T_p} is the corresponding ground truth.

Processing 3D data gives another parameter in the way to understand the direction of the human body. As for this matter, the evaluation metric is needed to evaluate the correctness of the prediction by the model. Mean Angle Error is determined to evaluate the prediction by computing the Euler rotation data processed by the model as the input. MAE is defined by:

$$E_{\text{MAE}} = \frac{1}{N}\|\mathbf{P}_{T_p} - \hat{\mathbf{P}}_{T_p}\| \tag{4.2}$$

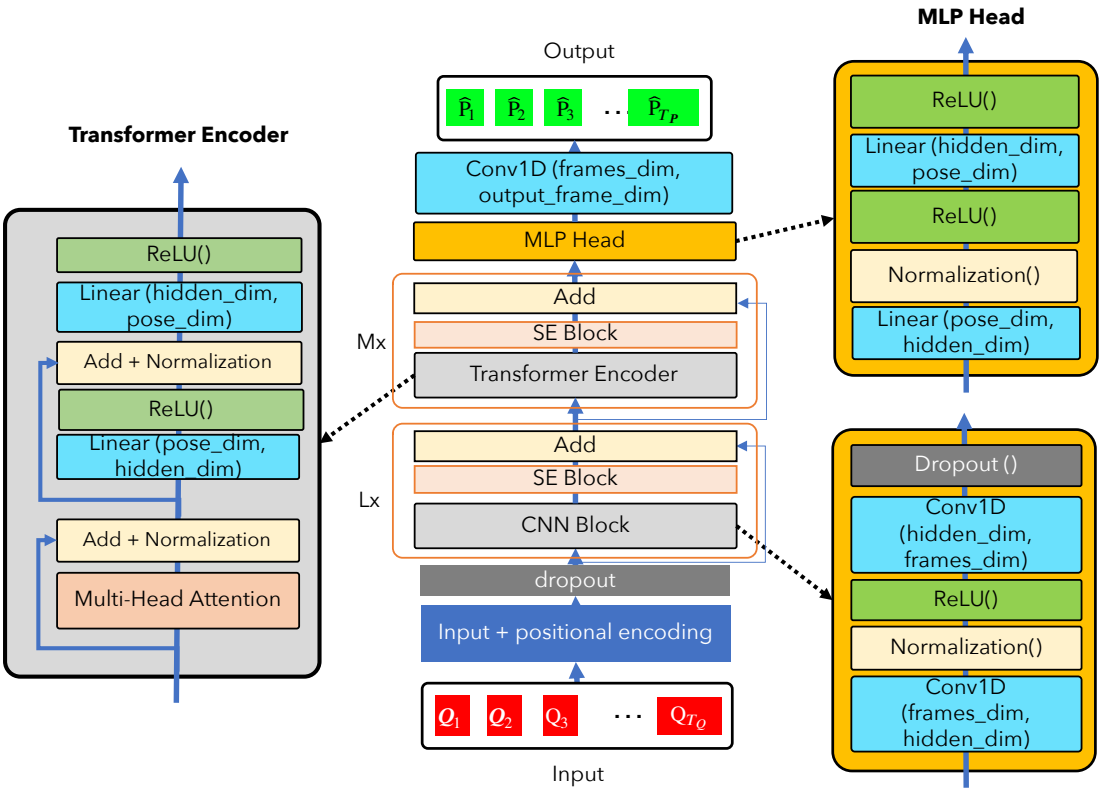


Figure 4.1: Temporal-Spatial Time Series Self-Attention architecture for 2D and 3D human motion forecasting. We defined the pose_dim as the input pose dimension that differs based on the dataset, frames_dim as the number of input frames, hidden_dim as the hidden dimension on the neural network, and output_frame_dim as the expected frame output dimension. The input data are processed by the positional encoding and dropout layer. The first unit consists of the CNN block, the squeeze-and-excitation (SE) block, and the skip connection as the temporal dimension computation. The first unit is repeated L times. Then followed by the second unit which consists of the transformer encoder block, the SE block, and the skip connection for context-relation awareness. The second block is repeated M times. Finally, the multilayer perceptron (MLP) Head computes the spatial dimension prediction, and we use the 1D convolutional layer to transform the frame dimension for the output.

where N is the total number of frames in the testing data, $\hat{P}_{T_P}$ determines the prediction on the number T_p frame of the output window, and the P_{T_p} is the corresponding ground truth.

4.5 Experiments

In this section, the author describes the overall experimental details, which include dataset setup, experimental setup, model configuration, and the device used to perform the experiment.

4.5.1 Dataset

In this section, the author describes the dataset that is used in this research as well as the dataset setup. Similar to the research on 2D human motion forecasting in Section. 2, the Human3.6M dataset is used as the main dataset to be evaluated with other previous related research however, if the 2D human motion forecasting research using the 2D input data that has been interpolated from the 3D data. In this research, the input data is the 3D cartesian coordinates and the Euler angle rotation map obtained by motion capture of the camera sensors. The data is divided into training, validation, and testing. The training data consists of subject numbers 1, 6, 7, 8, and 9. While subject number 11 is determined as the validation data, and subject number 5 is the testing data. 22 joints for forecasting the 3D body pose as the input and 16 for the angle-based prediction.

At the same time, the AMASS dataset is used to validate the method on another dataset. AMASS dataset is separated by training, validation, and testing data based on the sub-dataset provided in AMASS. Training data contains the samples, including CMU, MPI Limits, TotalCapture, Eyes Japan Dataset, KIT, EKUT, TCD handMocap, and ACCAD. Validation data contains the samples, including HumanEva, MPI HDM05, SFU, and MPI mosh. The testing data contains a sample of BioMotionLab NTroje. In this dataset, the 3D cartesian coordinates data is used with 18 key points as the input.

4.5.2 Experimental Setup

In this section, the author describes the experimental setup used to configure the features, train the model, and visualize the prediction. Part of this research is done by the previous research to keep the comparison in line with the previous related research [9]. The input window in this research is ten frames consecutively with the option of 1 or 5 steps over the sliding window. Expecting the 10 or 25 frames as the output in the sliding window to get the output of 400ms or 1000ms. As shown in the model overview, the CNN block layer is repeated three times, and the Transformer Encoder layer is repeated two times. The dropout layer is defined by 0.1, with the learning rate to train the model being 0.0001 using the ADAM optimizer. As for the loss function, the MPJPE is used to calculate when the coordinate data is used, and MAE is used to calculate the loss when the angle rotation is used. The author used the NVidia GeForce RTX 4090 to perform the experiment.

