Human Pose 2D to 3D Uplift and Prediction

Master of Computer Science

Research Project

from

UNIVERSITY OF WOLLONGONG

by

Chris Bunn

School of Computer Science and Software Engineering

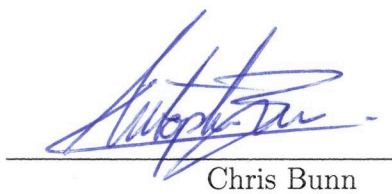
May 2022

Dedicated to

Dr Terence Bunn

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.



Chris Bunn
May 21, 2022

Abstract

Human pose estimation and prediction has many applications from autonomous vehicles to video games development, animation and security. In many instances humans are recorded by video in two dimensions and this two dimensional representation requires uplifting into three dimensions before being fully utilised. This thesis proposes lifting two dimension representations of human motion into three dimensions over a sequence of human action. In addition with the same algorithm and altered training predict the future of that human motion.

The proposed approach builds on the work in HP-GAN [4] utilising a generative adversarial network (GAN) with a recurrent neural network encoder decoder for the generator and a multi layer fully connected neural network as the critic. A novel approach adds random noise from a normal distribution to the z dimension of each joint and a custom loss function consisting of the joint position in 3D space and bone length

The proposed algorithm successfully uplifts 2D motion sequences into their respective 3D motion sequences with an average joint accuracy of 30.6mm and out performs a number of state-of-the-art methods and is within 0.1mm of the best state-of-the-art

model on the Human3.6M skeleton dataset. The proposed algorithm performed better than state-of-the-art in seven activity classes: Discussion, Photo, Sitting, Sitting Down, Smoking, Waiting and Walking Dog. In addition the algorithm predicts realistic future pose motion sequences with acceptable levels of accuracy.

Acknowledgments

Special thank you to my supervisor Associate Professor Wanqing Li for his great guidance and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning has been subject of significant research and development in recent years, most people interact with machine learning knowingly and in many cases unknowingly on a daily basis. From logging into a phone with facial recognition to recommendations on videos to watch and news to read. Machine learning is finding a use in all aspects of modern life.

A recent focus of deep learning research is human motion prediction and motion synthesis, motion prediction is the ability to predict the future flow of a human action from a small sample of the action that has already occurred.

Human motion synthesis expands on the prediction task by introducing the capability to not only predict the future actions but to also generate composite motions. For instance, a motion of sitting to standing is synthesized with walking motion to create a new motion of sitting to standing and walking in a single activity [5]. Another example of motion synthesis is synthesising a walking motion to follow an externally provided path, thus creating a walking motion that is synthesised to suit a specific purpose.

Prediction of human motion is dependent on understanding human joint locations in

3D space, recent large scale 3D datasets [43, 23] are enabling this research. While these data sets provide accurate 3D point clouds of human actions, in real world applications video cameras capture human motion in two dimensions and it is necessary to uplift this motion from two dimensions to three dimensions before making predictions.

Human pose estimation and motion 2D to 3D uplift and prediction has numerous important and very useful applications. Predicting pedestrian movement is an important safety aspect of autonomous driving solutions [14]. The computer gaming and animation industries are seeking to utilise this type of technology to automate parts of their creation and production workflow [24]. Other applications include action recognition [33], robotics/human computer interaction [53] and video surveillance/person re-identification [56] are a number of the numerous areas of research.

Current methods for 2D to 3D uplift typically focus on uplifting a single pose without considering the preceding or following poses in the sequence. This work proposes a method for uplifting human poses from 2D to 3D with a Generative Adversarial Network (GAN), utilising a sequence of 2D poses to construct a representation in 3D space. Further with the same network architecture and modified training the method predicts a 3D uplifted future motion sequence.

The evolution of Generative Adversarial Networks is a key technology in the development of computer vision and motion prediction and synthesis. Training deep learning networks typically requires extensive annotated data. Creating and labeling data is a time and labour intensive task. A GAN learns the training data distribution without then need for annotation, this unsupervised learning enables GANs to be applied to many problems where labeling is not available.

GAN places two neural networks a generator and a discriminator in a minimax game when the generators seeks to generate samples (images for example) from random noise and the discriminator seek to classify these samples as real or fake. The generator learns from the discriminators classifications and seeks to improve its generated images. The discriminator is trained on alternating real and generated samples and tries to maximise the value function while the generator seeks to minimise the value function. In essence the generator learns the distribution of the training samples with the added benefit of being an unsupervised training process. GAN research has progress quickly leading to numerous new architectures and value functions [2]. These new approaches, while improving a number of GAN limitations such as training stability and mode collapse [3], have widened the application of GAN to numerous new applications such as object detection [15], image super resolution [27], medical image segmentation [49] and person re-identification [38].

This thesis is structured as follows, Chapter two is a literature review of the last several years of generative adversarial networks and human pose 2D to 3D uplift and prediction. Chapter three is a review of the HP-GAN approach to human motion prediction and an implementation of this approach and discussion of the results. Chapter four presents the work of this thesis that is, a sequence to sequence GAN for 2D to 3D pose uplift, further utilising the same approach to predict future human motion. The model is trained on two large human pose and action datasets the NTU RGB+D [43] dataset, data is collected with the Kinect camera system. The second dataset is the Human3.6M [23, 6] with data collected via motion capture. Results for 2D to 3D uplift are presented and compared with current state of the art approaches.

Chapter five concludes the thesis. It summarises the advantages and disadvantages of the proposed algorithm and presents some possible future work.

Chapter 2

Literature Review

2.1 Generative Adversarial Networks

2.1.1 Overview

This chapter is a survey of the generative adversarial network literature and non generative methods as it applies to human pose two dimension to three dimension uplift and prediction. In addition this section reviews literature of the related methods utilised in this thesis.

Generative adversarial networks (GAN) were first introduced in [17]. This work establishes the foundation for generative adversarial networks where two networks a generator and discriminator are in competition. The generator to produce fake images images that the discriminator classifies as real or fake. The discriminator is trained to discriminate on a mixture of real and fake images created by the generator, both the generator and discriminator are multi layer perceptions. The generator G takes an input noise vector $G(z)$ and the discriminator D outputs a probability x of real or generated image. D is trained to maximise the classification of the correct label to both real and generated images, G is trained to minimise the discriminators ability to

classify between real and fake (generated images). Essentially the discriminator and generator are engaged in a minimax game, with the value function 2.1.

$$\min_{\mathbf{G}} \max_{\mathbf{D}} V(D,G) = \mathbb{E}_{z\sim p_{data(x)}}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.1}$$

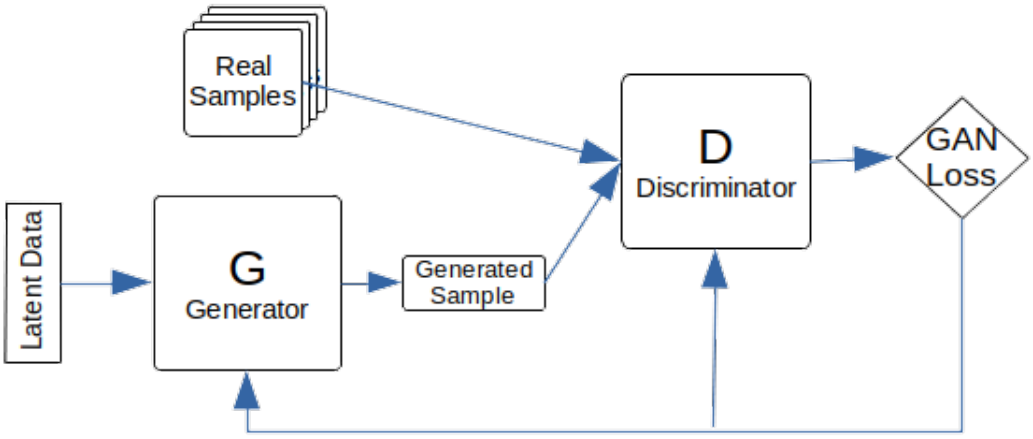


Figure 2.1: General GAN architecture

The authors also describe the training strategy and architecture. Figure 2.1 is the general GAN architecture. The discriminator is trained for given number of steps (a hyper parameter) on both generated and true images. The weights of D are fixed and the generator is trained for a single step with latent data (noise sample) with the same GAN loss function. Training continues for a number of training iterations and reaches optimum when the discriminator ascribes equal probability to both generated and real samples.

Since this introduction to generative networks, research into GANs has progressed quickly with new architectures, training methods and applications growing quickly. The application to computer imagery and imagery generation capabilities is a recent area of focus and research has led to in excess of twenty new GAN architectures and training approaches [2]. The application areas include [2],

- Image generation
- Image repainting
- Super resolution
- Person re-Identification
- Object detection
- Video prediction and generation
- Anime character generation
- Face aging
- Human pose estimation
- De-occlusion

2.1.2 Common Architectures

Four note-able architectures in the development of generative networks for computer vision are Deep Convolution generative adversarial network (DCGAN), Adversarial Encoders (AAE), Generative Recurrent Adversarial Networks (GRNA) and Bidirectional Generative Adversarial Network (BiGAN).

DCGAN

Convolutional neural networks (CNN) have been widely adopted for computer vision tasks, CNN is a supervised learning approach. The authors in [39] propose DCGAN as an unsupervised learning approach to assist closing the gap between the successful supervised learning approach of CNN and unsupervised learning, and in doing so provide a mechanism to utilise unlabeled images and video to image classification and similar tasks. To achieve this the authors propose a set of constraints on the convolutional GAN to stabilise the training. The authors use this trained discriminator for image classification. Through experimentation the authors found and established the following guidelines for stabilising the DCGAN

1. Replace pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator

2. Use of batch normalisation in the discriminator and generator, this is critical to prevent mode collapse in the generator.

3. Relu activation in the generator and LeakyRelu in the discriminator.

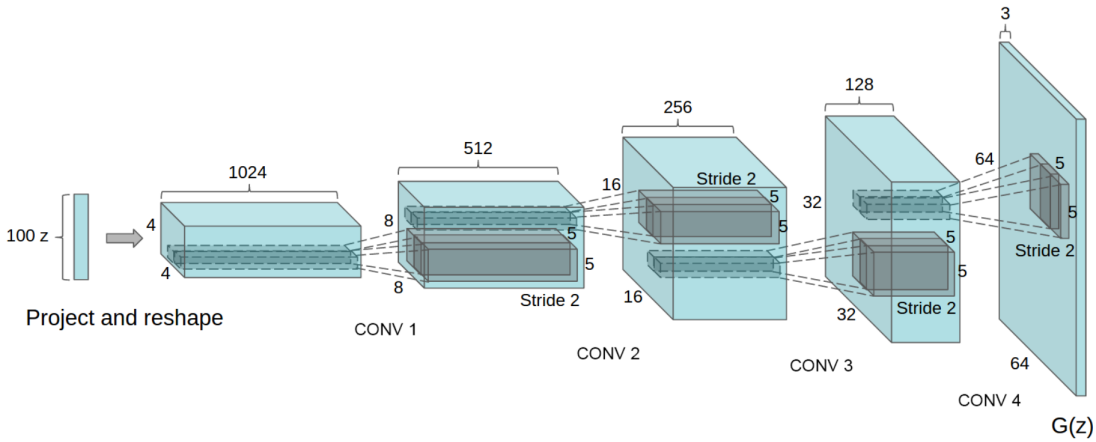


Figure 2.2: Convolutional structure of DCGAN [39]

Figure 2.2 details the DCGAN generator architecture with a 100 dimension input vector and sampling a 64x64 by 3 channel output image. No pooling or fully connected layers are utilised. Using the trained DCGAN discriminator feature set concatenated, regularised and trained with a L2-SVM classifier achieved a 82.8% classification accuracy on the CIFAR-10 [10] dataset, top CNN classifiers achieve 84.3% accuracy. Note the DCGAN was trained on the Imagenet-1k dataset and was used to classify CIFAR-10 samples. A second experiment of vector arithmetic on human faces constructed realistic composite images.

BiGAN

GAN generators learn to map latent space to the distribution of sample data, this learned space is generated by the generator under guidance of the discriminator. What is not learned is the inverse, that is, the mapping from the sample data back to the latent representation. To learn this inverse mapping from the sample data to the latent space the authors in [12] propose the addition of an encoder that maps the data to the latent space. The rest of the architecture is the standard GAN structure of a generator and discriminator. Figure 2.3 details the architecture of what the authors term BiGAN (bidirectional generative adversarial network). The BiGAN consists of two paths, the generator path G mapping latent data z to $G(z)$ in the data space x and the encoder E mapping x (data space) to $E(x)$ in the latent data z space. E learns to invert G and discriminator D discriminates for both E and G. As such objective function is a minimax game as in 2.2.

$$\min_{\mathbf{G},\mathbf{E}} \max_{\mathbf{D}} V(D,E,G) := \mathbb{E}_{x\sim px} \underbrace{[\mathbb{E}_{z\sim p_{E(.|x)}}[logD(x,z)]]}_{logD(x,E(x))} + \mathbb{E}_{z\sim p_z} \underbrace{[\mathbb{E}_{x\sim p_{G(.|z)}}[log(1-D(G(x,z))]]}_{log(1-D(G(z),z))} \tag{2.2}$$

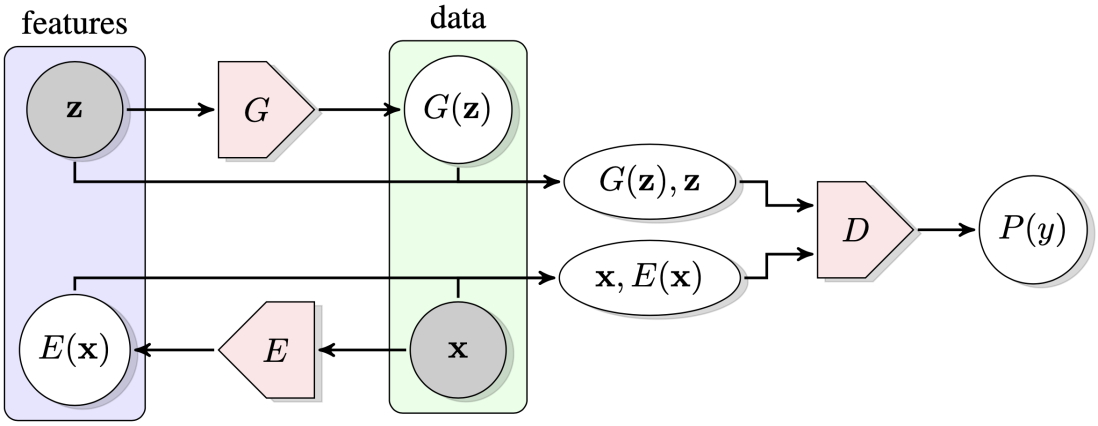


Figure 2.3: BiGAN architecture [12]

2.1.3 GAN Applications

Image In-painting

Image in-painting is the reconstruction of missing areas of an image and removing unwanted parts of an image. Current deep learning techniques as observed by the authors in [36] generally introduce artefacts such as over smoothing and blurring. To address these issues the authors propose a two stage adversarial network, the first stage is an edge generator that overlays an edge on the missing regions of the image, creating edges for the missing regions. The second stage fills in the missing sections of the image using the overlay edge as a prior. Each of the two generative networks consist of a generator and a discriminator combined into an end to end trainable network, figure 2.4 details this architecture.

The edge detector takes as input a gray scale image, edge map of the edges that

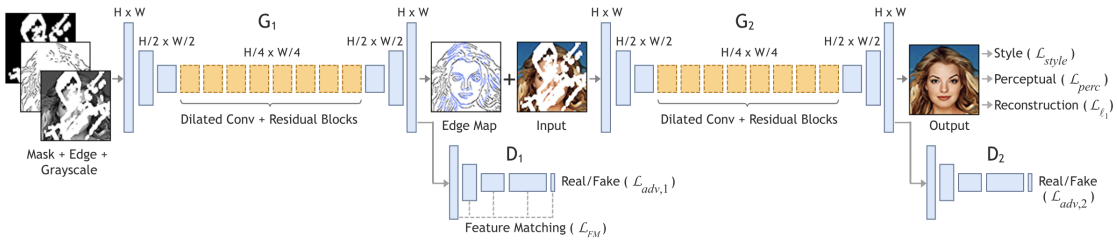


Figure 2.4: Image inpainting dual GAN architecture [36]

are present (created with canny edge detector) and image mask of missing region (1 = missing, 0 = background). The discriminator predicts if the edge map is real. Output is a composite edge map of the known edges and the predicted edges for missing regions. The objective function is composed of typical GAN loss [17] and a feature mapping loss. The second network or image completion network takes as input the composite

edge map and the colour incomplete image (image to be repaired) and outputs the colour image with the missing regions filled. This network is also trained with GAN loss and addition of two regulators, a perceptual loss and style loss.
The authors demonstrated the capability and generalisation capabilities by training on the CelebA [32], Places2 [57] and Paris Street View [11] datasets. The authors conducts qualitative tests using a visual Turing test and quantitative Frechet Inception Distance (FID) [20], the proposed approach out performs contemporary methods.

Person re-identification - pose normalisation

Tracking people across multiple non overlapping camera views is a difficult problem know as re-identification. An individuals appearance can change across views due to a number of factors, viewpoint, lighting, body orientation/configuration and occlusion. Body orientation, configuration or pose which the authors of [38] assert, to learn an effective re-identification model it is necessary to remove a persons pose from their appearance, allowing the re-identification model to learn with less data and make the model scaleable to large camera networks. To address these issues the authors are proposing a generative model that produces pose normalised images which is termed PN-GAN (pose normalisation GAN). PN-GAN learns to change the poses but keep the identity of the individual.

Figure 2.5 is the PN-GAN model as with GAN models in general it consists of a generator and discriminator. The generator "ResNet" is an encoder decoder network, it has as input a concatenation of an image I_i of the person and a skeleton I_{Pj} of the desired pose and outputs an image of the person in the I_{Pj} pose. The discriminator

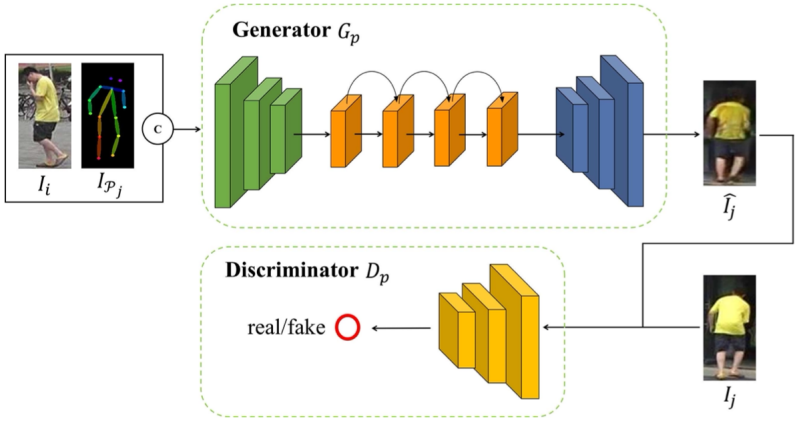


Figure 2.5: PN-GAN architecture [38]

performs the typical discriminator function, differentiating between real and fake output images. Both the generator and discriminator use the GAN [17] loss functions with the addition of a regulariser in the generator loss function. Loss is the $L_1 - norm$ of the generated image I_j and the ground truth I_j. With the generated image and original image a further two networks are trained to learn the features of each, the resultant features from each network are fused to create a re-identification feature set. Testing on four difference data sets and results based on accuracy and mean average precision this method achieved state of the art performance when compared to numerous other re-identification techniques.

Game and image generation

Video game development and anime character generation GANs are beginning to perform tasks of human animators, auto generating and creating colour cartoon characters. From these application areas new GAN methods and approaches are developing. In [24] the authors are creating new anime character faces from past samples and a modified SRResNet (Image resolution increasing model) network as the generator and modified

Figure 2.6: Generated anime faces [24]

discriminator. The model creates engaging and plausible anime facial images, figure 2.6

In [52] the authors improve on the original GAN structure with a Self Attention GAN (SAGAN), this broadens the image generation from spatially local points of lower resolution feature maps to generating cues from the complete feature map. In addition to the self attention module the authors also explore additional techniques to stabilise GAN training by adding spectral normalisation not only in the discriminator but also in the generator, it was found this also reduced the computation cost of training. Secondly to improve training time, different learning rates were utilised in the generator (0.0001) and discriminator (0.0004) reducing the number of discriminator updates per generator update. Testing on ImageNet dataset, SGAN improved upon state of the art inception scores from 36.8 to 52.5 and reduced the Frechet inception distance down to 18.7 from 27.6. Samples of SAGAN generated images are observed in figure 2.7.

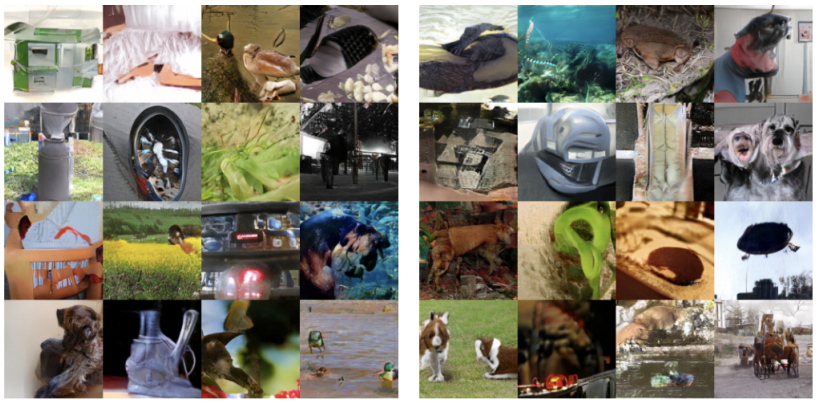


Figure 2.7: SAGAN image samples [52]

Human pose estimation

Human pose estimation has been a subject of computer vision research for some time and recently human pose prediction is attracting research attention. More recently generative networks research has been applied to this problem.

In [46] the authors propose to solve the issue of generating multi view point images, for example viewing a face from multiple angles with only sampling a single view point during training. The authors assert the "single path way" approach of GAN architectures inhibit the complete learning necessary for multi view point images, especially for unseen samples. The authors testing demonstrates that an unseen sample in a single pathway network is not mapped into the latent space of the network. To resolve the problem the authors are proposing a new GAN architecture termed Complete Representations GAN (CR-GAN). CR-GAN architecture consists of two generator / discriminator networks which share weights. One networks generator is feed with an encoder, this network servers as the reconstruction path. Training the encoder and discriminator holds the generators weights fixed. The generation path (second path, generator and discriminator) trains the generator on latent data input. Figure 2.8

details the CR-GAN dual pathway architecture.

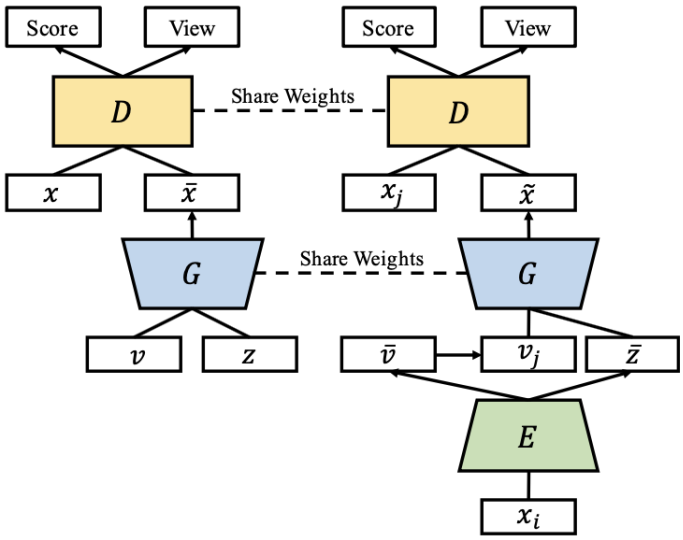


Figure 2.8: CR-GAN Architecture [46]

The proposed CR-GAN was compared against both single pathway GANSs, BiGAN [12], DR-GAN [47] and a dual pathway network TP-GAN [21]. Testing was on the Multi-PIE [18], CelebA [32] and IJB-A [25] image datasets. On visual inspection of the generated images CR-GAN out performed all these these current methods.

2.1.4 GAN Training and training advancements

In [17] the authors define the original GAN training algorithm based on mini batches and stochastic gradient descent. The algorithm simultaneously trains the discriminator D(x) on alternating samples of real data and samples from the generator G(z) trained on the random vector z. Training is considered optimal when the discriminator is not able to distinguish between D(x) and G(z), that is the probability of D(x) = $\frac{1}{2}$. Algorithm 1 as defined in [17] is the general GAN training method along with GAN architecture as depicted in Figure 2.1.

Algorithm 1 Minibatch with stochastic gradient decent training loop for the GAN model. k is a hyper parameter which governs the number of discriminator training steps per training loop

for number of training steps **do**
  for k steps **do**
    Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
    Sample minibatch of m examples $\{x^{(1)}, ..., x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
    Update the discriminator by ascending its stochastic gradient:

$$\bigtriangledown_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [logD(x^{(i)}) + log(1 - D(G(z^{(i)})))]$$

  end for
  Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$.
  Update the generator by descending its stochastic gradient:

$$\bigtriangledown_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^{(i)})))$$

end for

GANs training can be difficult to train as the cost function 2.1 is a minimax game where convergence is achieved when both the discriminator and generator costs are at a minimum. As the generator minimises its cost the discriminators cost increases and an optimum may not result and the training will fail.

In addition GANs are sensitive to hyerparameter setting and require considerable experience and tricks to avoid mode collapse [7]. A common failure is mode collapse when a weight of probabilities mass onto a small number of modes. Resulting in the generator only generating a very small number of samples with respect to the number used in training. As the discriminator tends to be close to 1 for real examples, large modes create gradients within the discriminator such that the generator collapses to a few larger modes to represent a single mode. The discriminators output is near 1 and the generator is not penalised for the collapse of multiple modes into one. The authors of [7] propose systematic methods to measure missing modes to stabilise training via metric and autoencoder based regularises. Two regularises are proposed.

1. Geometric Metrics - adding additional geometric distance measures, to measure the distance between the two manifolds (real and generated) and then minimise this distance. A subject of this thesis HP-GAN which will be discussed further, implements two geometric metric regularisers, a pose to pose consistency and bone length consistency loss metric.

2. Mode Regulariser - The addition of a regularises (autoencoder) that encourages the generator to move towards modes that have sufficient examples in the data but is not a major mode.

In addition [7] proposes a two step training method, a manifold step and a diffusion step. (1) manifold, training with both geometric and mode regularises. (2) Diffusion step, attempts to match distribution probability mass in the generating manifold according to the real data manifold.

To address the issue of training stability the authors in [3] propose a new approach approach to stabilising training termed Wasserstein GAN (WGAN). This approach is based on the "Earth Mover" distance, the measure of distance between two probability distributions and weight clipping to keep weights within a defined space. The critic network C predicts the Wasserstein distance and seeks to minimise the loss equation 2.3 or distance between the real distribution x and generated distribution $G(x)$.

$$L_{wgan} = -C(x) + C(G(X)) \tag{2.3}$$

In addition the training algorithm trains the critic (discriminator) before the generator update, this allow the loss function to be an estimate of the earth mover distance before the generator update. Using DCGAN [39] as the baseline the authors were

able to train with a stable loss curve and high quality image generation. The critic is trained to an optimum point and then operates as a loss function in the generator. This reduces the complication of needing to balance the discriminator and generator and this simplifies hyper parameter selection. Notably, no mode collapse occurred during training.

Developing upon [3] the authors of [19] propose to replace weight clipping with a gradient penalty. While Wasserstein GAN (WGAN) has made progress in stabilising GAN training and significantly reducing mode collapse, GANs are still prone to failing to converge and producing poor samples. The authors assert this is due to the weight clipping to contain the weights within the space of 1-Lipschitz function.

To address the problems experienced due to weight clipping the authors propose a gradient penalty in place of weight clipping. The authors termed this method WGAN-GP. Under testing where the generators distribution is held constant plus Gaussian noise, it is observed that weight clipping pushes the weight to the extremes of the clipping values resulting in the discriminator learning simple approximations of the optimal functions. In addition it was observed without careful selection of the clipping values WGAN experiences either vanishing or exploding gradients.

The gradient penalty replacing weight clipping, the gradient penalty is the norm of the gradient with respect to the critic input. The gradient penalty (GP) is calculated in equation 2.4.

$$GP = (\|\bigtriangledown_x D(\hat{x})\|_2 - 1)^2 \tag{2.4}$$

$$\hat{x} = \epsilon x + (1 - \epsilon)G(z)) \tag{2.5}$$

ϵ a random number $U[0, 1]$, z a latent variable

The new loss function as described in equation 2.6

$$L = D(G(z) - D(x) + \lambda(\|\bigtriangledown_x D(\hat{x})\|_2 - 1)^2 \tag{2.6}$$

λ is a hyper parameter set to 10.

The authors demonstrate this method on a series of 200 variations of the DCGAN [39] architecture by varying the activation functions, network depth, batch normalisation and filter counts. WGAN-GP out performed the standard GAN objective function as rated by inception scoring [42]. Testing on the LSUN bedrooms data set [50] and comparing against six other variations of DCGAN [39], GAN and LSGAN [34] architectures with common hyper parameters. Only WGAN-GP was able to be successfully trained and consistently producing clear generated images. In addition no mode collapse was experienced.

2.2 Human Pose Prediction

An area of particular interest is human posed prediction where the future human body configuration is predicted from sequence of human poses. The authors of [4] propose a sequence to sequence GAN model involving three individual neural networks, a sequence to sequence encoder/decoder generator, a critic network which performs the function of what is commonly termed the discriminator in the generator/discriminator structure. The third network is a classifier described as the discriminator. The discriminator provides a mechanism for scoring the generated sequences by way of likelihood of the sequence bring real, it does not contribute to the training of the generator. The authors propose a modified Wasserstein GAN in combination with a custom loss

function. This work will be explored further on in this document.

Further to pose prediction which typically occurs over shorter sequences, research into methods that are able to predict motion over longer periods and wider activity set is very active.

In [5] the authors seek to synthesise long term human motion over a large set of human actions. To achieve this the authors are proposing a two stage generation method, the first stage learns to synthesize a motion trajectory and the second stage takes this input and generates dense motion trajectories. The method is able to synthesise between two activities to create a composite motion sequence. The example given by the authors is using a drinking sitting down activity and sitting down to standing up activity to generate a sitting down to standing up and drinking activity. Another aspect to this method is the ability to control the speed of motion trajectories giving different speeds at which activities are performed.

The first stage "GloGen" is a Seq2Seq network with bi-directional LSTM, it takes as input a sparse set of human poses concatenated with their action label as a single human pose vector. All states of the encoder are passed to the decoder along with the predicted poses from the encoder. The output of the encoder is a predicted sparse global motion. The second stage "LocGen" has the same architecture as the GloGen network, it's objective is to fill the sparse sequence from GloGen with dense motion. The encoder is feed with euclidean interpolated poses embed into a higher dimension. The hidden states are feed to the decoder (without class priors) and outputs the dense motion to fill in the sparse motion. Figure 2.9 details the architecture and stages of the GlocalNet network.

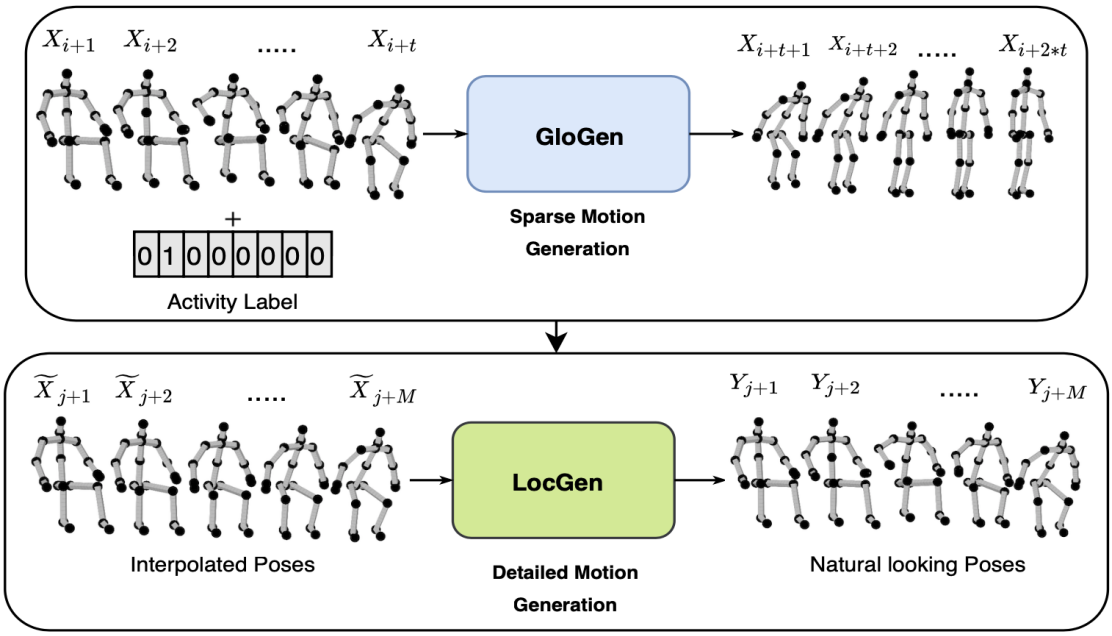


Figure 2.9: Architecture of GlocalNet with the two stages GloGen and LocGen [5]

Using the Maximum Mean Discrepancy (MMD) metric, often used for measuring the quality of motion sequences by measuring the similarity between generated sequences and their corresponding ground truth. The results demonstrated lower MMD_{avg} (average per frame) and MMD_{seq} (average over the sequence) against seven other methods on the Human 3.6M data set [23] Using a second measure euclidean distance, as used in [30], measures the error as the distance between generated sequence and ground truth on a frame by frame basis. The authors method out performs current methods on the CMU data set [1] especially over longer sequences with multiple action types.

The authors in [48] propose to resolve the problem of artefacts entering generated motion sequences as a result of RNN prediction errors being passed along in the predicted sequence. An example of such an artefact is foot sliding when the foot position between frames changes giving the effect of it sliding along the floor. Another such

artefact is a sequence converging to a static pose [35, 22] especially when predicting long sequences. The authors propose to predict future human motion from past recorded motion data with the added information of embedding contact information to improve the quality of the generated motion sequences. The proposed model consists of multiple components, a state feature representation, which takes each pose within the context of the root joint, root joint angles and the joints of the remaining joints. The authors propose a three network model and a RNN generator that generates synthesised motion a discriminator and a GAN model the authors call a refiner. The objective of the refiner/discriminator combination is to smooth or add realism to the RNN (LTSM) generated sequences. To aid in these more realistic refinements, the authors propose a contact aware learning model that understands foot fall contact and adds this information to the motion features and utilises this in the learning to improve the quality of the motion synthesis in the refiner GAN network. The effect of this is to prevent a foot in contact with the floor surface slipping before it lifts from the floor. The proposed model is able to generated random motion, for instance walking motions in a generated path. The model also supports a user defined input such as a path the motion is to follow, producing a synthesised walking motion following the prescribed path. The authors demonstrated their refiner GAN, smoothing the motion to be much closer to the data set [1] than just generated motion alone. In user studies where users were asked to rate the quality of the generated motion, the authors method was rated favourably in comparison to motion capture.

2.2.1 Sequence to Sequence

An important capability for making predictions from a sequence of priors is sequence to sequence learning. In [45] the authors propose an approach for sequence to sequence learning of variable length sequences. Sequence to Sequence learning is challenging in DNN's as they require a fixed or known dimensions. The approach uses an encoder - decoder approach with a multi layer LTSM structure. The model reads a sentence until a token is reached, indicating the end of the sentence, this sequence is read into a two layer LTSM encoder and then into a two layer decoder. It was found a deep model (4 LTSM) outperformed shallow 2 layer network. Training maximises the log probability of a translation given the source sentence. The translation is determined by the highest probability translation. The challenge with sequence training sequences is lengths can vary, for instance a sentence can be a few words through to tens of words long. The authors test the proposed method between English and French sentences. The results were comparable with current non neural network based approaches, not only on short sentences but the model performed well over longer sentences. An additional discovery was the reversing of the input sentences improved the results further.

2.2.2 Estimation of 3D from 2D representation

The creation of 3D pose information requires specialised motion capture or camera equipment, this can be expensive and time consuming to organise subjects to perform actions and record the data into 3D skeleton pose representations. Considerable image and video data exits for the generation of 2D human pose information, can this 2D data be projected to 3D skeleton pose representations. The methods for 2D to 3D

representation are categorised into three categories [13], "Fully Supervised", "Weakly Supervised" and "Learning Using Adversarial Loss". Fully supervised, utilises a paired 2D-3D using the 3D data for learning. Weakly Supervised, utilises unpaired 2D-3D data using 3D data for learning but unpaired 2D data for projection to 3D representations. Adversarial loss learning utilises a generator to synthesise a 3D representation from a latent sample and discriminator to decide between real and fake.