4.6 Results

4.6.1 Quantitative Evaluation

In this section, the author describes the evaluation results based on the MPJPE and MAE metrics for short-term and long-term 3D human motion prediction tasks.

4.6.1.1 2D Human Motion Forecasting

Human3.6M Dataset

Comparison based on MPJPE. Table 4.1 shows the MPJPE for short and long-term forecasting tasks and Table 4.2 shows the MPJVE to measure the proximity between the ground truth and the prediction in terms of joint positions and velocities, respectively. In comparison, LSTM[51] and GRU[52] are employed to predict human motion. The results show that our method outperformed the other two methods in terms of both MPJPE and MPJVE at both time intervals. Specifically, our method achieved an MPJPE of 12.32 and 15.17 pixels and an MPJVE of 0.87 and 1.19 pixels at 400 ms and 1000 ms, respectively, while the other methods achieved lower performance on both metrics. Table 4.3 shows the MPJE and MPJVE metrics when the data obtained by OpenPose is used. Compared to the ground truth test, MPJPE is

Table 4.1: MPJPE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data.

Motion	400 msec			1000 msec		
	TS-TSSA (Ours)	LSTM	GRU	**TS-TSSA (Ours)**	LSTM	GRU
Walking	**14.06**	14.38	13.90	**19.61**	12.11	12.74
Eating	**7.17**	13.14	14.48	**9.01**	14.09	13.02
Smoking	**7.10**	16.47	17.40	**9.81**	15.21	16.53
Discussion	**12.37**	20.62	20.75	**16.94**	21.21	19.48
Direction	**9.33**	21.55	21.77	**12.25**	20.35	20.56
Greeting	**8.24**	36.67	35.42	**14.72**	33.92	32.56
Phoning	**10.80**	19.00	18.89	**14.46**	18.41	16.93
Waiting	**9.92**	26.98	25.84	**12.37**	24.46	22.37
Walking Dog	**10.69**	41.19	39.60	**13.48**	39.99	38.82
Walking Together	**12.00**	45.42	46.32	**12.26**	40.82	29.62
Posing	**8.19**	30.76	27.86	**16.17**	26.38	25.96
Sitting	**18.37**	18.91	21.74	**19.45**	20.73	19.48
Sitting Down	31.80	**27.95**	28.84	32.98	31.25	**25.74**
Taking Photo	**6.52**	36.58	34.52	**8.93**	40.70	29.50
Average	**12.32**	26.40	26.24	**15.17**	25.69	23.09

significantly changed from 12.32 to 30.78 pixels on our method at 400 ms prediction task and 15.17 to 34.96 pixels at 1000 ms prediction task. While on LSTM and GRU, the MPJPE is changed, but not more than 2 times the ground truth testing. Furthermore, the evaluation of MPJVE is also increased significantly. These changes happened due to the uncertainty of the pose estimation obtained by OpenPose. However, to evaluate whether the forecasting result is reliable or not, the qualitative evaluation is explained in section 4.6.2.

3DPW dataset is used to evaluate the model on outdoor activity with undefined scenarios. As shown in table 4.4, the forecasting result is evaluated based on the MPJPE metric. Our method obtained the best forecasting result compared to the LSTM and GRU. However, the MPJPE metric obtained is too big compared to the dataset frame size. This forecasting failure appeared due to the undefined scenarios of the 3DPW dataset. This summarizes our method is not reliable in the unknown scenario activity.

Table 4.2: MPJVE of 2D joint positions in pixel on the Human3.6M dataset using the real position data as the testing data.

Motion	400 msec			1000 msec		
	TS-TSSA (Ours)	LSTM	GRU	**TS-TSSA (Ours)**	LSTM	GRU
Walking	**1.06**	1.15	1.23	1.16	**1.07**	1.22
Eating	**0.64**	0.97	1.00	**0.79**	1.04	1.10
Smoking	**0.56**	0.86	0.89	**0.79**	0.84	0.88
Discussion	**1.03**	1.27	1.33	1.59	**1.31**	1.24
Direction	**0.67**	1.15	1.23	**1.09**	1.15	1.16
Greeting	**1.21**	2.77	2.91	**1.54**	2.39	2.67
Phoning	**0.86**	1.11	1.10	1.15	**1.11**	1.14
Waiting	**0.89**	1.60	1.62	**1.28**	1.45	1.62
Walking Dog	**1.15**	2.51	2.38	**1.68**	2.16	2.37
Walking Together	**0.69**	2.36	2.35	**0.98**	2.00	2.03
Posing	**1.04**	1.58	1.60	**1.46**	1.58	1.67
Sitting	**0.67**	1.18	1.44	**0.88**	1.41	1.29
Sitting Down	**1.03**	1.30	1.21	**1.28**	1.61	1.53
Taking Photo	**0.67**	1.31	1.24	**1.01**	1.40	1.33
Average	**0.87**	1.51	1.54	**1.19**	1.47	1.52

Table 4.3: Evaluation based on MPJPE and MPJVE metrics of 2D joint position on the Human3.6M dataset with the position data obtained by OpenPose as testing data. (Pixels)

Error	400 ms			1000 ms		
	Ours	LSTM	GRU	**Ours**	LSTM	GRU
MPJPE	**30.78**	33.61	34.64	34.96	**33.83**	33.85
MPJVE	**5.88**	12.01	15.70	**7.11**	11.82	15.56

Table 4.4: Evaluation based on MPJPE of 2D joint position on the 3DPW dataset. (Pixels)

Error	1000 ms		
	Ours	LSTM	GRU
MPJPE	**156.85**	240.94	236.31

4.6.1.2 3D Human Motion Forecasting

Human3.6M Dataset

Comparison based on MPJPE. Table. 4.5 shows the comparison based on the MPJPE metric from the current related research methods. As mentioned in Section 4.4.2, the MPJPE is calculated by computing the distances between the prediction result and its corresponding ground truth with respect to the key points and frames.

Thus, in the **Table. 4.5** our method is compared with another state-of-the-art for the short and long-term prediction task. The research on 3D human motion prediction has been conducted since 2015 using the Res. Sup resulting in the MPJPE score 88.3mm for the short-term prediction and 136.6mm for the long-term prediction task on average. Followed by the convSeq2Seq in 2019, with a slightly better performance on average, obtaining 72.7mm for short-term prediction tasks and 124.2mm for the long-term prediction task. LTD-10-25, RNN-GCN, and MultiAttention obtained small improvements in the prediction results over time. Then, STS-GCN comes out with the breakthrough of the MPJPE score below 100mm. STS-GCN obtained the average prediction over all motions with 75.6mm and obtained the best current prediction result on the Greeting motion with an MPJPE score of 91.6mm for the long-term prediction task. While after that, MotionMixer with MLP Based method obtained the current best prediction almost in overall motions and the average for the short

and long-term prediction tasks. MotionMixer obtained the best result on average with 33.6mm for the short-term prediction task and 71.6mm for the long-term prediction task by MPJPE. On the other hand, our method, based on the current evaluation result of MPJPE, obtained second place for the short and long-term prediction tasks. Our method obtained an MPJPE score average of 36.4mm for short-term prediction tasks and 73.2mm for long-term prediction tasks. Our method achieved the best result on the Walking Dog for the short-term prediction task with 54.1mm and the Walking Together motion with 49.9mm for the long-term prediction task.

Even though. our method is not achieving the best result over other previous related research, the prediction results are good enough to tell the future human motion prediction for 400$msec$ and 1000$msec$ ahead. The visualization of this prediction result can be seen in Section 4.6.2.

Comparison based on MAE. Table. 4.6 shows the comparison based on the MAE metric from the current state-of-the-art methods. Our method performed quite well with the average prediction result obtained the MAE with 1.56 for the short-term prediction task, and 1.77 for the long-term prediction task. However, comparing to the other previous related methods, our method could not perform better when using the angular data. Difference between the best result obtained by the STS-GCN [8] is quite significant with around 0.7 MAE metric different. With this in mind, our method could predict when the coordinate data is used while in terms of the angular data, our method could not improve the prediction result from the previous related works.

AMASS Dataset

In this section, the author describes the evaluation of the experiment using the AMASS dataset based on the MPJPE metrics. Since AMASS dataset does not provide the angular data, the evaluation based on MPJPE is the only evaluation metric that could fit on this dataset. **Table. 4.7** shows the evaluation result based on MPJPE using the AMASS dataset for the short and long-term prediction tasks.

4.6.2 Qualitative Evaluation

In this section, the author describes the evaluation based on the qualitative results.

Table 4.5: MPJPE evaluation using Human3.6M dataset for short-term and long-term 3D human motion forecasting task.

	Walking		Eating		Smoking		Discussion		Directions	
Forecasting time (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
Res. sup [17]	66.1	79.1	61.7	98.0	65.4	102.1	91.3	131.8	84.1	129.1
convSeq2Seq [53]	63.6	82.3	48.4	87.1	48.9	81.7	77.6	129.3	69.7	115.8
LTD-10-25 [36]	44.4	60.9	38.6	75.8	39.5	72.1	68.1	118.5	58.0	105.5
RNN-GCN [54]	39.8	58.1	36.2	75.7	36.4	69.5	65.4	119.8	56.5	106.5
MultiAttention [55]	39.0	57.1	45.2	73.7	29.0	68.7	64.0	117.5	62.6	105.7
STS-GCN [8]	32.9	51.8	25.4	52.4	25.8	50.0	40.2	78.8	34.7	71.0
MotionMixer [9]	**28.6**	**49.2**	**20.9**	**47.4**	**21.4**	**45.4**	**35.5**	**78.0**	**29.2**	**66.5**
Ours	32.5	51.7	23.5	48.2	23.7	47.6	39.1	79.4	33.3	70.1

	Greeting		Phoning		Posing		Purchases		Sitting	
Forecasting time (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
Res. sup [17]	108.8	153.9	76.4	126.4	114.3	183.2	100.7	154.0	91.2	152.6
convSeq2Seq [53]	96.0	147.3	59.9	114.0	92.9	187.4	89.9	151.5	63.1	120.7
LTD-10-25 [36]	82.6	136.8	50.8	105.1	79.9	174.8	78.1	134.9	58.3	118.7
RNN-GCN [54]	78.1	138.8	49.2	105.0	75.8	178.2	73.9	135.9	56.0	138.8
MultiAttention [55]	85.4	136.7	44.1	104.6	78.7	172.9	67.9	133.1	66.3	115.0
STS-GCN [8]	49.2	**91.6**	30.9	66.1	45.6	106.4	48.7	93.5	35.0	75.2
MotionMixer [9]	**46.2**	93.6	**27.8**	**63.4**	**40.1**	**99.7**	**42.7**	**88.7**	**29.8**	**68.9**
Ours	49.8	96.1	29.5	63.9	44.4	103.1	46.2	90.5	31.7	70.3

	Sitting Down		Taking Photo		Waiting		Walking Dog		Walking Together	
Forecasting time (msec)	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*	*400*	*1000*
Res. sup [17]	112.0	187.4	87.6	153.9	87.7	135.4	110.6	164.5	67.3	98.2
convSeq2Seq [53]	82.7	150.3	63.6	128.1	69.7	117.7	103.3	162.4	61.2	87.4
LTD-10-25 [36]	76.4	143.8	54.3	115.9	44.4	108.3	38.6	146.4	39.5	65.7
RNN-GCN [54]	72.0	143.6	51.5	115.9	54.9	108.2	86.3	146.9	41.9	64.9
MultiAttention [55]	66.3	141.8	49.4	115.2	71.1	105.1	119.0	141.4	42.0	63.2
STS-GCN [8]	47.9	94.3	33.6	76.9	35.2	72.0	59.6	102.6	30.5	51.1
MotionMixer [9]	**42.6**	**89.3**	**27.9**	**66.6**	**30.0**	**68.2**	**54.1**	**99.6**	**27.4**	50.4
Ours	45.2	90.5	30.1	67.6	33.7	69.5	**54.1**	99.8	29.8	**49.9**

	Average	
Forecasting time (msec)	*400*	*1000*
Res. sup [17]	88.3	136.6
convSeq2Seq [53]	72.7	124.2
LTD-10-25 [36]	68.1	112.4
RNN-GCN [54]	58.3	112.1
MultiAttention [55]	60.0	110.1
STS-GCN [8]	38.3	75.6
MotionMixer [9]	**33.6**	**71.6**
Ours [9]	36.4	73.2

Table 4.6: MAE evaluation using Human3.6M dataset for short-term and long-term 3D human motion forecasting task.