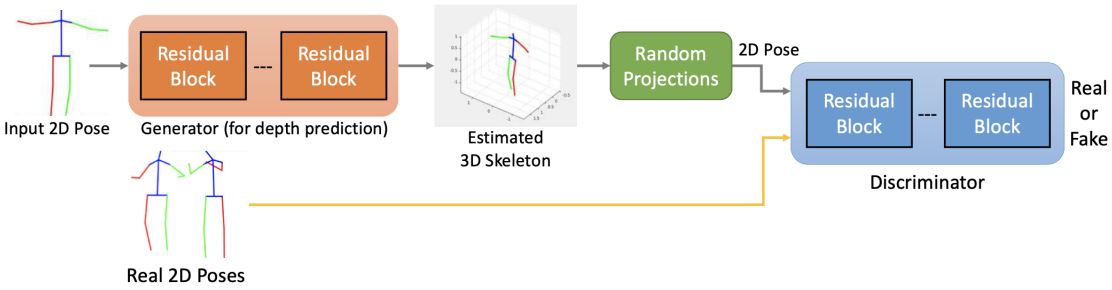


Figure 2.10: 2D to 3D pose generative model [13]

In [13] the authors propose a weakly supervised learning model, that inputs a 2D representation and generates a 3D representation from a random latent projection that is converted back to 2D projection for evaluation by the discriminator. Figure 2.10 details the architecture of the proposed generative model. The generator G produces an output offset equation 2.7, o_i is the offset for each point x_i. Then the depth z_i is calculated (equation 2.8) where d is a hyper-parameter representing the distance from the camera. A back projection layer then computes the 2D point into 3D space using the predicted z_i.

$$[ht!]G(x_i) = o_i \tag{2.7}$$

$$z_i = max(0, d + o_i) + 1 \tag{2.8}$$

Next a Random Projection Layer projects the 3D representation using random camera orientations back to a 2D representation for input to the discriminator D. The discriminator D classifies the fake 2D representation as either fake or real $[0, 1]$. The model is trained with the standard GAN objective function [17]. The proposed weakly supervised model outperformed most of the comparable 2D to 3D lifting methods in bench marking. Further the model compares well against fully supervised models while not outperforming the top performing methods.

The authors of [28] assert, collected 3D datasets are typically biased toward indoor environments where they are collected and the actions are day to day actions such as typing on a keyboard, using a phone and as such models, learn these biases and do not generalise to "in the wild" situations. The authors propose a two stage

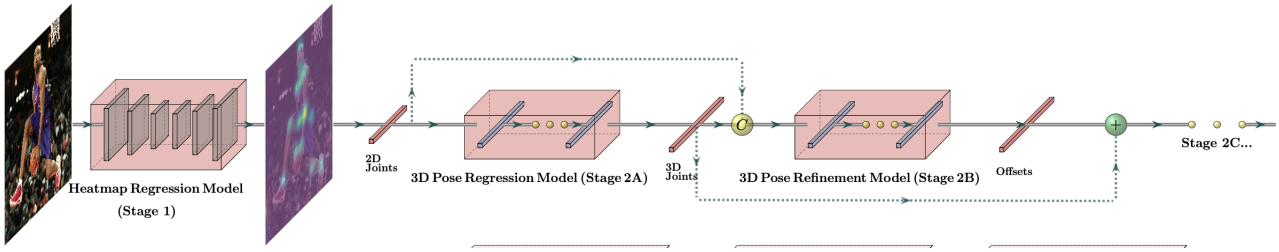


Figure 2.11: TAG-Net architecture [28]

architecture, the first stage locates the 2D joints and bone locations, the second stage projects this 2D representation into 3 dimensions, this architecture the authors term TAG-Net, "Transits from Appearance to Geometry". To address the dataset biases the authors propose an evolutionary framework to manage the limitations of training data. The evolutionary frame work consists of three components, a crossover operator which given two 3D skeletons assembles two new skeletons from the limbs of the two original skeletons by randomly selecting bones from the set of original skeletons and

attaching them to the new skeletons while maintaining the overall skeleton kinematics. The second method is a mutation operator modifying the orientation of a bone vector with the addition of noise into vector creating a new pose, for example, swinging a leg forward. The third component is a "Natural Selection" fitness function determining skeletons pose validity. These processes are repeated to obtain a valid set of new skeletons. Figure 2.11 details the architecture of the weakly supervised TAG-Net approach. The first stage is a high resolution heat map using HR-Net [64] with some changes to reduce errors localising key points, namely the joint locations. The second stage consists of a full connected DNN regressor predicting 3D representation from the 2D poses, this feeds a second fully connected DNN refiner network fine tuning the positioning of the 3D coordinates. Testing with the Human 3.6M dataset [23] using Mean Per Joint Position Error (MPJPE) and MPI-INF-3DHP (3DHP) a benchmark to evaluate 2D to 3D model ability to generalise. When compared against state of the art models [41, 26, 37], TAG-Net out performed these weakly supervised models. The same result was experienced with the comparison against five state of the art fully supervised models. TAG-Net outperformed all but one OriNet [35] (MPJPE measure) of the models in 3DHP generalisation testing.

In [8] the authors propose an unsupervised learning approach to estimate 3D pose information from 2D pose information. Their approach does not require 2D-3D paring or any 3D prior input. The authors generative approach trains a 2D to 3D lifter neural network, the 3D projections are randomly transformed and re-projected back to a 2D projection. This re-projection is feed to a discriminator for real or fake determination and back through an inverse of the random transformation for comparison with the

original 2D representation. Figure 2.12 details this high level architecture, the architecture contain three loss functions $\mathcal{L}_{adv}$, $\mathcal{L}_{2D}$, $\mathcal{L}_{3D}$ and a forth not depicted temporal consistency $\mathcal{L}_T$. Training updates network parameters to optimise these four losses as in equation 2.9

$$\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{2D} + \mathcal{L}_{3D} + \mathcal{L}_T \tag{2.9}$$

$\mathcal{L}_{adv}$ is the standard GAN loss [17] function with a regulariser to limit corrections to a small amount. $\mathcal{L}_T$ also utilises the GAN loss function between consecutive pairs of real 2D poses and generated fake 2D poses. $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$ is the $\ell^2 - norm$ of the real and generated pose. Testing on the Human3.6 and Kenetics datasets and

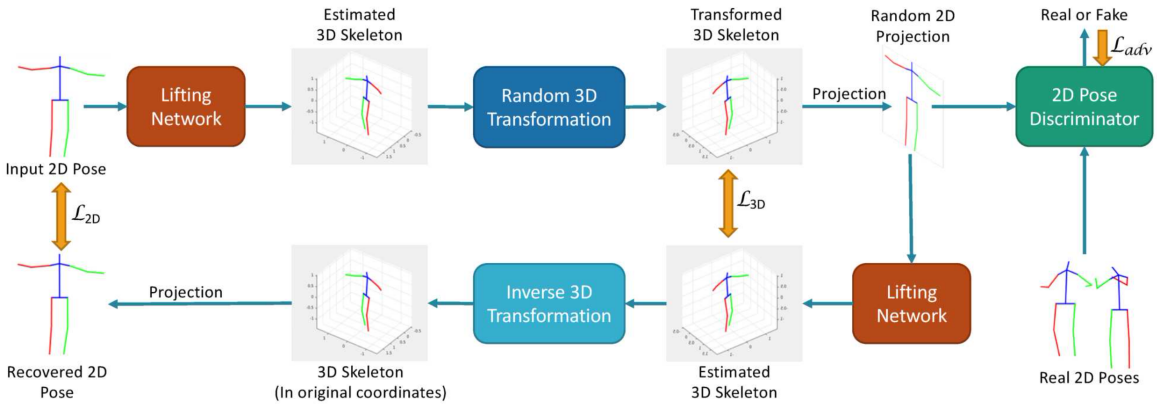


Figure 2.12: Generative 2D to 3D lifting network [8]

using Mean Joint per Joint Position Error (MPJPE) with a subset of the number of subjects in the dataset. The authors model reduces error by 30% over state of the art [40] unsupervised model and out performs a number of weakly and fully supervised approaches. Interestingly the authors introduced a 5% Human3.6M3D [23] prior to the training and increased the performance to be comparable with fully supervised methods.

Semi and fully supervised 2D to 3D pose estimation is dependent on the availability

of suitable matched 2D to 3D pose data. In [44] the authors propose a multi-view pose generator to provide different rotational 2D views on a 2D pose generating additional 2D pose inputs. In addition a Graph Convolutional Network (GCN) projects the multi-view 2D to 3D pose estimations. The authors also propose a loss function consisting of bone length and joint location constraints. Figure 2.13 is a high level depiction of the end to end model. The Multi-view Pose Generator (MvPG) generates multiple views of a single 2D pose, this network consists of a fully connected network which takes 2D skeleton input and rotated 2D representations of the the 3D ground truth with mean squared error loss (MSE). The 3D to 2D is completed with a rotation operation fixes the y joint position and x, z rotated with matrix multiplication and sampling rate for the number of projections required. The MvPG network is a fully connected network

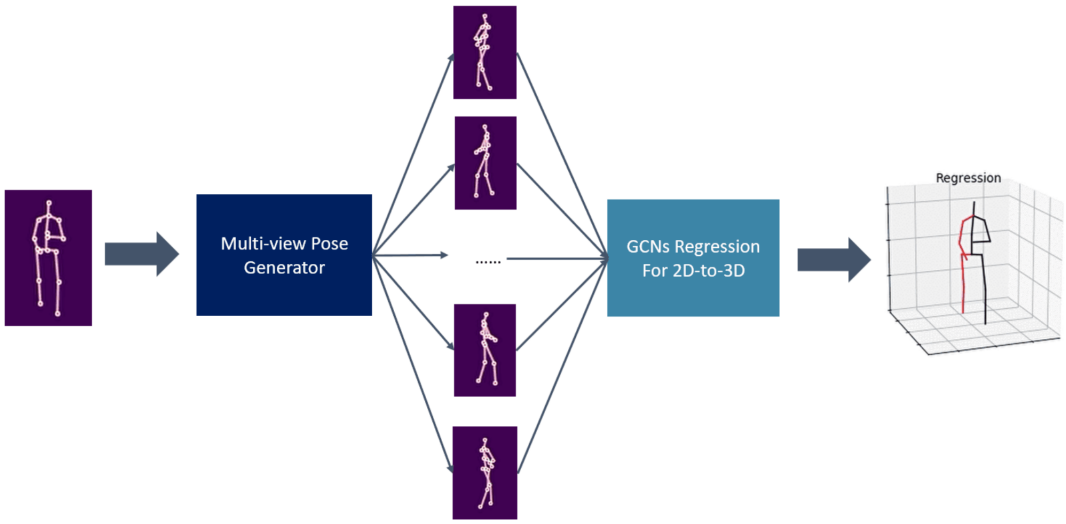


Figure 2.13: MvPG framework [44]

which is pre-trained and mated with a SemCGN [54] network for 2D to 3D projection, trained with MSE loss of joint position and bone length. MvPG improved the SemCGN [54] by 1.3 percent and outperformed a number of state of the art methods across a

number of measurement dimensions when tested on the Human3.6M [23] dataset.

In [31] the authors propose a model which captures the structure of the human body beyond the typical tree like structure of joints that connect with each other in a fixed way that represents a simple graph. The model uses a hypergraph and semi-dynamic hypergraph structures that allows joints to be connected beyond the fixed assembly of body joints but rather by the dynamic interaction between the joints to capture the kinematics of the body. For instance, consider the pose sitting, unconnected joints like a knee and elbow can be close, the dynamic hypergraph allows these joints to be connected. The authors developed a method for determining the relationship between joints in the feature space based on distance. Both the static and dynamic hypergraphs are inputs to a hypergraph convolutional network as in figure 2.14, with a formula combining dynamic and static hypergraphs with a regression loss function. Using

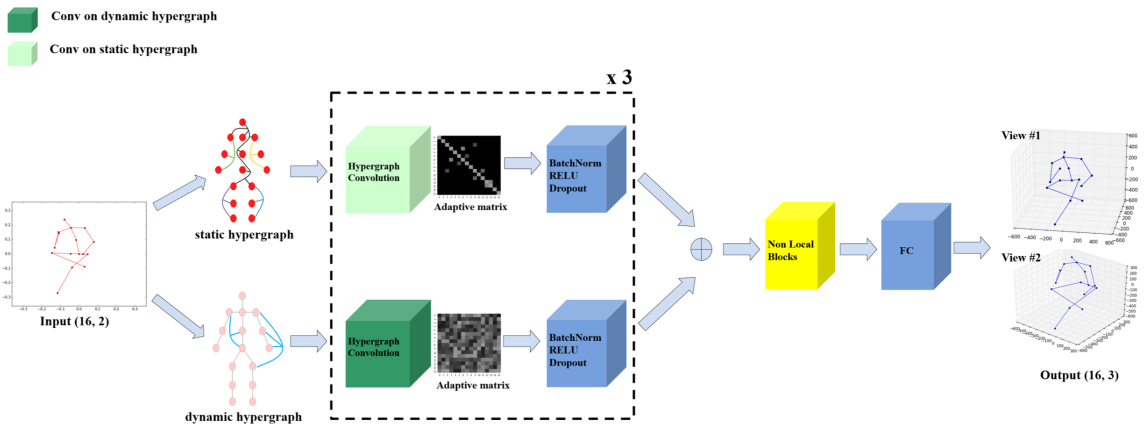


Figure 2.14: SD-HNN model framework [31]

mean per joint position error (MPJPE) measure with the Human3.6M [23] dataset the model outperformed state of-the-art models. Further, with qualitative review the model deals with complex actions and occlusion, producing accurate results.

Chapter 3

HP-GAN and Implementation

This chapter reviews the HP-GAN [4] model for the probabilistic prediction of 3D human motion. The chapter includes introduction of the concept including an overview of the model, description of the various components, loss functions and training strategy. As part of the review into HP-GAN, the model and testing is reproduced and the results presented.

3.1 Overview

The ability to predict human motion in an important development and necessary capability in a number of fields, animation, computer games, security, autonomous vehicles and motion synthesis. In daily human life the ability to predict the actions of people and objects is critical to navigating safely through day to day activities. For instance while driving mentally predicting the likely actions and path of pedestrians is critical, is a person likely to try and cross traffic or a child likely run into traffic. Less dramatic examples such as, is a person likely to cross your path while walking, predicting the path of another persons hand for a hand shake. Even more complex is predicting the outcome of movement when multiple outcomes are possible, such as does the person

catch or drop the ball. The authors propose HP-GAN to predict the short term future motion of a human based on the immediate past motion. Further HP-GAN predicts multiple future motions providing a motion quality assessment of the future predicted motions.

HP-GAN implements a generative adversarial network to predict the future motion from a variable number of human pose sequences and predicts a variable number of future poses. HP-GAN implements a three network architecture consisting of a generator, critic and discriminator, where the generator/critic learn and predict human motion and the discriminator is trained to predict a metric that represents the probability a predicted motion is human.

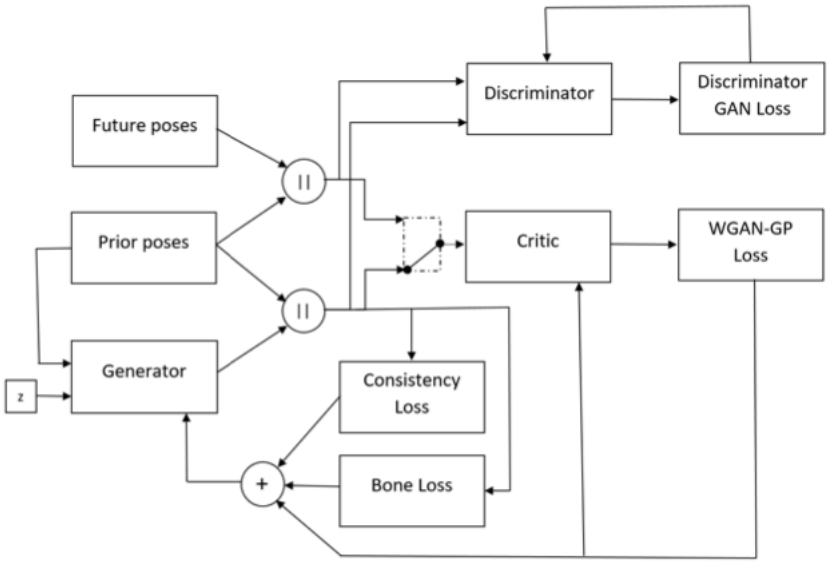


Figure 3.1: HP-GAN Model [4]

Figure 3.1 is the architecture of the HP-GAN depicting the three neural network elements in the model. Each neural network will be discussed in more detail in section

3.2. In addition to the architecture the authors propose a modification to the WGAN-GP [42] loss function with two geometric distance measure regularises designed for human motion. Figure 3.2 depicts the four individual loss function elements and their respective use in the HP-GAN architecture, the loss functions are discussed further in section 3.3. Future poses and prior poses are the sequence of human motions, these sequences are 3D skeleton representations of humans during an activity such as walking. Prior poses are inputs to the generator from which predictions are made. Future poses are the ground truth sequence of motion that follows the prior poses.

3.2 Network Components

3.2.1 Generator

The generator is a sequence to sequence recurrent neural network implemented in a encoder decoder structure. Figure 3.2 depicts the encoder/decoder structure of the sequence generator. Prior poses are input to the encoder, it is important to note in this diagram the uniform noise vector z is added to the states of the encoder, these states then serve as the initial states for the decoder. This is a notable difference to other models where the noise vector is added to the data. ((cite language transaltion and text to image papers)) The last sequence from the encoder is the first data input to the decoder. The decoder then generates the predicted future poses.