	Average MAE	
Forecasting time (msec)	*400*	*1000*
Res. sup [17]	1.15	-
convSeq2Seq [53]	1.13	1.82
LTD-10-25 [36]	1.04	1.68
RNN-GCN [54]	1.04	1.65
MultiAttention [55]	0.93	1.57
STS-GCN [8]	0.66	**1.07**
MotionMixer [9]	**0.63**	1.08
TS-TSSA (Ours)	1.56	1.77

Table 4.7: Evaluation based on MPJPE using AMASS dataset for short-term and long-term 3D human motion forecasting task.

	Average MPJPE	
Forecasting time (msec)	*400*	*1000*
convSeq2Seq [53]	67.6	93.5
LTD-10-25 [36]	45.3	75.2
RNN-GCN [54]	42.0	67.2
MultiAttention [55]	41.2	65.8
STS-GCN [8]	24.5	45.5
MotionMixer [9]	**21.9**	**41.6**
TS-TSSA (Ours)	46.0	63.7

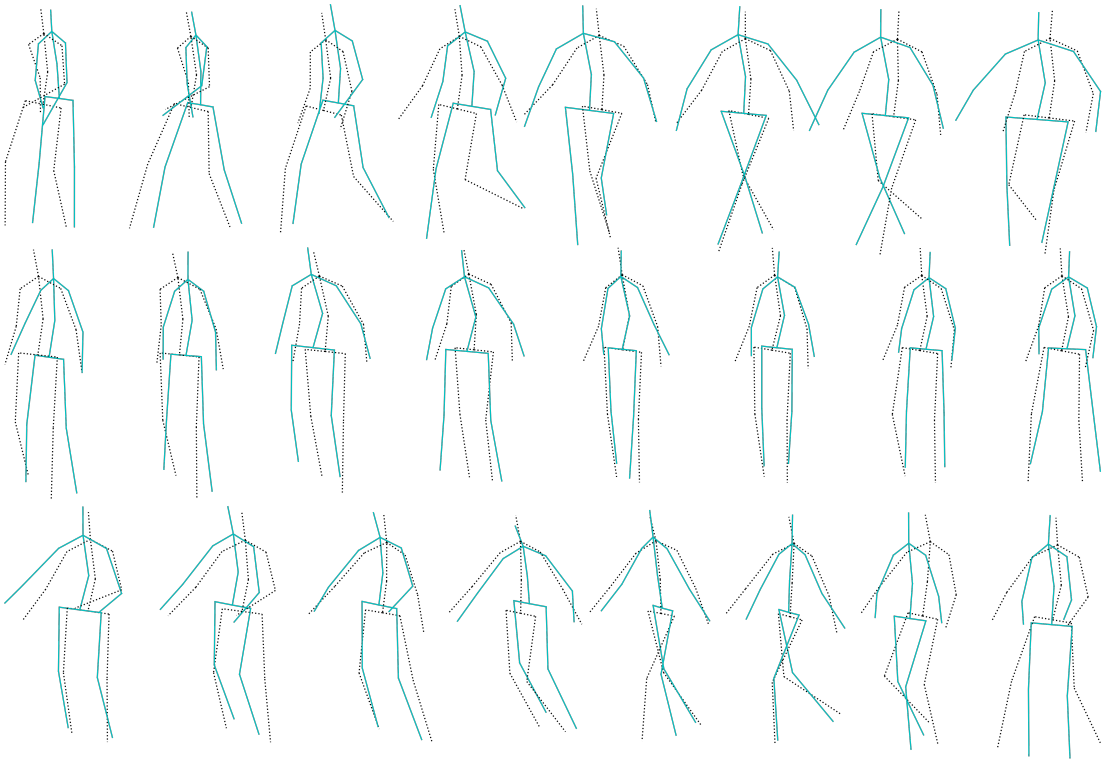


Figure 4.2: 2D qualitative evaluation on Walking, Walking Together, and Walking with Dog motions respectively from top to bottom.

4.6.2.1 2D Human Motion Forecasting

Figure 4.2 shows the qualitative evaluation of Walking, Walking Together, and Walking with Dog motions. Ground truth is defined by the grayscale dashed lines, while the straight blue lines define prediction. Our method could perform the 2D human motion forecasting task well with a very slight error over the pose. In addition, qualitative evaluation based on the OpenPose testing is shown in Figure 4.4. Our method could perform well in forecasting the human body position, but it could not provide a good forecasting result on the human body pose. The OpenPose pose estimation looks rigid, which is becoming the reason for the rigid forecasting result from our method.

4.6.2.2 3D Human Motion Forecasting

Figures. 4.5, **4.6**, and **4.7** show the long-term human motion prediction result comparing with its correspondent ground truth. The colored lines green and purple are

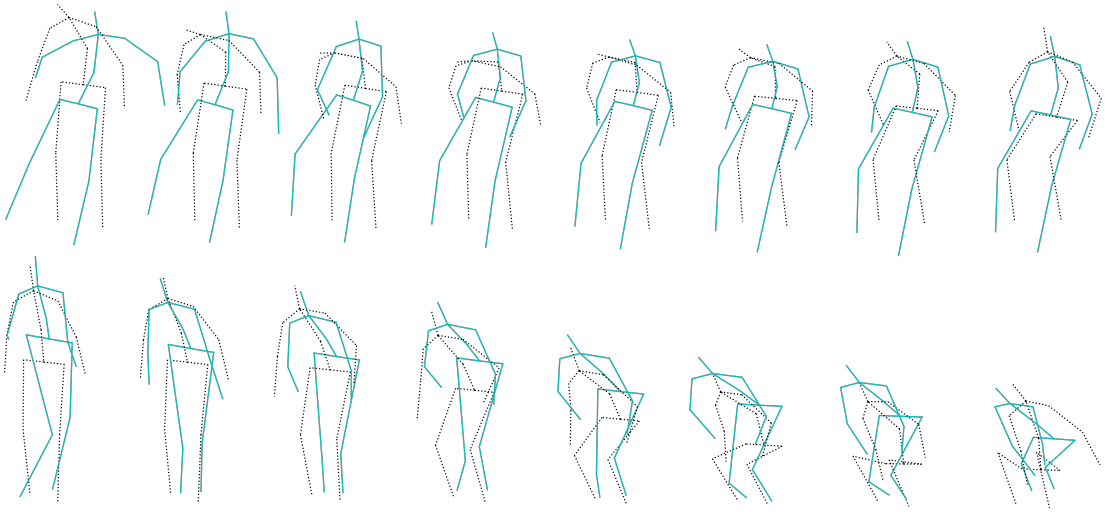


Figure 4.3: 2D qualitative evaluation on Sitting, and Sitting Down motions respectively from top to bottom.

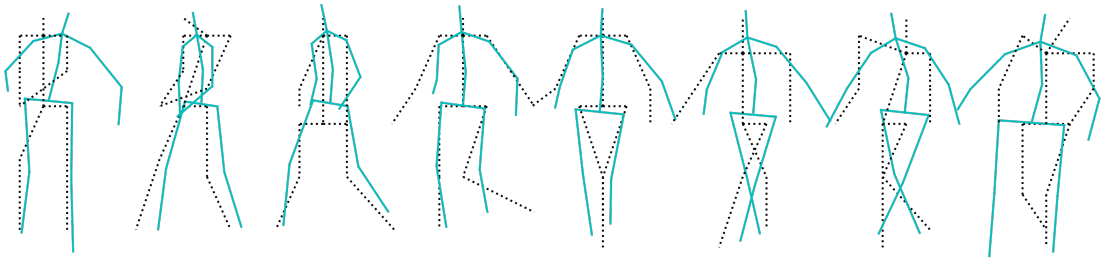
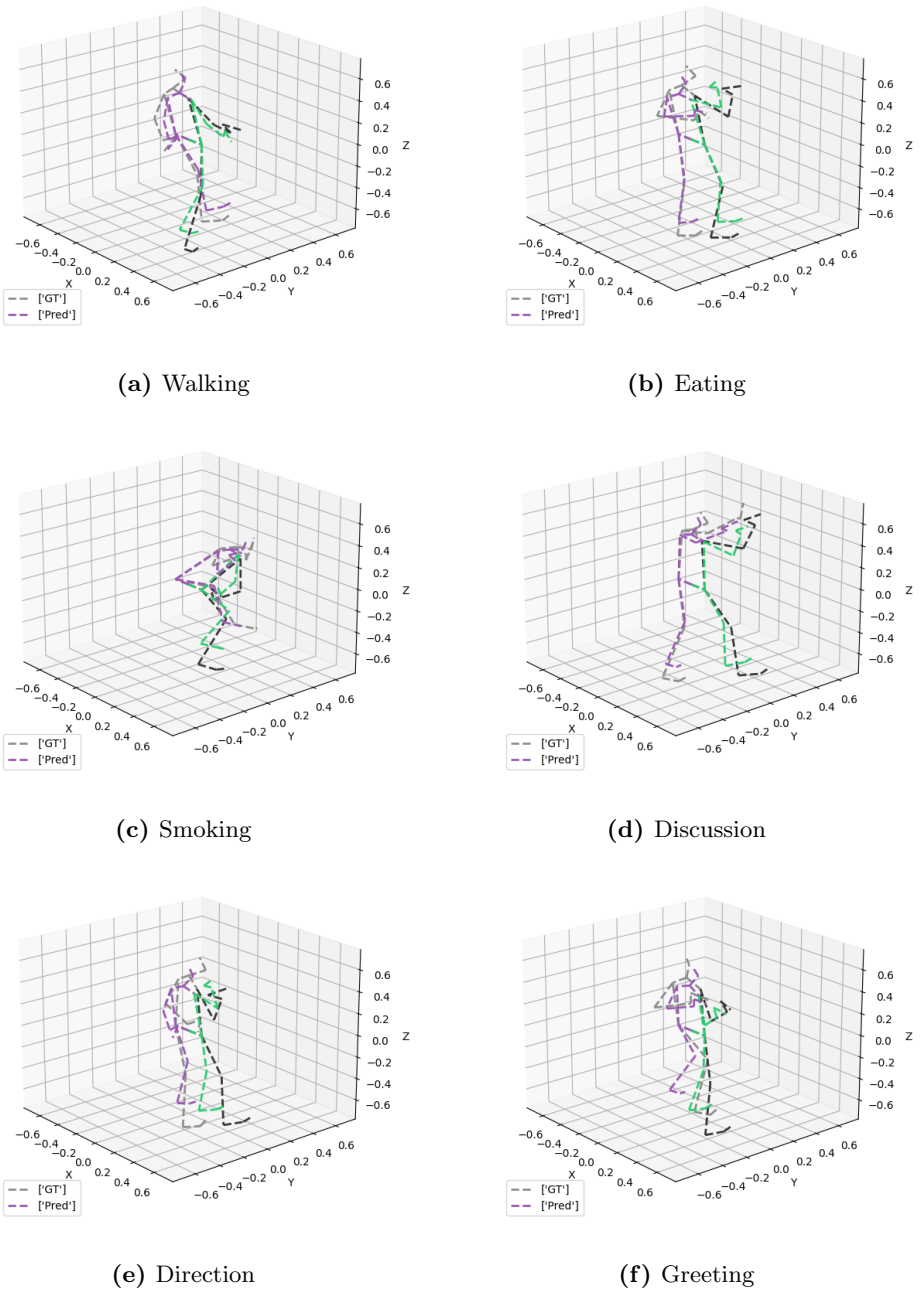


Figure 4.4: 2D qualitative evaluation on Walking motion using the data obtained by OpenPose.

defined as the prediction results obtained by our model, while the black and grey are defined as the corresponding ground truth. The visualization is made by generating a random frame in motion. Generally, our model could predict human motion quite well qualitatively. **Figures 4.6f** and **4.6d** show the visualization of the taking photo and sitting motion. Our model failed to predict the left-hand and right-hand gestures. Despite that, the result of the phoning and greeting motions failed to be predicted in the right-leg and left-leg pose. These results aligned with the quantitative evaluation based on MPJPE when the phoning and greeting motions obtained big MPJPE values, which indicates the motion to be quite difficult to predict due to the dynamic movement over time, even though the other motions show the tendency to be predicted well.