Figure 3.3 is the internal layer view of the generator. It consists of a Linear layer that transforms the pose sequence into a 128 feature vector followed by the recurrent layer consists of two 1024 unit GRU cells. The noise z input is a 128 dimension vector

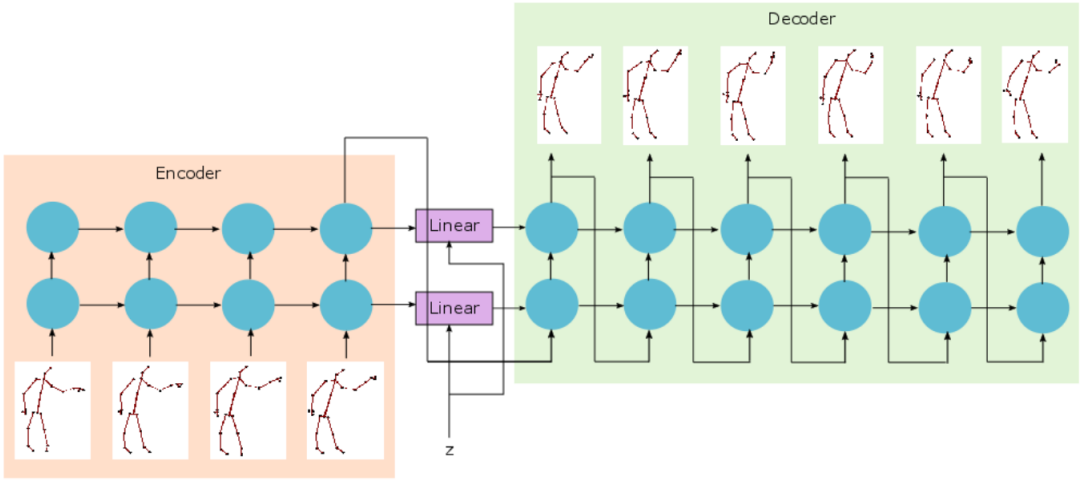


Figure 3.2: Sequence to Sequence representation of the model [4]

drawn from a uniform distribution and mapped to the encoder output state dimensions of 1024 units and added to the output states of the encoder. This addition initialises the input states of the decoder. The last data output sequence of the encoder is the first input to the decoder. The decoder is a 2 layer RNN consisting of 2 GRU cells and a final fully connected layer mapping the sequence to the required output dimensions to form the predicted sequence.

3.2.2 Critic and Discriminator

The critic and discriminator share the same architecture, they are fully connected multi level perceptron, with 5 layers as detailed in figure 3.4 they only differ in the loss functions used and the activation function in the final layer, discriminator sigmoid and critic no activation. Citric loss function is as defined in section 3.3.1 and the discriminator is the GAN [17] loss function as in equation 2.1 with the addition of a weights regulariser (4.6). Input to the critic is an alternating sequences of ground truth poses (concatenation of prior and future poses) and generated sequence of poses

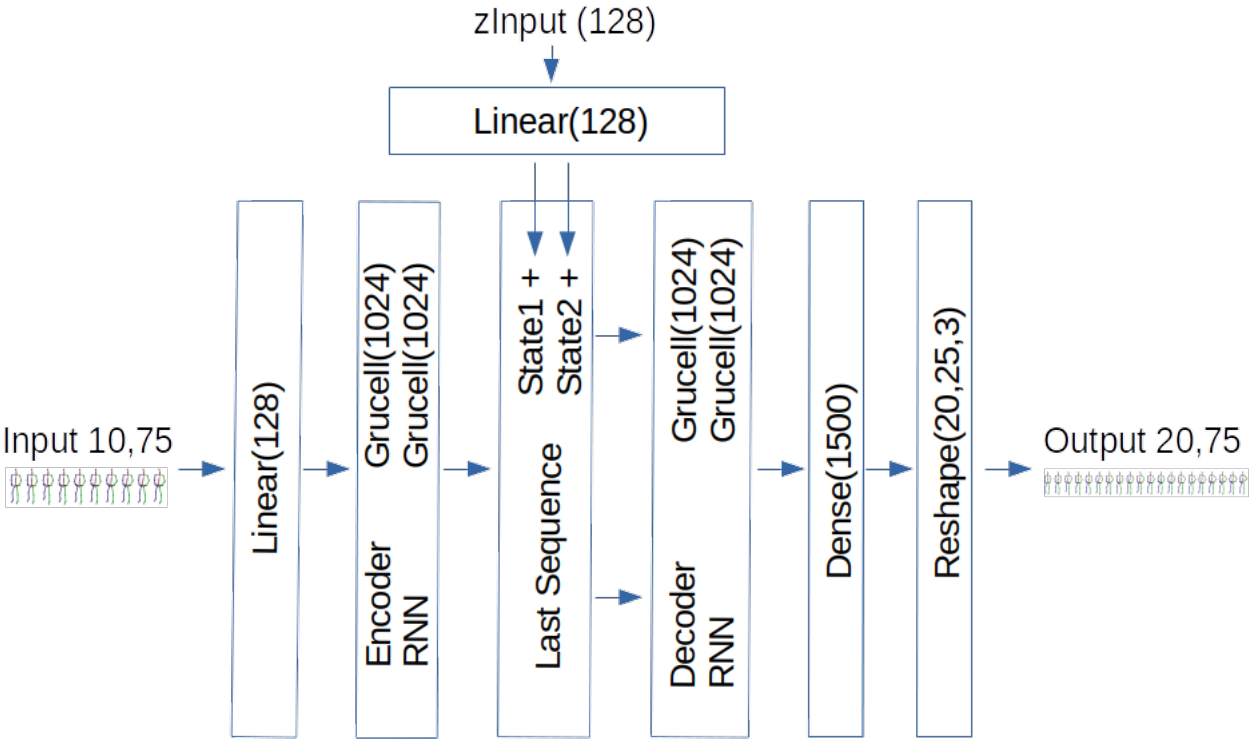


Figure 3.3: Internal layers of the generator ((redraw this image))

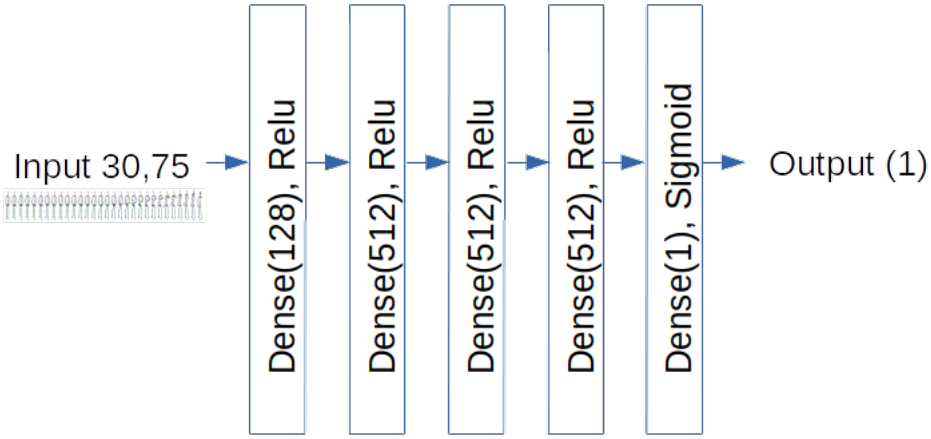


Figure 3.4: The layers within the critic and discriminator model

(concatenation of prior poses and generated sequences), this is visualised in Figure 3.1 by the switch in the centre of the diagram. The output of the critic is the WGAN-GP loss value which is used in the generator loss function, these loss functions are detailed in section 3.3.

The discriminator determines the quality of predicted motion sequences. The discriminator in HP-GAN is not involved in the training of the generator and is separate to the GAN structure consisting of the generator and critic. Input to the discriminator are both the sequence ground truth (concatenation of prior and future poses) and generated sequences (concatenation of prior poses and generated sequences) the output is a single probability value. The greater the probability the higher the classification of the sequence as a real. For instance a probability of 0.9 indicates the generated sequence is being judged very likely a real sequence.

The architecture of the critic and discriminator in figure 3.4 consists of 5 layers a linear 128 dimension input layer followed by three fully connection 512 dimension output layers with relu activation. The output layer is a fully connected single dimension output, the discriminator with sigmod activation and linear activation in the critic.

3.3 Loss Functions

In addition to the neural networks, figure 3.1 also details four loss functions. GAN used by the discriminator, WGAN-GP utilised by the critic and in combination with two geometric regularisers consistency loss and bone loss utilised by the generator. This section details these loss functions and their roles in the HP-GAN model.

3.3.1 Critic

The critic loss L_c consists of three components:

$$L_c = L_{wgan} + \lambda L_{gp} + \alpha L_2 \tag{3.1}$$

L_{wgan} is the WGAN loss [19], which is critic's returned value of the concatenated prior poses x with the generated future poses $G(x, z)$, less the critics returned value the corresponding ground truth motion sequence (concatenation of prior poses and ground truth future pose sequence). z is a noise vector drawn from a uniform distribution.

$$L(x, y, z)_{wgan} = D(x||G(x, z) - D(x||y) \tag{3.2}$$

Gradient penalty L_{gp} is utilised rather than the typical weight clipping method applied in WGAN. It was found by [19] that weight clipping pushes the weights to the clipped values while a gradient penalty represents a more Gaussian distribution with improved optimisation and improved training stability. In addition a gradient penalty removes the need for batch normalisation layers.

$$L(x, y, z)_{gp} = (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \tag{3.3}$$

$$\hat{x} = \epsilon(x\|y) + (1 - \epsilon)(x\|G(x, z)) \tag{3.4}$$

ϵ is drawn for a uniform distribution between 0 and 1

The regulariser L_2 is the L_2 norm of the critics weights.

$$L_2 = \|\theta_d\|_2 \tag{3.5}$$

λ and α are hyper parameters set as described in table 3.1.

3.3.2 Generator

The authors add two geometric distance regulators to the generator loss function. L_{pg} equation 3.8 is pose to pose gradient loss and represents consistency from pose to pose

within the generated sequence. Given that human motion changes gradually during a motion sequence, pose gradient loss regulates this change in the generated sequence to achieve the gradual change.

The individual bone length in a generated sequence should be representative of the bone lengths in the real sequence. Without constraining bone length a skeleton is likely to generate abnormal limb lengths for example an arm that is longer and out of ratio with the other limbs. The bone loss regulator L_b, equation 3.9, seeks to minimise the total bone length differences between the generated sequence of skeletons and the ground truth sequence of skeletons.

The GAN loss 3.6 is the addition of the critic loss and the two regulators, bone loss 3.9 and pose consistency 3.8.

$$L_g = L_{adv} + \alpha Lpg + \beta L_b \tag{3.6}$$

α and β are hyper parameters set as detailed in table 3.1

L_{adv} is the WGAN-GP generated loss 3.7, the negative of the critics value of the concatenated driver pose sequence and the generated sequence.

$$L_{adv} = -D(x \| G(x, z)) \tag{3.7}$$

L_{pg} is the pose to pose consistency loss, equation 3.8, calculated as the gradient loss, L_2 norm of the pose to pose gradient. The pose consistency assumes from frame to frame the body position position does not change by a large degree. To avoid the pose gradient reaching zero it is set to the maximum of the calculated value or a hyper

parameter C. C is set as per table 3.1 and $p = 2$

$$L(x, z)_{pg} = \|\nabla_t y\|_p = [\sum_t |y_t - y_{t-1}|^p]^{\frac{1}{p}} \tag{3.8}$$

L_b, equation 3.9 is the bone loss, L_2 norm of bone length between the predicted sequence and the ground truth over the full generated sequence. b_i^t is the predicted bone length at time t and b_{gt}^i is the corresponding ground truth. On examination of the authors code the bone loss is calculation between the poses in the generated sequence rather than on the ground truth. Testing both these options provides similar results in the generated sequences.

$$L_b = \sum_t [\sum_i |b_t^i - b_{gt}^i|^2]^{\frac{1}{2}}. \tag{3.9}$$

3.3.3 Discriminator

The discriminator outputs a single probability between 0 and 1, 0 being fake and 1 real. The purpose of the discriminator is determine the quality of the generated sequences and utilises the standard GAN loss function in equation 3.11 from [17] with the addition of a regulariser.

$$L_d = L_{gan} + \alpha L_2 \tag{3.10}$$

$$L_{gan} = log(D(x\|y)) + log(1 - D(x\|G(x, z))) \tag{3.11}$$

The $L2$ regulariser is the same as in equation 4.6.

3.4 Implementation

3.4.1 Training

The NTU RGB+D Dataset is pre-processed before training all the joints are normalised between [-1, 1], removing the mean and dividing by the standard deviation. Each pose has an in memory skeleton representation constructed, this representation is a definition of the individual bones with their joint information for each end of the bone. The in memory skeleton is utilised when calculating the bone loss regulariser.

HP-GAN training algorithm 2 is similar to the GAN training algorithm 1 with the addition of training a third network, the discriminator. Within each training step a noise vector z is sampled from a uniform distribution, a prior sequence of poses s_{prior} (driver sequence) are sampled from data and the corresponding future sequence f_{future} of poses are sampled from data. A sequence of predicted poses $f_{predicted}$ is sampled from the generator G.

The critic is trained for k steps, 10 in this case, with a forward pass, loss calculation and back propagation for each k step. The generator and discriminator are trained for a single step. The optimiser for all three networks is ADAM with the learning rates set as detailed in 3.1.

The HP-GAN was trained for a total of 250 and results shown in section 3.5.

The critic trains on both complete 30 pose sequence (prior 10 poses and 20 future pose sequence) and the 10 prior pose sequence concatenated with the 20 generated sequences from the generator. The output is a single value used to calculate the WGAN-GP loss.

Parameter	Setting
λ	10
β	0.01
α	0.001
C	0.0001
Critic Learning Rate	5×10^{-5}
Generator Learning Rate	5×10^{-5}
Discriminator Learning Rate	25×10^{-5}

Table 3.1: Hyper parameters and settings

The generator accepts the 10 prior pose sequence and a random noise vector as input. The generator outputs the predicted sequence of 20 poses. Figure 3.2 details the sequence to sequence model, note that the noise vector z is applied to the encoders output states and these then serve as the initial state of the decoder. The last output sequence of the encoder is passed as the first input of the decoder.

The discriminator trains on both complete ground truth 30 pose sequence (prior 10 poses and 20 future pose sequence) and the predicted (10 prior pose sequence concatenated with the 20 generated sequences) from the generator. The output is a probability the sequence is real.

Algorithm 2 HP-GAN training. k is a hyper parameter which governs the number critic training step per epoch. j is the number of generator training steps per epoch. m = 128, n = 10, p = 20

for number of training steps **do**
  Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from $U[-0.1, 0.1]$.
  Sample prior sequence minibatch $\{s_{prior}(x)^{(1)}, ..., s_{prior}(x)^{(n)}\}$ from $p_{data}(x)$.
  Sample future sequence minibatch $\{f_{future}(x)^{(11)}, ..., f_{future}(x)^{(p)}\}$ from $p_{data}(x)$.
  Sample predicted sequence $\{f_{predicted}(y)^{(1)}, ..., f_{predicted}(y)^{(p)}\}$ from G.
  for k steps **do**
    Update the critic C by:

$$L_{wgan} + \lambda L_{gp} + \alpha L_2$$

  end for
  for j steps **do**
    Update the generator G:

$$L_{adv} + \alpha Lpg + \beta L_b$$

  end for
  Update the discriminator D:

$$L_d = L_{gan} + \alpha L_2$$

end for

3.5 Results

To assess the quality of predicted sequences a number of empirical assessments are conducted along with visual analysis. Firstly the training and validation losses are presented and discussed. Then a series of results are included for visual assessment and an euclidean distance measurement and maximum mean discrepancy between the ground truth and the generated sequence are included. Additionally as per the paper a discriminator is trained as a classifier that returns a probability the sequence is real.

3.5.1 NTU RGB+D Dataset

The dataset utilised for this implementation is the NTU RGB+D [43] dataset. The NTU RGB+D data sets consists of sixty action classes performing daily activities (eating and drinking), health related activities (sneezing, staggering) and mutual activities that involve two subjects in the scene. Performing these actions are forty different subjects (actors). Each activity is represented as a sequence of individual poses represented by skeleton data and the activities vary in sequence length. Each pose is a skeleton consisting of twenty five body joints each represented by its position in three dimensions space with x,y and z coordinates. Only the single subject actions are utilised and multi subject sequences are removed from the dataset during preprocessing. In addition there are a total of 302 incomplete sequences, these are also removed during preprocessing.

Figure 3.5 is a single skeleton pose representation from the action sequence "fan self". Each dot represents one of the 25 joints that form the skeleton, each line between dots represents a bone. Green bones on the right side of the body, blue left side and red the centre.

Figure 3.6 is a Representative sample of skeleton sequences that form a human motion activity, in this case "taking a selfie" activity.

During pre-processing the dataset is divided into training, validation and test datasets. The number of sequences in each set is summarised in table 3.2

Figure 3.5: Sample skeleton for activity "fan self"

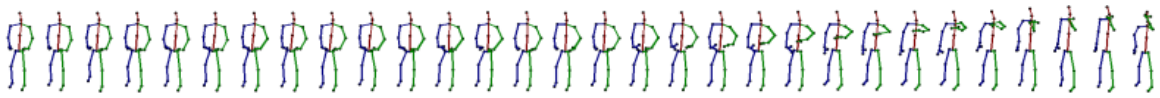


Figure 3.6: Sample of skeleton sequence for activity "taking a selfie"

Data Set	Count
Training	15,296
Validation	1,904
Test	1904

Table 3.2: Activity sequences per dataset

3.5.2 Results

The HP-GAN model is training for 250 epochs on the training and validation sets as per table 3.2. Figure 3.1 is the critic loss that trends to convergence at 250 epochs, training beyond 250 did not show considerable improvement beyond this point. It is difficult to

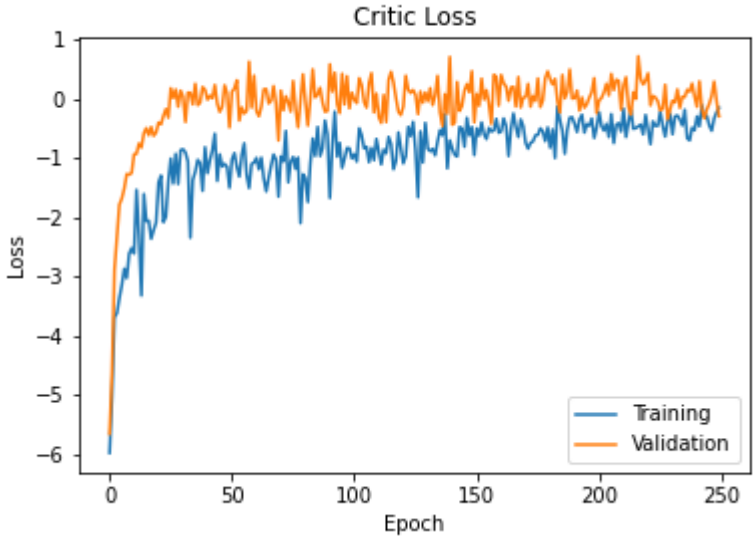


Figure 3.7: Critic training and validation losses.

draw a point for convergence or optimum training point for the generator from Figure 3.8. Given the generator (GAN) loss equation 3.6 consists of three individual losses, critic (3.7), bone loss (3.9) and pose consistency loss (3.8) review of these losses is more helpful. Figure 3.9 is the bone and pose consistency loss, bone loss continually improves across the training period and does not fully converge with additional training but does provide a good signal. Pose consistency loss figure 3.9b provides a good improving training signal, both the bone loss and pose consistency loss provide a good training signal for the generator.