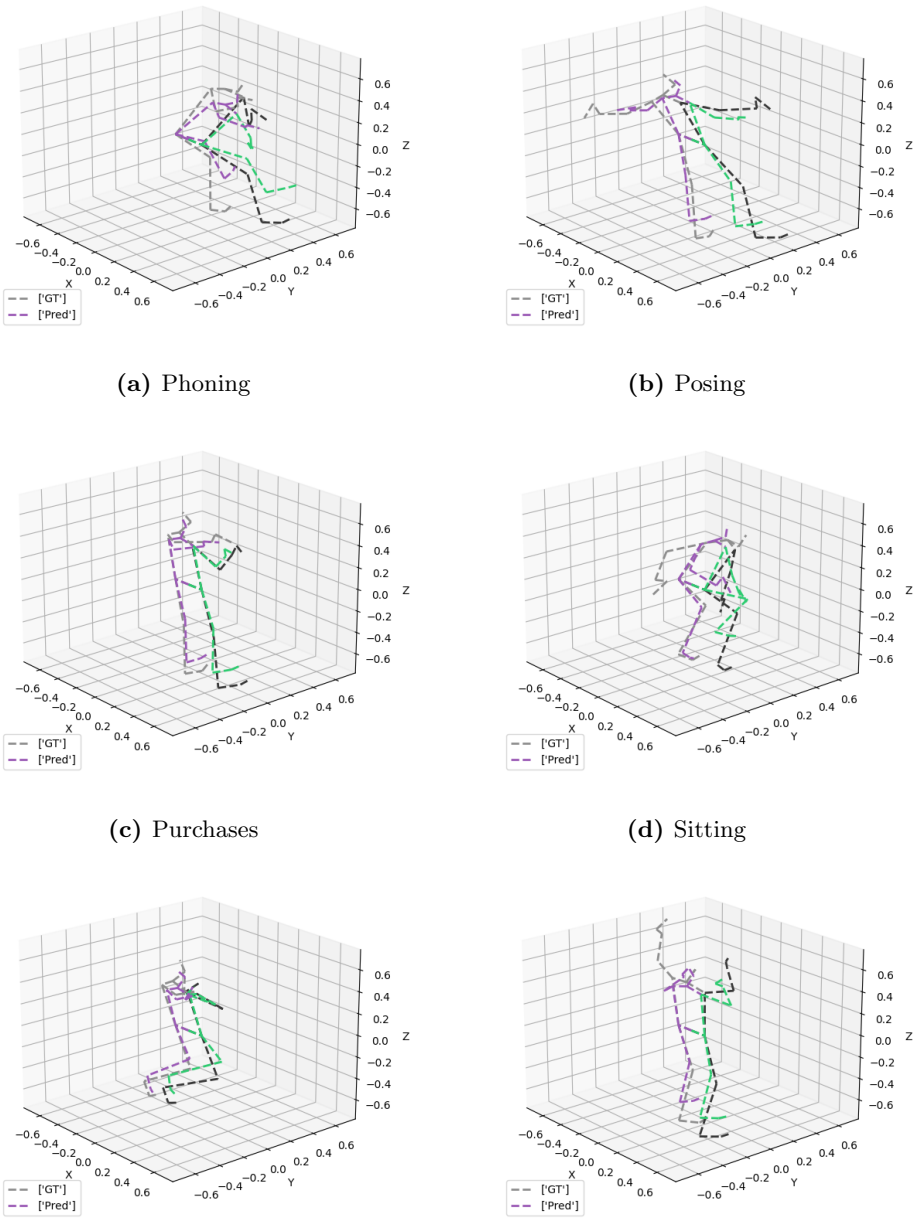


(a) Walking **(b)** Eating

(c) Smoking **(d)** Discussion

(e) Direction **(f)** Greeting

Figure 4.5: Long-term prediction result by our model in Human3.6M dataset.

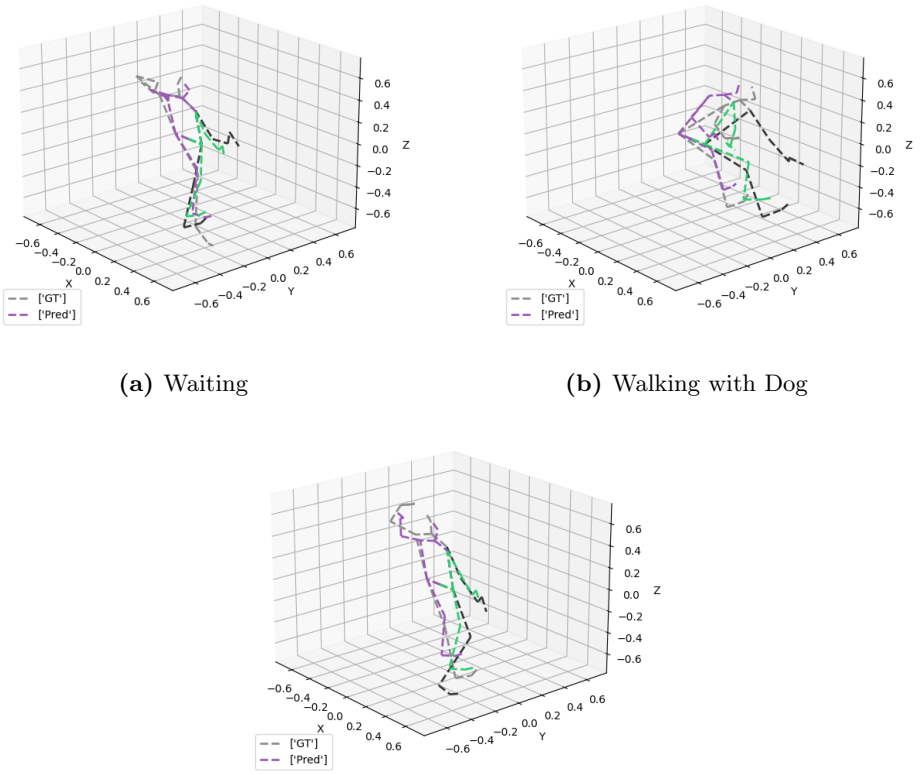


(a) Phoning **(b)** Posing

(c) Purchases **(d)** Sitting

(e) Sitting Down **(f)** Taking Photo

Figure 4.6: Long-term prediction result by our model in Human3.6M dataset.



(a) Waiting

(b) Walking with Dog

(c) Walking Together

Figure 4.7: Long-term prediction result by our model in Human3.6M dataset.

Table 4.8: Computational complexity analysis on long-term 2D human motion forecasting.

Method	Parameters	≈ FLOPs	MPJPE
LSTM	300k	109M	25.69
GRU	230k	82M	23.09
TS-TSSA (Ours)	313k	0.89M	15.17

Table 4.9: Computational complexity analysis on long-term 3D human motion forecasting.