The discriminator loss figure 3.10 provides a stable training signal progressing and convergence while taking at least 150 epochs to train. As will be observed in section 3.5.3 the discriminator proves very strong classification of true and generated sequences.

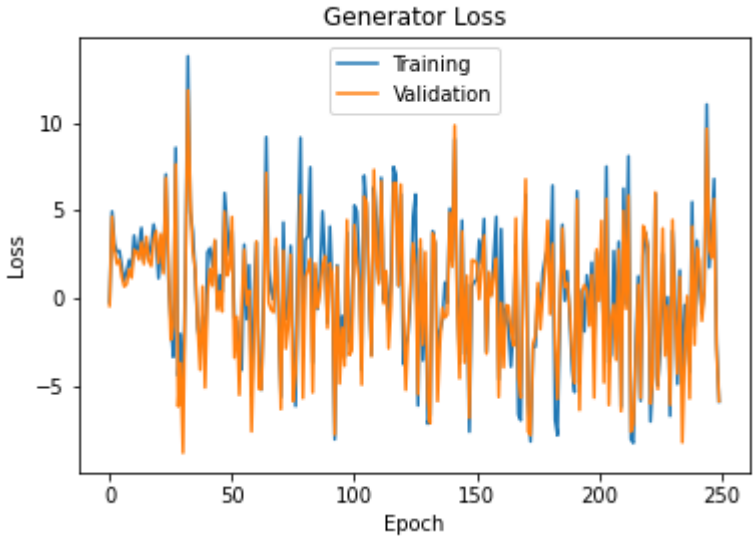
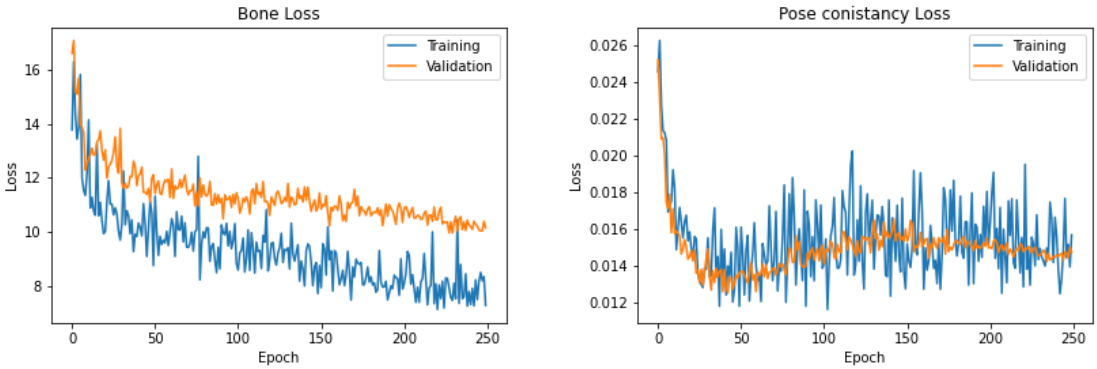


Figure 3.8: GAN training and validation losses.



(a) Bone training and validation loss

(b) Pose consistency loss

Figure 3.9: Bone and pose consistency training/validation losses

3.5.3 Generated Sequences

The following series of results are the result of training the HP-GAN model for 250 epochs according to the method described in section 3.4.1 Training. In figure 3.11a the first line is the ground truth of the activity sequence. Each following line is a concatenation of the 10 pose prior sequence and the subsequent 20 generated sequences.

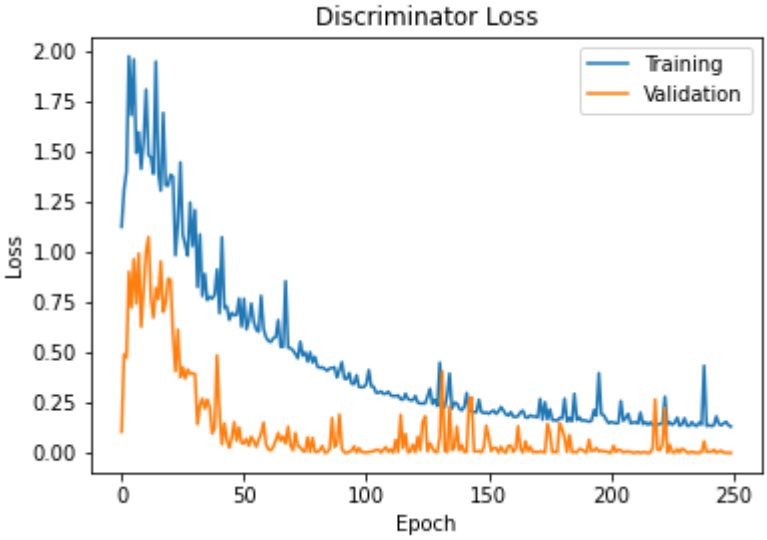
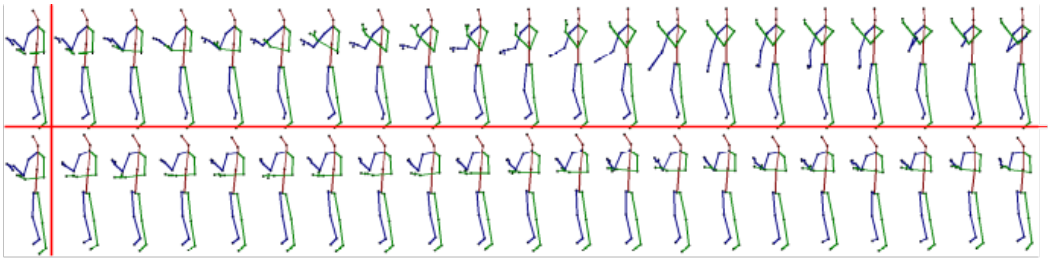


Figure 3.10: Discriminator training and validation losses.

Sequences to the left of the red line are the 10 prior sequences and right of the red line are the generated sequences. Each line below the first line (ground truth) is generated with a different noise vector. Figure 3.11b zooms in on the sequence and depicts the ground truth sequence on the first line and the first generated sequence, the figure to the left of the red horizontal line is the 10^{th} pose of the driver poses. The data lines are the discriminator classification score on the sequence, euclidean distance measures the poses joint by joint distance from the real pose, MMD does the same for maximum mean distance for each pose. These sequences are generated with a test dataset.

Figure 3.11a is the "phone call" activity, as can be observed the first four generated poses are a reasonable representation of the activity. At pose five the representation starts to lose its likeness as can be observed in the arms failing to cross over. This is validated in figure 3.11b where the euclidean distance per pose starts to increase to 0.327 and then increases for each following pose until the final predictions distance is 3.044 indicating a significant departure from the real pose. The increasing MMD

(a) Multiple generations from a single driver sequence



Discriminator 0.0168
Euclidean distance 0.184 0.258 0.282 0.283 0.327 0.513 0.475 0.494 0.505 0.569 0.717 0.750 0.926 1.251 1.562 1.895 2.338 2.485 2.792 3.044
MMD 0.002 0.004 0.005 0.005 0.007 0.011 0.011 0.009 0.011 0.013 0.020 0.020 0.026 0.034 0.046 0.058 0.070 0.071 0.080 0.085

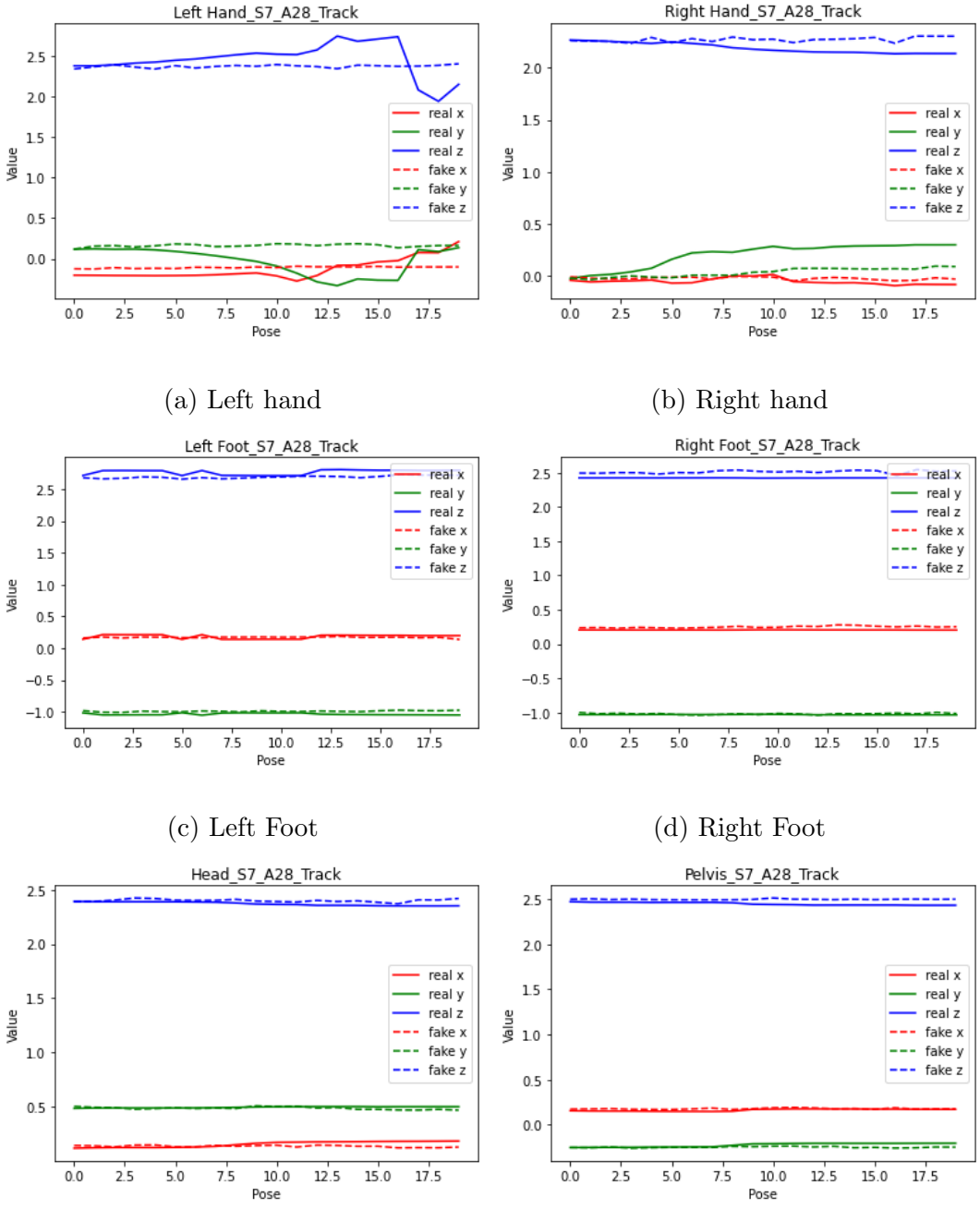
Average euclidean distance - 1.083
Average MMD - 0.029

(b) Generated sequence measurements

Figure 3.11: Activity 28/Subject 7 - phone call, trained for 250 epochs

also supports what is observed. The figures within figure 3.14 plot specific x, y, z positions of a number of joints which typically move during an action, head, pelvis, left/right foot and left/right hands. The real action is represented with solid lines and the generated(fake) sequences with dashed lines. x in red y in green and z in blue. As observed in figures 3.12a and 3.12b the predictions of the hand positions deviate from pose five on wards, the other features are reasonably static through the sequence so are predicted well through to the end. This suggests HP-GAN predicts reliably through

to pose five and degrades from there, examination of further dynamic and static poses will assist in this assessment.



(a) Left hand

(b) Right hand

(c) Left Foot

(d) Right Foot

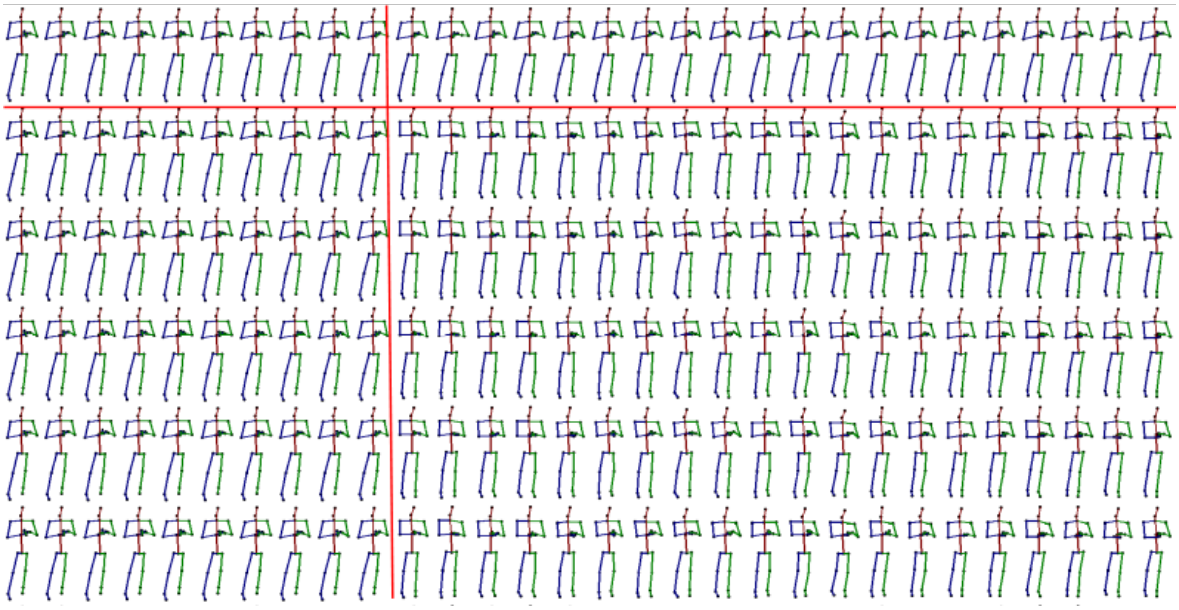
(e) Head

(f) Pelvis

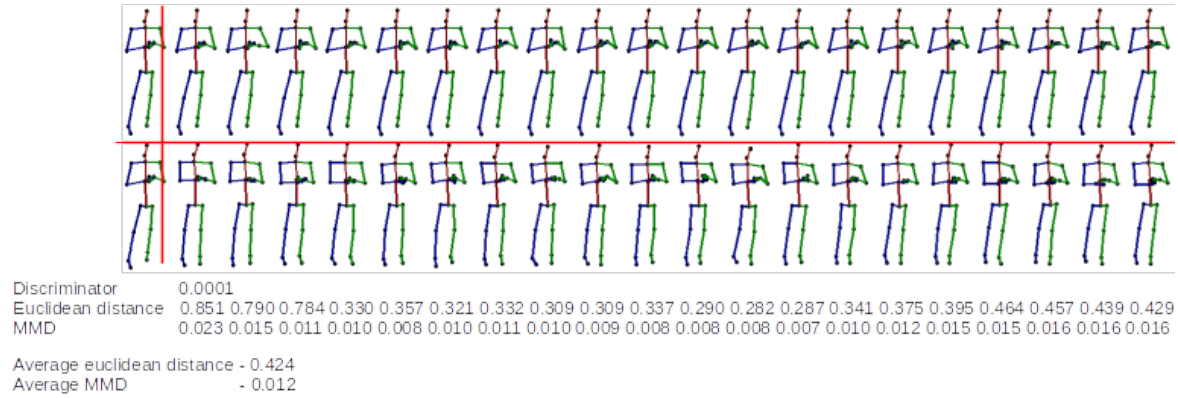
Figure 3.12: Activity 28 phone call, Subject 7 joint tracking

Figure 3.13 is the same activity "phone call" performed by a different subject, in

this sequence the subject is more static with minimal hand movement. Visually the predictions follow the real sequence and the euclidean distance and MMD measures are consistent through the sequence and validated with the joint tracking in figure 3.14. The long term (20 pose) prediction is more reliable with a more static activity, given the generator component of the GAN only ever processes the 10 (driver) poses it is challenging for the model to learn any unseen poses.