Method	Parameters	≈ FLOPs	MPJPE
MultiAttention [55]	3.4M	-	110.1
MSR-GCN [56]	6.3M	192.4M	114.2
STS-GCN [8]	57.5k	7.1M	75.6
MotionMixer [9]	30.2k	2.1M	71.6
TS-TSSA (Ours)	308.1k	105.6M	73.2

4.6.3 Complexity Evaluation

Evaluations based on the number of parameters and floating operations FLOPs on the model are compared for the computational cost and performance. Table.4.8 shows the comparison of the parameters and approximate FLOPs on the long-term 2D human motion forecasting task. In comparison with the RNN-based method, our method TS-TSSA obtained the best performance with nearly the same number of parameters and estimated FLOPs 121 times smaller. Meanwhile on the Table.4.9 shows the 3D human motion forecasting task. TS-TSSA (ours) obtained around the same performance with 5 to 10 times the number of parameters compared to the STS-GCN[8] and MotionMixer[9].

4.7 Summary

The objective of this research is to demonstrate the viability of 2D and 3D human motion forecasting in real-world applications. Human motion forecasting research has been conducted with quite a various methods, separating 2D and 3D input data. In

this paper, Temporal-Spatial Time Series Self-Attention (TS-TSSA) is proposed to forecast human motion for short and long-term prediction tasks. As a result, TS-TSSA outperformed the RNN-based method based on the MPJPE and MPJVE evaluation metrics for the 2D forecasting task using the Human3.6M dataset and 3DPW dataset. Using the data obtained from the pose estimation method as the input data, our method did not manage to yield satisfactory predictions of the human pose. Nevertheless, it still could provide good forecasting results regarding the human body's position. This indicates that our method is applicable to real-world applications to provide human movement forecasting. While for the 3D forecasting task, our method performed well based on the MPJPE evaluation metric. The results show comparable achievement to the previous related works. In conclusion, our method could be suitable for both 2D and 3D human motion forecasting tasks. The author believes that these outcomes could give more improvement on the way of using the self-attention-based approach.

Chapter 5

Conclusion

Autonomous systems have been developed for various applications. However, for safety, these systems must consider the movement of objects around them. This is where human motion prediction comes into play, as it helps to prevent accidents, both for others and for the autonomous devices themselves. For example, self-driving cars can use human motion prediction to anticipate and respond to human behavior, robots can use it to interact more effectively with humans, and devices designed to support the elderly can use it to prevent falls. Many more potential applications could benefit from human motion prediction. With this in mind, the author proposes steps of research to realize human motion prediction in real-world applications.

In the second chapter, the author proposed a method to predict human motion using the unannotated data obtained from the commonly used method to generate the human body pose. The 2D pose estimation: OpenPose is used to generate the human body pose in real-time, then the RNN-LSTM and Kalman Filter are used to generate the future human motion for one second ahead. As a result, this research confirmed the usability of 2D human motion prediction in real-world applications.

In the third chapter, the author proposed the improvement of the 2D human motion prediction by using the annotated data and the novel proposed method. The Human3.6M and 3DPW datasets are used as the main dataset to be compared with the other state-of-the-art methods. The author proposed the Time Series Self-Attention method as the model to predict human motion for the short and long term. As a result, our proposed method outperformed the RNN-LSTM and RNN-GRU in the short and long-term prediction task using the Human3.6M dataset. However, when using the

3DPW dataset, our method, as well as the RNN-based method, could not perform well due to the varied uncategorized data in the 3DPW dataset. In addition to the usability confirmation, the author added the evaluation using the data obtained by the pose estimation method. As a result, our method could perform very well in predicting the human location but could not predict well regarding the human pose. In conclusion, this research could provide improvement of the 2D human motion prediction and could be used as the baseline to be compared with other works in the future.

The technologies to obtain a more precise location of the human pose are growing. The more specific data that could be obtained means the more complex process of generating the human motion prediction. Due to this reason, in the fourth chapter, the author applied the Time Series Self-Attention method in the 3D human motion forecasting task. Since this research has been developed by many other previous works, our proposed method could be compared with other related research with respect to the dataset, the configuration of the data, and the evaluation metric. As a result, our method could predict well the human pose using the Human3.6M and reach the 2nd world best position based on the MPJPE evaluation metric for the short and long-term prediction task. However, our method could not predict well when using the angular data in which the MAE evaluation metric takes place. By using AMASS dataset, our method could perform well in predicting human motion, but the results are not quite competitive compared to the other previous methods.

In conclusion, the author performed the study on 2D and 3D human motion prediction. the author confirmed the usability and performance of the proposed method in both 2D and 3D human motion prediction for the short and long term. The self-attention-based method is applicable for time-series tasks such as human motion prediction, which could lead to future works for more applications of deep learning, such as the classification task, object recognition, and many more applications.

References

[1] www.robotiksistem.com, "What is a sensor?." `http://www.robotiksistem.com/robot_sensors.html`, 2009-2019. [Online; accessed 11-September-2022]. 1

[2] A. Carullo and M. Parvis, "An ultrasonic sensor for distance measurement in automotive applications," *IEEE Sensors journal*, vol. 1, no. 2, p. 143, 2001. 1

[3] H. A. Ignatious, Hesham-El-Sayed, and M. Khan, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, 2022. 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare. 1

[4] D. J. Yeong, J. Barry, and J. Walsh, "A review of multi-sensor fusion system for large heavy vehicles off road in industrial environments," in *2020 31st Irish Signals and Systems Conference (ISSC)*, pp. 1–6, 2020. 1

[5] K. Fragkiadaki, S. Levine, and J. Malik, "Recurrent network models for kinematic tracking," *CoRR, abs/1508.00271*, vol. 1, no. 2, p. 4, 2015. 2, 25, 33, 57, 58

[6] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 23, 26, 33, 58

[7] H.-K. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles, "Action-agnostic human pose forecasting," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1423–1432, 2019. 2, 23, 26, 33, 58

[8] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11209–11218, October 2021. 2, 23, 26, 33, 58, 59, 60, 67, 68, 69, 75

[9] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer, and V. Belagiannis, "Motionmixer: Mlp-based 3d human body pose forecasting," 2022. 2, 23, 26, 33, 58, 59, 60, 63, 68, 69, 75

[10] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015. 2, 27

[11] M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," *CoRR*, vol. abs/1710.10000, 2017. 2, 26

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017. 3, 24, 28, 30, 59

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. 3, 24

[14] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," *Advances in Neural Information Processing Systems*, vol. 34, 2021. 3, 24, 29

[15] Q. Zhang, Y. Xu, J. Zhang, and D. Tao, "Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond," *arXiv preprint arXiv:2202.10108*, 2022. 3, 29

[16] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," *CoRR*, vol. abs/2001.08317, 2020. 3, 24, 27

[17] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," *CoRR*, vol. abs/1705.02445, 2017. 10, 17, 23, 25, 32, 33, 68, 69

[18] Y. Tang, L. Ma, W. Liu, and W. Zheng, "Long-term human motion prediction by modeling motion context and enhancing motion dynamic," *arXiv preprint arXiv:1805.02513*, 2018. 10, 17

[19] E. Wu and H. Koike, "Real-time human motion forecasting using a rgb camera," in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–2, 2018. 10, 17

[20] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013. 10

[21] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018. 10

[22] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1–6, IEEE, 2016. 11

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 11, 14

[24] K. K. Singh, S. Kumar, P. Dixit, and M. K. Bajpai, "Kalman filter based short term prediction model for covid-19 spread," *Applied Intelligence*, vol. 51, no. 5, pp. 2714–2726, 2021. 11, 13, 18

[25] A. P. YUNUS, N. C. SHIRAI, K. MORITA, and T. WAKABAYASHI, "Time series human motion prediction using rgb camera and openpose," in *International Symposium on Affective Science and Engineering ISASE2020*, pp. 1–4, Japan Society of Kansei Engineering, 2020. 11

[26] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 12, 13

[27] K. Kruusamäe and L. Tammeveski, "Human detection and distance estimation with monocular camera using yolov3 neural network," 2019. 12

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. 13

[29] R. J. Meinhold and N. D. Singpurwalla, "Understanding the kalman filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, 1983. 13

[30] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001. 13

[31] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 533–549, Springer Nature Switzerland, 2022. 23

[32] G. Singh, S. Akrigg, M. D. Maio, V. Fontana, R. J. Alitappeh, S. Khan, S. Saha, K. Jeddisaravi, F. Yousefi, J. Culley, T. Nicholson, J. Omokeowa, S. Grazioso, A. Bradley, G. D. Gironimo, and F. Cuzzolin, "Road: The road event awareness dataset for autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1036–1054, 2023. 23

[33] C. A. Aubin, B. Gorissen, E. Milana, P. R. Buskohl, N. Lazarus, G. A. Slipher, C. Keplinger, J. Bongard, F. Iida, J. A. Lewis, and R. F. Shepherd, "Towards enduring autonomous robots via embodied energy," *Nature*, vol. 602, pp. 393–402, Feb 2022. 23

[34] P. M. S. Ribeiro, A. C. Matos, P. H. Santos, and J. S. Cardoso, "Machine learning improvements to human motion tracking with imus," *Sensors*, vol. 20, no. 21, 2020. 23

[35] X. Zhao, W. Zhang, T. Zhang, and Z. Zhang, "Cross-view gait recognition based on dual-stream network," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 25, no. 5, pp. 671–678, 2021. 23

[36] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 23, 26, 33, 68, 69

[37] C. Wang, Y. Wang, Z. Huang, and Z. Chen, "Simple baseline for single human motion forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2260–2265, October 2021. 23

[38] I. Chalkidis, A. Jana, D. Hartung, M. J. B. II, I. Androutsopoulos, D. M. Katz, and N. Aletras, "Lexglue: A benchmark dataset for legal language understanding in english," *CoRR*, vol. abs/2110.00976, 2021. 24

[39] N. Safi Samghabadi, P. Patwa, S. PYKL, P. Mukherjee, A. Das, and T. Solorio, "Aggression and misogyny detection using BERT: A multi-task approach," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 126–131, European Language Resources Association (ELRA), May 2020. 24

[40] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *International Conference on Learning Representations*, 2022. 24

[41] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014. 25, 28, 33, 34

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014. 26

[43] B. Wang, E. Adeli, H.-k. Chiu, D.-A. Huang, and J. C. Niebles, "Imitation learning for human pose prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 26, 33

[44] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, pp. 5442–5451, Oct. 2019. 26, 59

[45] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *European Conference on Computer Vision (ECCV)*, sep 2018. 26, 28, 32

[46] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 28

[47] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 28

[48] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d human pose estimation with a single RGB camera," *CoRR*, vol. abs/1907.00837, 2019. 29

[49] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3d human pose regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 32, 33

[50] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 33

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 57, 63

[52] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014. 63

[53] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 68, 69

[54] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 474–489, Springer International Publishing, 2020. 68, 69

[55] W. Mao, M. Liu, M. Salzmann, and H. Li, "Multi-level motion attention for human motion prediction," *International Journal of Computer Vision*, vol. 129, no. 9, pp. 2513–2535, 2021. 68, 69, 75

[56] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li, "Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11467–11476, 2021. 75

List of Publications

Journals

1. Yunus, Andi Prademon, Nobu C. Shirai, Kento Morita, and Tetsushi Wakabayashi. "Comparison of RNN-LSTM and Kalman Filter Based Time Series Human Motion Prediction." In Journal of Physics: Conference Series, vol. 2319, no. 1, p. 012034. IOP Publishing, 2022.

2. Yunus, Andi Prademon, Kento Morita, Nobu C. Shirai, and Tetsushi Wakabayashi. "Time Series Self-Attention Approach for Human Motion Forecasting: A Baseline 2D Pose Forecasting." In Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol.27, No.3, 2023.

International Conferences

1. Yunus, Andi Prademon, Nobu C. Shirai, Kento Morita, and Tetsushi Wakabayashi. "Human Motion Prediction by 2D Human Pose Estimation using OpenPose." In The International Workshop on Frontiers of Computer Vision (IW-FCV) 2020, 2020.

2. Yunus, Andi Prademon, Nobu C. Shirai, Kento Morita, and Tetsushi Wakabayashi. "Time series human motion prediction using RGB camera and OpenPose." In International Symposium on Affective Science and Engineering ISASE2020, pp. 1-4. Japan Society of Kansei Engineering, 2020.

3. Yunus, Andi Prademon, Nobu C. Shirai, Kento Morita, and Tetsushi Wakabayashi. "Comparison of RNN-LSTM and Kalman Filter Based Time Series Human Mo-

tion Prediction." In The 5th International Conference on Engineering Technology 2021 (ICET 2021).

4. Yunus, Andi Prademon, Kento Morita, Nobu C. Shirai, and Tetsushi Wakabayashi. "Temporal-Spatial Time Series Self-Attention 2D & 3D Human Motion Forecasting" 2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT).

Awards

1. Best Presenter Award, ICIT2021, Yunus Andi Prademon, Nobu C. Shirai, Kento Morita, Tetsushi Wakabayashi, "Comparison of RNN-LSTM and Kalman Filter Based Time Series Human Motion Prediction", October 2021.