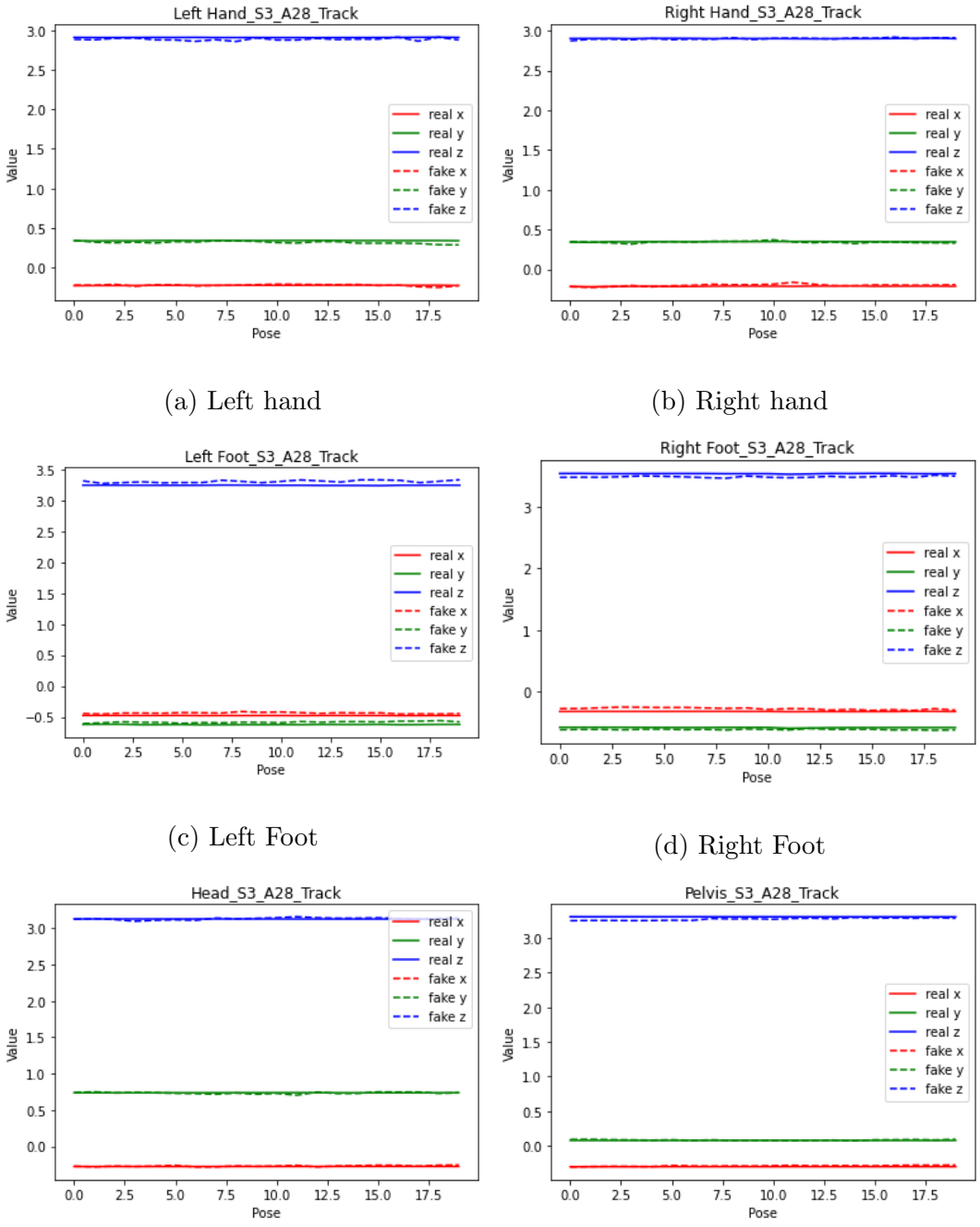


(a) Multiple generations from a single driver sequence



(b) Generated sequence measurements

Figure 3.13: Activity 28/Subject 3 - phone call, trained for 250 epochs



(a) Left hand

(b) Right hand

(c) Left Foot

(d) Right Foot

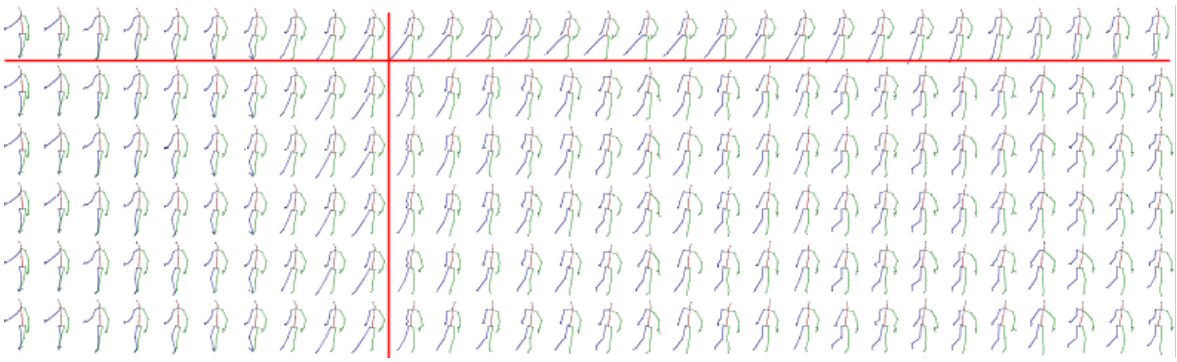
(e) Head

(f) Pelvis

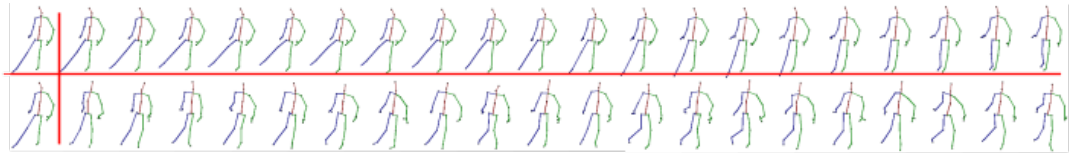
Figure 3.14: Activity 28 phone call, Subject 3 joint tracking

The "kick something" activity in figure 3.15 is a more dynamic sequence with more leg movement, the pose and euclidean distance quickly degrade after the first generated pose. Poses 3 and 4 provide little resemblance to the real pose, by the end

of the sequence both the euclidean distance and MMD improve from the middle of the sequence, however this is due more to the real poses returning to a more neutral stance than the model predicting the pose. Note as the left leg remains in a "kick" position, where as it should return to a neutral stance. The figure 3.16c left foot tracking indicates this.



(a) Multiple generations from a single driver sequence



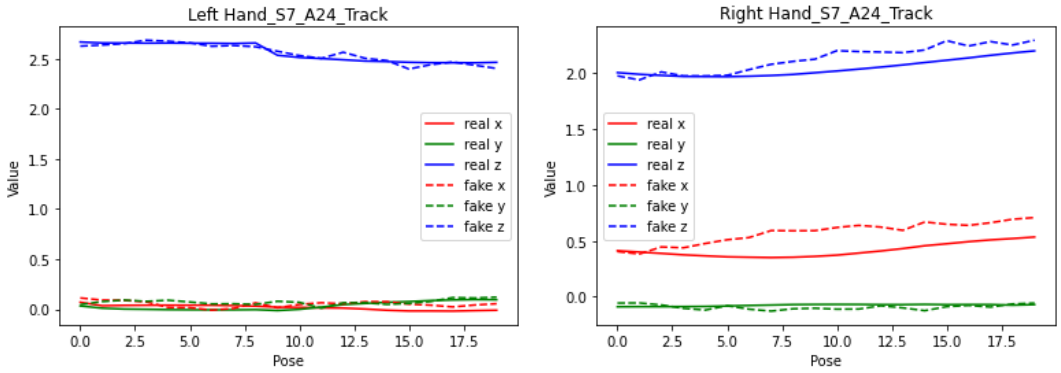
Discriminator 0.5504
Euclidean distance 0.288 0.498 0.651 0.880 0.964 0.966 1.000 1.254 1.387 1.207 1.236 1.063 0.801 0.660 0.677 0.663 0.593 0.719 0.781 0.607
MMD 0.002 0.004 0.006 0.008 0.011 0.009 0.012 0.017 0.018 0.024 0.027 0.023 0.021 0.016 0.018 0.015 0.010 0.011 0.013 0.009

Average euclidean distance 0.845
Average MMD 0.014

(b) Generated sequence measurements

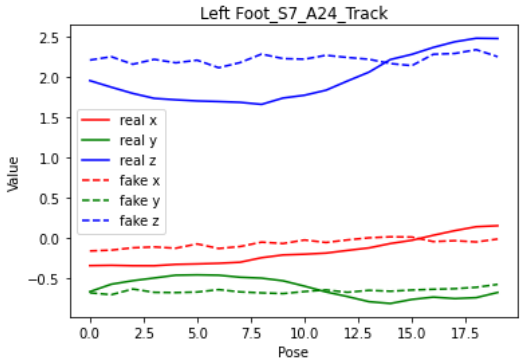
Figure 3.15: Activity 24/Subject 7 - kicking something, trained for 250 epochs

Joint occlusion where limbs obscure each other effect the reliability of the generated poses. Figure 3.17 is a sequence of "take off jacket" action, the arms in the real sequence 3.17b occludes each other for the duration of the sequence and poses 2,3,4,5 and 6 the body is side on and the right hip is occluded by the left hip giving the appearance of no hips. This activity presents a significant challenge to the model, both arms fail to learn correct positioning, positioned to the wrong side of the body for the whole sequence.

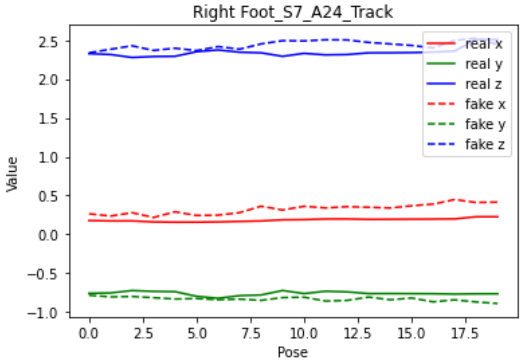


(a) Left hand

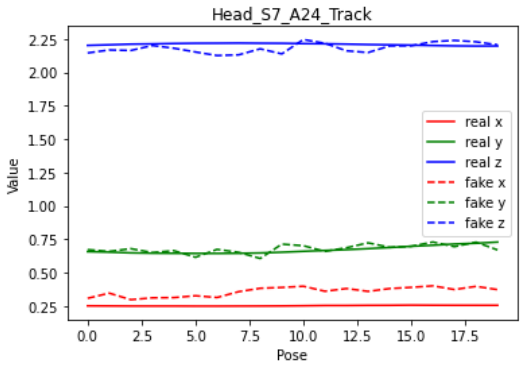
(b) Right hand



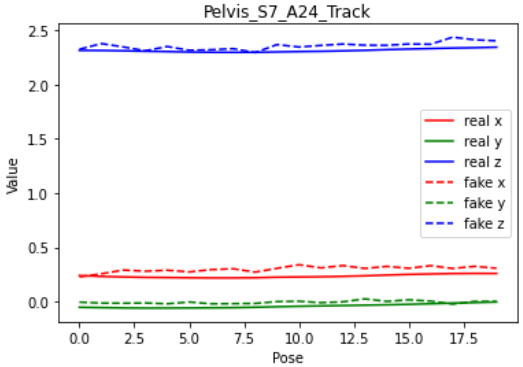
(c) Left Foot



(d) Right Foot



(e) Head

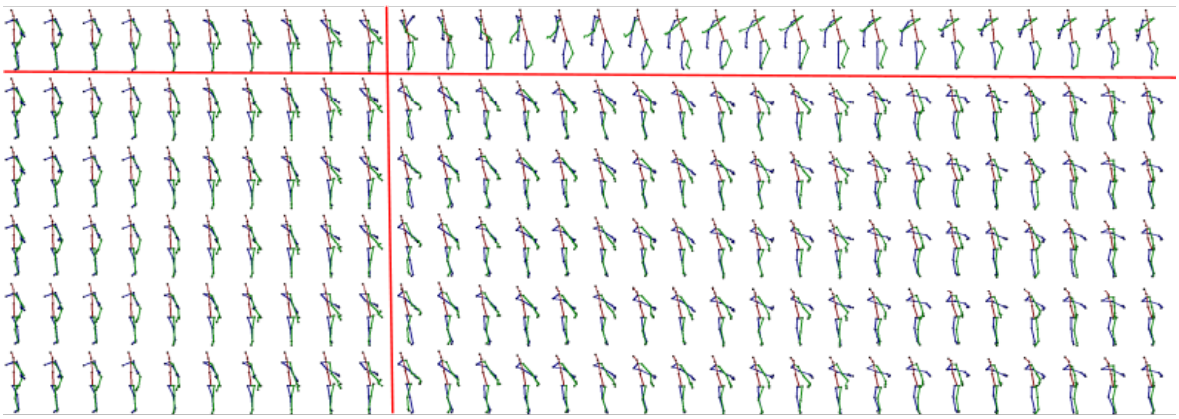


(f) Pelvis

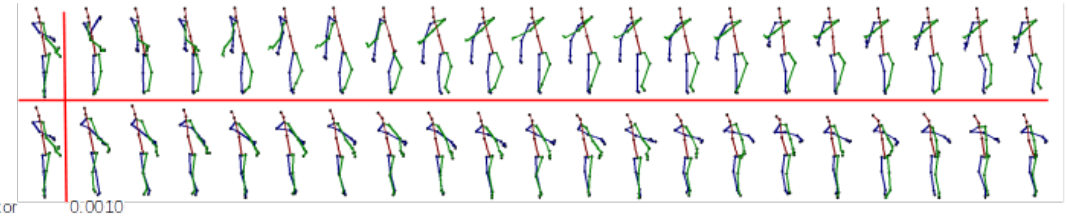
Figure 3.16: Activity 24 "kick something", Subject 7 joint tracking

Both the euclidean distance and MMD measures show a major departure from the real sequence from early in the prediction. In addition the model fails to learn the rotation of the hips in poses 3 to 6. Joint tracking for both hands (figures 3.18a and 3.18b)

clearly indicates the models difficulty to learn arm position in their occluded state.



(a) Multiple generations from a single driver sequence



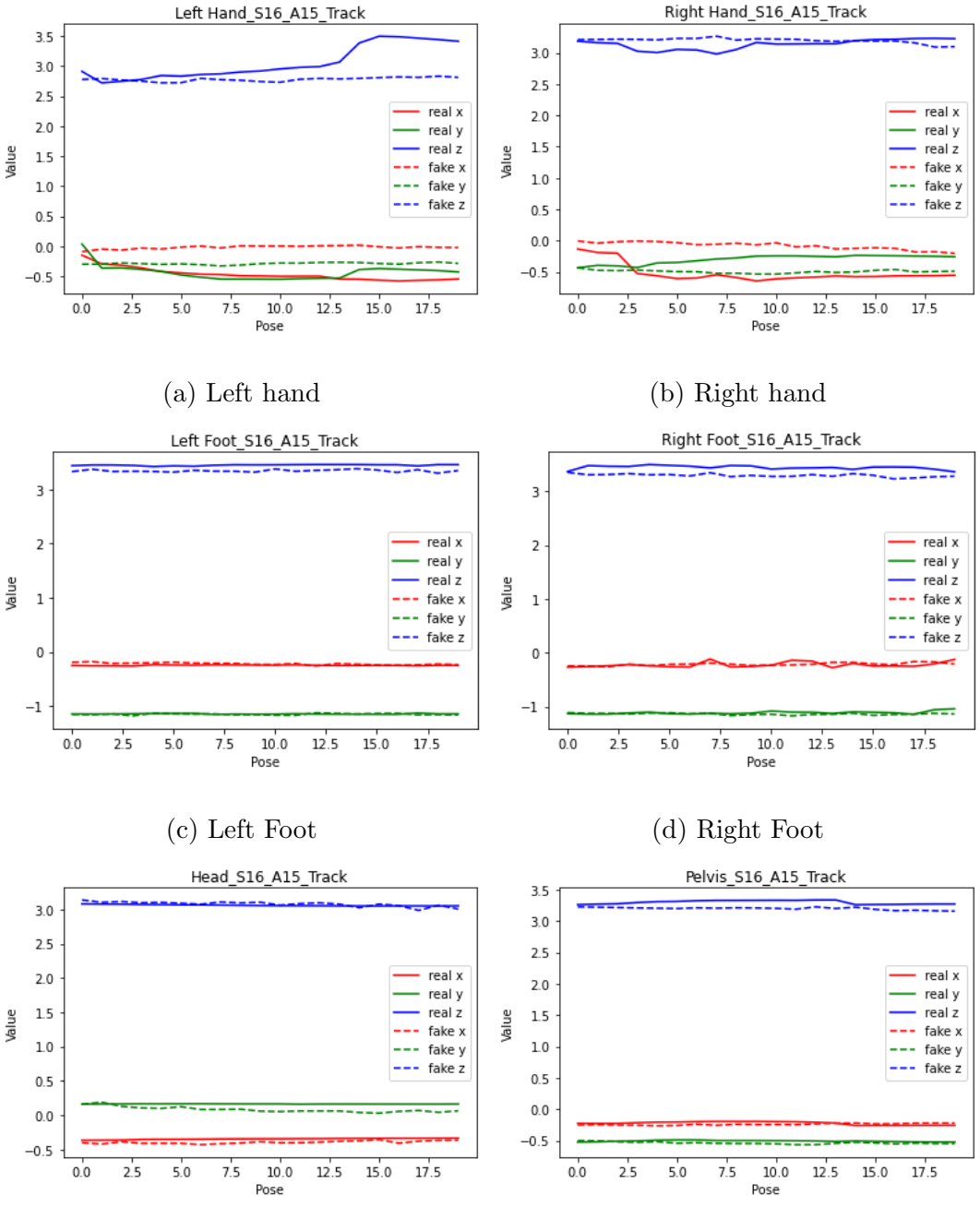
Discriminator 0.0010
Euclidean distance 0.721 0.584 0.700 1.929 2.393 2.737 2.954 2.832 3.243 3.587 3.761 3.416 3.160 3.394 4.189 4.489 4.228 4.359 4.173 3.750
MMD 0.005 0.008 0.011 0.032 0.036 0.044 0.050 0.036 0.046 0.055 0.058 0.054 0.054 0.063 0.085 0.091 0.093 0.100 0.099 0.092

Average euclidean distance 3.030
Average MMD 0.056

(b) Generated sequence measurements

Figure 3.17: Activity 15/Subject 16 - take off jacket, trained for 250 epochs

Tables 3.3 and 3.4 are measurement metrics by activity for the NTU dataset.



(a) Left hand

(b) Right hand

(c) Left Foot

(d) Right Foot

(e) Head

(f) Pelvis

Figure 3.18: Activity 15 "take off jacket", Subject 16 joint tracking

Table 3.3: Metrics for actions 1 thru 24

Activity	drink water	eat meal	brush teeth	brush hair	drop	pick up	throw	sit down	stand up	clapping	reading	writing	tear up paper	put on jacket	take off jacket	put on a shoe	take off a shoe	put on glasses	take off glasses	put on hat	take off hat	cheer up	hand waving	kicking something
Euclidean Dist																								
Average	0.356	0.265	0.204	0.330	0.268	1.476	1.446	1.115	0.664	0.246	0.253	0.192	0.413	1.009	0.848	0.683	0.635	0.217	0.360	0.572	0.544,	0.513	0.449	1.271
Max average	3.415	0.765	1.215	0.966	1.133	8.937	2.548	3.835	1.581	1.318	1.040	0.745	1.218	4.502	3.031	2.104	1.891	0.948	1.283	2.276	1.955	1.142	1.483	2.537
Min average	0.033	0.066	0.056	0.116	0.053	0.378	0.295	0.251	0.238	0.086	0.073	0.048	0.059	0.197	0.076	0.142	0.188	0.057	0.051	0.085	0.093	0.121	0.065	0.435
MMD																								
Average	0.007	0.004	0.003	0.006	0.005	0.038	0.031	0.026	0.016	0.004	0.004	0.003	0.413	0.018	0.017,	0.013	0.011	0.004	0.007	0.010	0.011,	0.010	0.008	0.018
Max Average	0.119	0.014	0.014	0.034	0.018	0.297	0.071	0.079	0.042	0.037	0.025	0.018	0.013	0.131	0.113	0.070	0.039	0.022	0.032	0.052	0.063	0.026	0.042	0.046
Min Average	0.000	0.000	0.001	0.001	0.000	0.004	0.004	0.004	0.003	0.001	0.001	0.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.005
Discriminator																								
Average	0.025	0.003	0.003	0.001	0.003	0.000	0.022	0.007	0.001	0.002	0.003	0.014	0.002	0.004	0.015	0.001	0.000	0.002	0.001	0.004	0.004	0.009	0.001	0.016

Table 3.4: Metrics for actions 25 thru 49

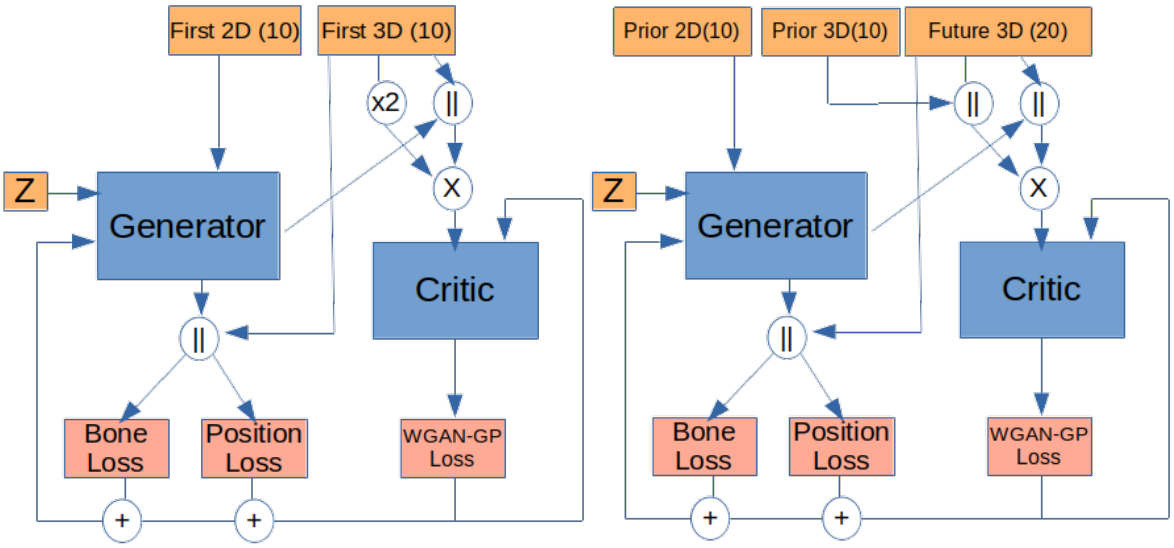
Activity	each into pocket	hopping	jump up	phone call	play with phone/ tablet	type on a keyboard	point to something	taking a selfie	check time	rub two hands	nod head/bow	shake head	wipe face	salute	put palms together	cross hands in front	sneeze/ cough	staggering	falling down	headache	chest pain	back pain	neck pain	nausea vomiting	fan self
Euclidean Dist																									
Average	0.223	0.528	1.224	0.320	0.176	0.171	0.697	0.361	0.329	0.188	1.251	0.289	0.236	0.358	0.286	0.978	0.433	1.359	2.853	0.234	0.272	0.405	0.304	0.960	0.434
Max average	0.841	1.553	5.086	1.869	1.072	0.572	3.199	2.224	1.042	1.336	4.329	2.126	2.090	1.367	1.126	2.222	1.951	4.552	8.474	0.697	1.057	2.351	1.668	3.292	1.196
Min average	0.052	0.171	0.253	0.046	0.035	0.057	0.092	0.065	0.067	0.039	0.166	0.050	0.053	0.086	0.055	0.098	0.074	0.413	0.477	0.037	0.045	0.059	0.052	0.269	0.075
MMD																									
Average	0.004	0.014	0.034	0.004	0.004	0.003	0.013	0.005	0.005	0.003	0.028	0.008	0.004	0.005	0.006	0.023	0.010	0.045	0.090	0.004	0.007	0.008	0.005	0.026	0.007
Max Average	0.031	0.058	0.133	0.017	0.031	0.021	0.064	0.021	0.019	0.025	0.126	0.077	0.070	0.014	0.029	0.052	0.074	0.112	0.333	0.013	0.034	0.049	0.040	0.130	0.028
Min Average	0.001	0.002	0.003	0.001	0.000	0.000	0.001	0.001	0.001	0.000	0.002	0.000	0.000	0.001	0.000	0.001	0.001	0.009	0.014	0.000	0.001	0.001	0.000	0.001	0.001
Discriminator																									
Average	0.002	0.001	0.003	0.008	0.002	0.002	0.001	0.025	0.036	0.002	0.001	0.002	0.001	0.004	0.011	0.004	0.021	0.002	0.000	0.001	0.002	0.018	0.002	0.002	0.001

Chapter 4

2D to 3D Uplift and Prediction

This section presents the research of this thesis which is a generative method for 2D to 3D dimension human pose uplift, 2D to 3D uplift and future pose prediction. As discussed earlier in 2.2.2 and 2.1.3, human pose 2D to 3D uplift and human pose prediction has numerous applications from autonomous vehicles to computer games and animation applications.

This section describes the components of this method, a generative adversarial network (GAN) consisting of a sequence to sequence encoder/decoder generator, multi level perception critic and loss functions. This work draws from the experiences of HP-GAN [4], WGAN-GP [19] and [37]. Figure 4.1 details the high level architecture of the proposed GAN implementation. This model inputs a 2D skeleton action sequence and lifts sequence into a 3D skeleton representation. By modifying the input and output dimensions of the generator and input to the loss functions, the model is able to predict and lift a 3D series of future poses from the input sequence. This method is a generative unsupervised model with a matching 2D and 3D ground truth dataset.



(a) GAN High level architecture Mode 1

(b) GAN High level architecture Mode 2

Figure 4.1: GAN High level architectures

4.1 Network Architecture

4.1.1 Generator

The generator block in figure 4.1 is a sequence to sequence encoder decoder network which operates in one of two modes. In the first mode, "Mode 1", figure 4.1a, the generator inputs a 10 pose sequence from 2D action sequence and outputs a lifted to 3D sequence of the input.

In the second mode, "Mode 2", figure 4.1b, the generator takes as input the first 10 poses of a 30 pose 2D action sequence and outputs a 3D lifted future prediction of the next 20 poses in the sequence. Let $x = \{x_1, x_2, x_3, ..., x_{30}\}$ be the sequence of 2D input poses and $x_{gt} = \{x_{1gt}, x_{2gt}, x_{3gt}, ..., x_{30gt}\}$ be the corresponding 3D ground truth. $y = \{y_1, y_2, ..., y_{10/20}\}$ represents the predicted uplifted 3D poses.

In addition, the generator G takes as input a 128 dimension noise vector z drawn

from a uniform distribution $[-0.1, 0.1]$. As detailed in figure 4.2, this vector is projected to 250 dimensions in a fully connected layer to match the dimensions required to fill the third dimension of the input sequence. This creates a 3D sequence with the z dimension containing noise.

The generator G has an encoder/encoder structure as depicted in figure 4.2. The first layer concatenates the noise vector as the third dimension to the pose sequence. The second and third layers are two RNN (GRU) cells of 1024 states. The states and output from the encoder are the initial states and first input to the decoder. The decoder mirrors the encoder with two RNN (GRU) layers of 1024 states, the final layer is a fully connected layer to project the decoder output to the required dimensions of the output sequence, in the case of Mode 1 750 and Mode 2 1500. To calculate

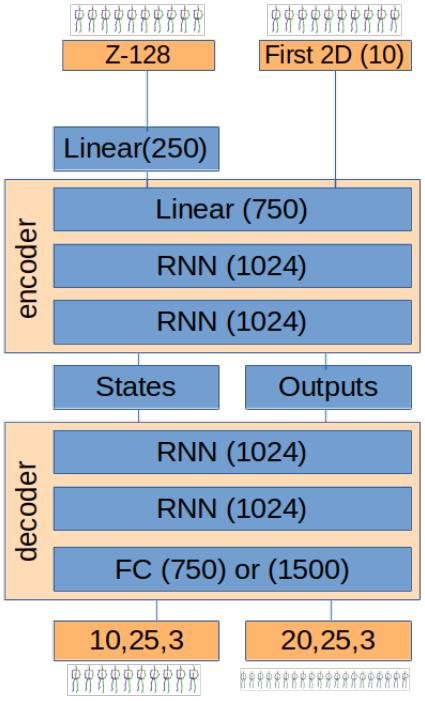


Figure 4.2: Generator architecture

the generator losses in mode 1, the generated 3D sequence is concatenated with the

matching 3D ground truth of the 10 pose prior ($\{x_{gt_1}...x_{gt_{10}}\}||\{y_1,...,y_{10}\}$). In mode 2 operation the generated 3D sequence is concatenated to the 20 future 3D poses of the sampled sequence, ($\{x_{gt_{11}},...x_{gt_{30}}\}||\{y_1,...,y_{10}\}$)

Generator loss in figure 4.1 is sum of critic loss L_c, bone loss L_b and position loss L_p. Bone loss focuses on reducing the change in bone lengths between predicted skeletons and the ground truth. Position loss focuses on reducing the relative total position of joints from the ground truth. Loss functions are discussed further described in section 4.1.3

4.1.2 Critic

The critic C network (figure 4.3) is a three layer fully connected network with an output fully connected layer that outputs a single value without an activation, output is an unbounded value used in WGAN-GP loss [19] calculation. Input to C varies between mode 1 (2D-3D uplift)and mode 2 (2D to 3D uplift with prediction). Input for mode 1 is the concatenation of the generated 3D uplift sequence and corresponding 3D ground truth ($\{x_{1_{gt}},...,x_{10_{gt}}\}||\{y_1,...,y_{10}\}$). Utilisation of a multi pose sequence in the critic allows the style over that sequence to be learned rather than just a single pose.

Input for mode 2 is the concatenation of 3D ground truth of the 2D prior sequence and the generated 3D uplifted twenty pose prediction ($\{x_{1_{gt}},...,x_{10_{gt}}\}||\{y_1,...,y_{20}\}$) (total of 30 poses). The critic is trained on alternating sequence of generated poses and ground truth poses as indicated by the $\bigotimes$ symbol in figure 4.1.

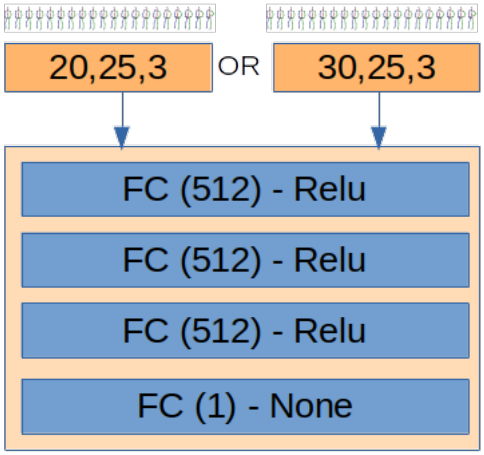


Figure 4.3: Critic architecture

4.1.3 Loss Functions

Critic loss

The critic loss (4.1) is WGAN-GP [19] loss with the addition of L_2 regularisation of the networks weights.

$$L_c = L_{wgan} + \lambda L_{gp} + \gamma L_2 \tag{4.1}$$

λ and γ are hyper parameters set as detailed in table 4.1

L_{wgan} (4.2, 4.3) is WGAN [3] loss with the addition of a gradient penalty (4.4) from [19] rather than WGAN's weight clipping method. Mode 1 loss 4.2 and equation 4.3 loss for mode 2 operation. The difference being, mode 1 $G(x, z)$ is the 3D lifted sequence (10 poses) of the 2D prior sequence ($x_1 ... x_{10}$), mode 2 $G(x, z)$ is the 3D predicted future sequence (20 poses) of the 2D prior sequence.

$$L(x, x_{gt}, z)_{wgan} = C(G(x, z)) - C(x_1 ... x_{10gt}) \tag{4.2}$$

$$L(x, x_{gt}, z)_{wgan} = C(x_1 ... x_{10gt}||G(x, z) - C(x_1 ... x_{30gt}) \tag{4.3}$$

$$L(x, x_{gt}, y, z)_{gp} = (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \tag{4.4}$$

$$\hat{x} = \epsilon(x_{gt} \| y) + (1 - \epsilon)(x_{gt} \| G(x, z)) \tag{4.5}$$

ϵ is drawn for a uniform distribution between 0 and 1

The regulariser L_2 is the l_2 norm of the critics weights.

$$L_2 = \|\theta_d\|_2 \tag{4.6}$$

Generator loss

Generator loss (4.7) consists of three components the WGAN-GP loss L_{crit}, position loss L_{pos} and bone loss L_{bn}. Position loss measures each joints distance between the generated skeleton and the ground truth. Bone loss measures the difference in length of each bone between the generated sequence and the ground truth.

$$L_g = L_{crit} + \alpha L_{pos} + \beta L_{bn} \tag{4.7}$$

$$L_{crit} = -C(x_1 ... x10gt \| G(x, z)) \tag{4.8}$$

Position loss (4.9)is the L_2 norm of the difference of each skeletons joint positions between the generated (predicted) and ground truth sequences, where t is the time step.

$$L_{pos} = \|x_{gt} - y\|_2 = \left[\sum_t |x_{gt_t} - y_t|^2 \right]^{\frac{1}{2}} \tag{4.9}$$

Bone loss is the l_2 norm of each bones difference in length between the generated (predicted) and ground truth sequences, where t is time step and i is bone index.

$$L_{bn} = \sum_t \left[\sum_i |b_t^i - b_{gt}^i|^2 \right]^{\frac{1}{2}} \tag{4.10}$$

α and β are hyper parameters set as detailed in table 4.1

Generator loss is the same for both mode 1 and mode 2, the input differs between the modes.

Parameter	Setting
λ	10
β	0.01
α	0.001
γ	0.00005
Critic Learning Rate	5×10^{-5}
Generator Learning Rate	5×10^{-5}

Table 4.1: Hyper parameters and settings

4.1.4 Implementation

Data Pre-processing

The dataset contains human activities represented as a sequences of poses, with each individual pose represented by a skeleton containing 25 (NTU) or 32 (Human3.6M) joint positions of three dimensions (z,y,z). To produce a two dimensional representation the pelvis/root joint value is subtracted from each joint and each joint is normalised into the range [-1,1] using a global minimum and maximum. The z dimension is removed leaving a two dimensional (x,y) representation. As a result of the pelvis joint subtraction, the pelvis position is 0. To avoid learning problems on the pelvis joint, it is set to 0.0001.

Training

Training follows the GAN [17] training method. For each training step (batch) the critic is trained for ten steps and the generator for two steps. Both networks use ADAM optimiser with learning rates as in table 4.1, decay rates were experimented with no improvement. Batch size is 16. All training is for 125 epochs.

Algorithm 3 Training. k is a hyper parameter which governs the number critic training step per batch. j is the number of generator training steps per epoch. m = 128, n = 10, p = 20, k = 10, j = 2, batch = 16

for number of training steps **do**
  Sample batch of m noise samples $\{z_1, ..., z_m\}$ from $U[-0.1, 0.1]$.
  if Mode 1 **then**
    Sample 2D driver sequence batch $\{x_1, ..., x_n\}$ from $data(x)$.
    Sample 3D truth sequence batch $\{x_{gt_1}, ..., x_{gt_n}\}$ from $data(x_{gt})$.
    Sample uplifted sequence $\{y_1, ..., y_n\}$ from G.
  end if
  if Mode 2 **then**
    Sample 2D driver sequence batch $\{x_1, ..., x_n\}$ from $data(x)$.
    Sample 3D truth future sequence batch $\{x_{gt_1}, ..., x_{gt_p}\}$ from $data(x_{gt})$.
    Sample uplifted sequence $\{y_1, ..., y_p\}$ from G.
  end if
  for k steps **do**
    Update the critic C by:

$$L_{wgan} + \lambda L_{gp} + \gamma L_2$$

  end for
  for j steps **do**
    Update the generator G:

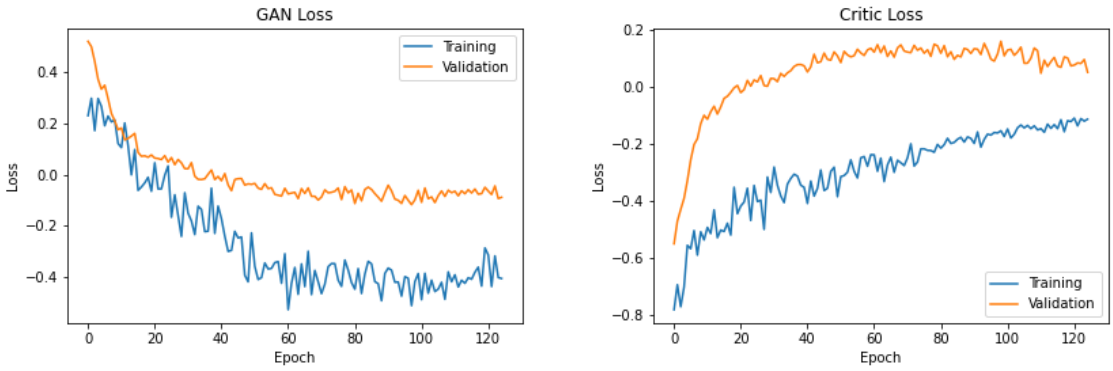
$$L_{adv} + \alpha L_{pg} + \beta L_b$$

  end for
end for

Training Losses

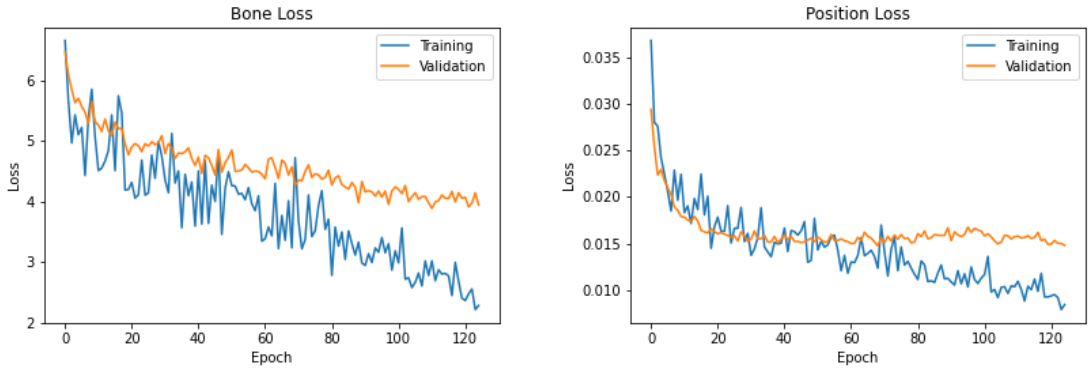
Figures 4.4 and 4.5 are the 2D to 3D uplift training losses (mode 1). The blue line is training losses and yellow validation losses. The critic (4.4b) learns through to 100 epochs and slows. Both the position loss (4.5b) and bone loss (4.5a) begin to over-fit

at about 80 epoch on-wards and provide a better training signal than the GAN loss (4.4a).



(a) GAN loss (b) Critic loss

Figure 4.4: GAN and Critic training/validation losses, 2D to 3D uplift

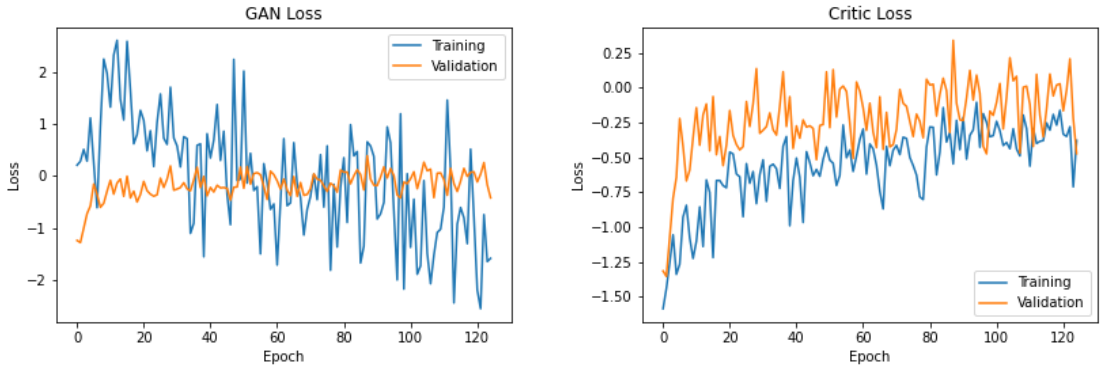


(a) Bone loss (b) Position loss

Figure 4.5: Bone and position training/validation losses, 2D to 3D uplift (mode 1)

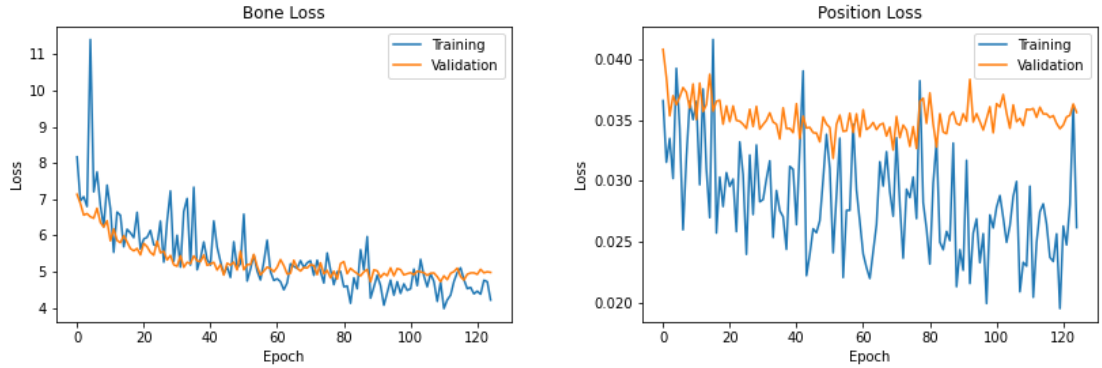
Figures 4.6 and 4.7 are the 2D to 3D uplift with prediction training losses (mode 2) and the blue line is training losses and yellow validation losses. The critic (4.6b) demonstrates a reasonable learning signal and may continue further with additional training epochs. The GAN (4.6a) signal does not provide any clear indication, however the bone loss (4.7a) provides good indication that bone lengths are maintained

consistent. The position loss (4.7a) is over fitting and indicates another regulator may provide a better loss function.



(a) GAN loss - Prediction (Mode 2) (b) Critic loss - Prediction (Mode 2)

Figure 4.6: GAN and Critic training losses, 2D to 3D uplift and Prediction (mode 2)



(a) Bone loss - Prediction (Mode 2) (b) Position loss - Prediction (Mode 2)

Figure 4.7: Bone and position training/validation losses, 2D to 3D uplift and Prediction

4.1.5 Experiments and Results

This section presents the results to quantify the capability of this method. Visual results are presented after training separately on both the NTU RGB+D [43] and

Human3.6M datasets. Sequences with two subjects are removed and sequences with incomplete skeletons are also removed during pre-processing.

Evaluation and comparison against current state of the art models is conducted on the Human3.6M [23] dataset. This dataset contains 15 daily activities such as eating, smoking, walking and talking on the phone. These actions are performed by 11 different actors/subjects at an indoor setting for a total of 3.6 million images captured. The model is trained on subjects S1, S5, S6, S7 and S7 and tested on subjects S9 and S11 and is consistent with common practice [29]. The activity sequences in Human3.6M are long, in many cases greater than $2,000$ poses. In addition, the subjects movement varies slowly from pose to pose and are sequentially sampled.

Testing protocols follows a common method in literature: Protocol 1 is MPJPE which measures the mean euclidean distance between the generated skeleton and the ground truth. Protocol 2 is the MPJPE after aligning the predicted pose with the ground truth with a rigid transformation, referred to as P-MPJPE. Training subjects are 1, 5, 6, 7, 8 and testing subjects 9, 11.

To evaluate the models ability to manage noisy joint data collection, for instance in 2D skeleton data collected from in the "wild" video, a second experiment adding 5% Gaussian noise to the test data compares uplifts against unaltered ground truth.

The model is implemented in Keras on a GeForce RTX 3080Ti.

	NTU RGB+D	Human3.6M
Minimal	Drink Water	Directions
Dynamic	Cheer Up	Greeting
Occlusion	Take off jacket	Walking

Table 4.2: Visual analysis actions

2D to 3D Uplift

The following are results of mode 1, 2D to 3D uplift for visual analysis. The sequences are selected to demonstrate the models capability on actions will minimal body movement, dynamic actions with greater limb movement and actions where the body position leads to joint occlusion. Table 4.2 lists these action types. In these figures (4.8, 4.9, 4.10, 4.11, 4.12, 4.13), the top row (above the red line) is the 3D ground truth for each pose. To the right of the central red line is the 2D pose prior sequence and to the right are the generated 3D uplift of the 2D prior sequence. Each row below the red line is an uplift with a different random z drawn from a uniform distribution. Figures (4.8, 4.9, 4.10) are results trained and tested on the NTU dataset.

Visual Analysis: Figure 4.8 is a reasonably static pose (drink water) and the legs, pelvis, torso and head remain constant through the sequence with just the arms moving slightly. The orientation of the feet and pelvis are well maintained along with the head and shoulders. The thumb and tip of the right hand varies slightly through the sequence. This leads to the uplifted right hand deviating from the ground truth slightly, with the thumb remaining visible in the middle sequence poses where it should not be

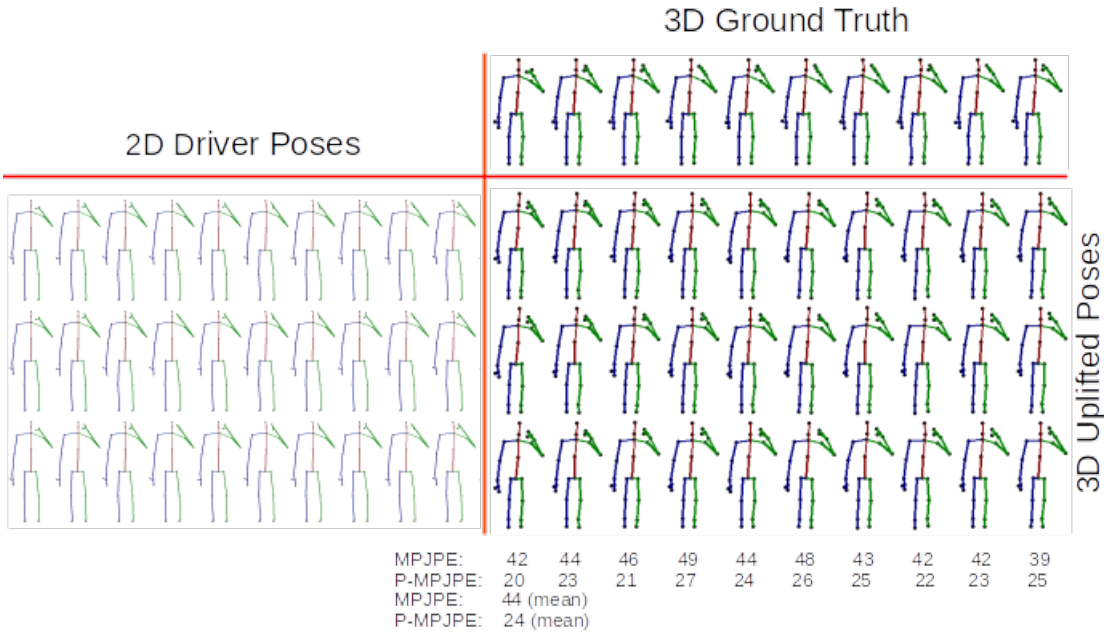


Figure 4.8: Subject 16, Activity 1 Drinking water (NTU)

seen.

More dynamic sequences as in figure 4.9, where the upper body and arms have a much larger range of motion, visually appear to have an effective uplifted representation. The arms are uplifted keeping the elbow and hand positions and orientations aligning with the ground truth. The MPJPE increases from frame to frame with greater body movement, evident in frames 2 to 9. This also suggests the more extended the body (limbs in this case), the greater the error. The MPJPE results indicate a loss of accuracy when compared to the accuracy of a more static pose.

Figure 4.10 demonstrates the models ability on actions with joint occlusion. This action "taking off jacket" is a complicated action with a number of overlapping joint in the legs and arms. In this case, the legs remain static and the uplifted sequence is an accurate representation. The arm and hand joints are occluded, the uplift noticeably loses joint accuracy in poses 7 on-wards, and the MPJPE values increase indicating

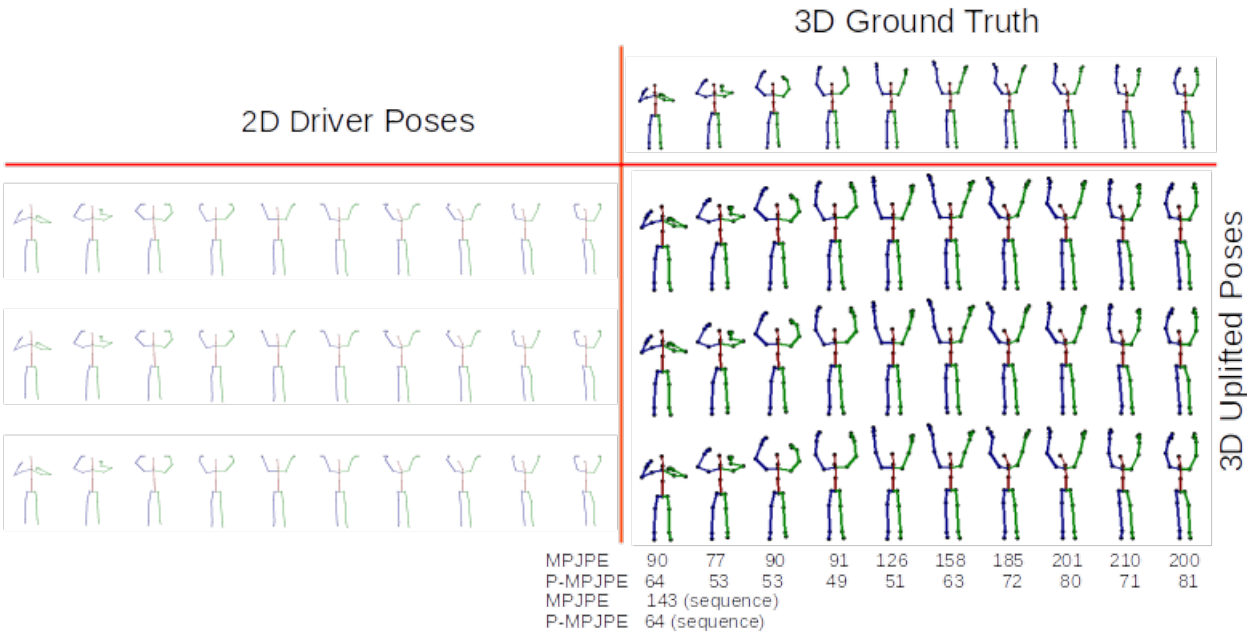


Figure 4.9: Subject 1, Activity 22 Cheer up (NTU)

this inaccuracy. In addition these joints remain occluded in the uplift and fail to learn the separation in the arms.

The Human3.6M data set is a widely used data set for human pose estimation and bench-marking, and provide a points of reference with other approaches. This series of figures (4.11, 4.12, 4.13) are results trained and tested on the Human3.6M (H36M) dataset.

As discussed previously the model is effective uplifting poses when the sequence is static (small limb movements pose to pose), and figure 4.11 further confirms this on the Human3.6M data set. The MPJPE measurements are small, and from an individual pose perspective, exceed state-off-the art in table 4.3.

Similar to uplifts on dynamic sequences from the NTU dataset, uplifts on dynamic sequences from the Human3.6M dataset experience difficulty with large limb movement between poses. This is observed in figure 4.12 where the left arms in the later sequences

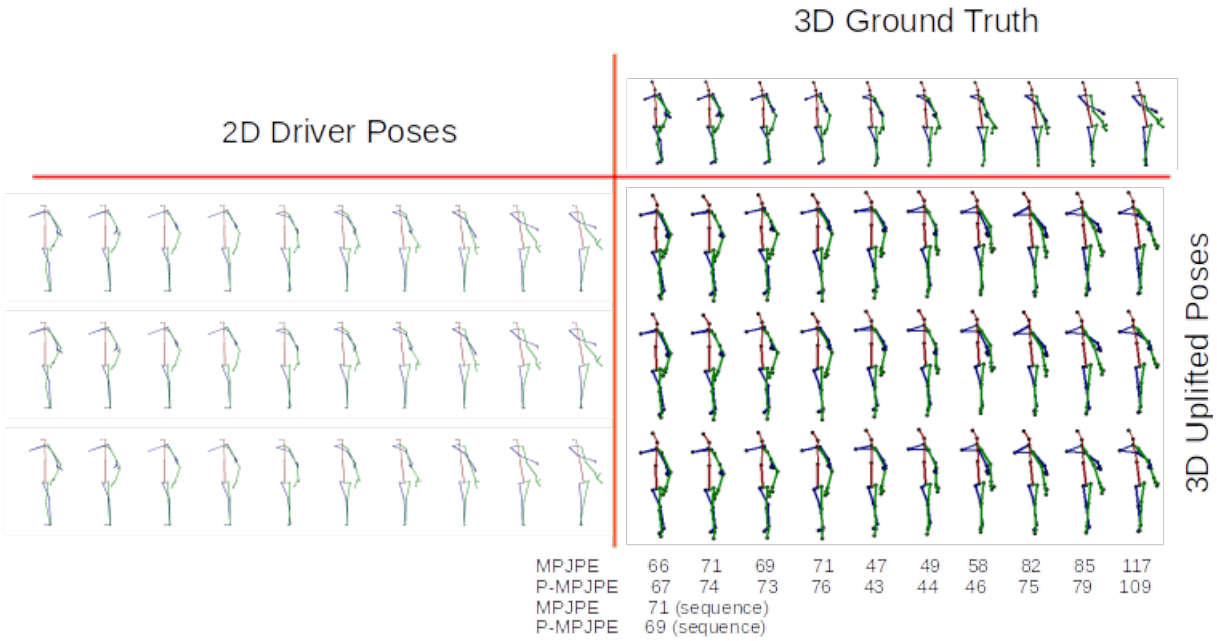


Figure 4.10: Subject 16, Activity 15 Take off jacket (NTU)

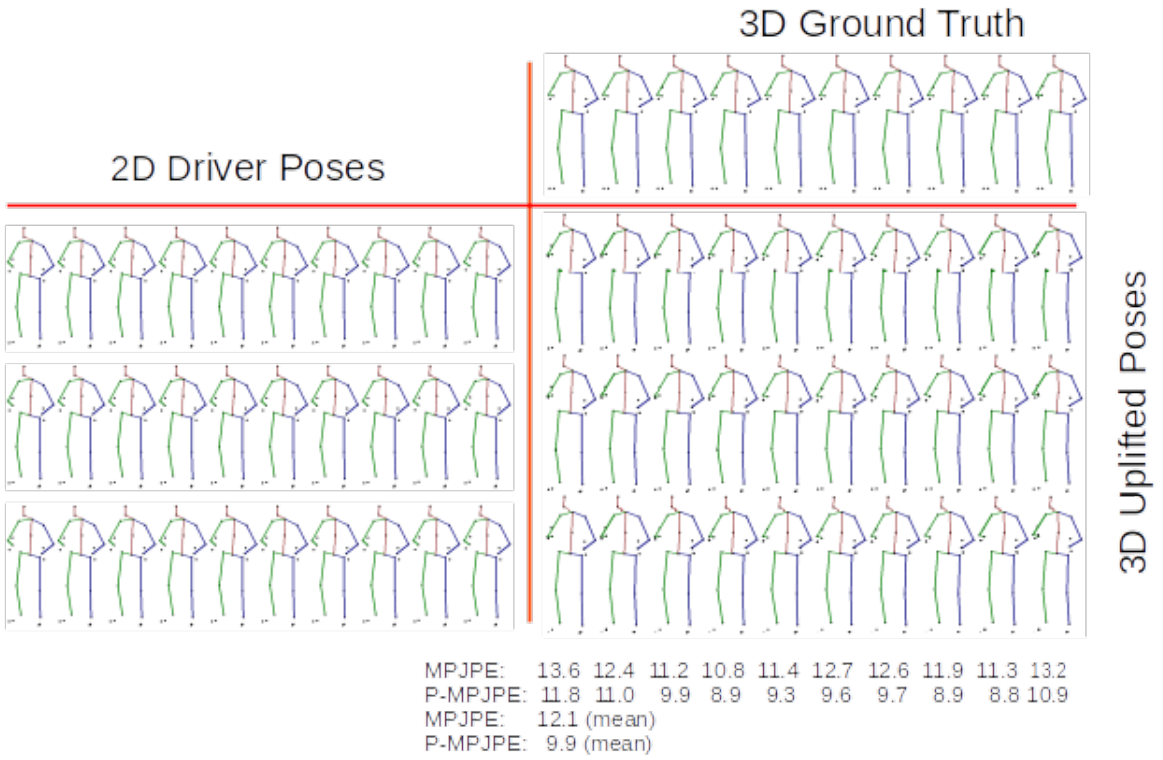


Figure 4.11: Subject 11, Activity 1 Directions (H36M)

5 to 10 lose position accuracy and orientation. The left arm remains raised but should be below shoulder level with the hand in the middle of the torso. Increasing MPJPE values indicates this, especially in poses 9 and 10 where the visual difference is obvious.

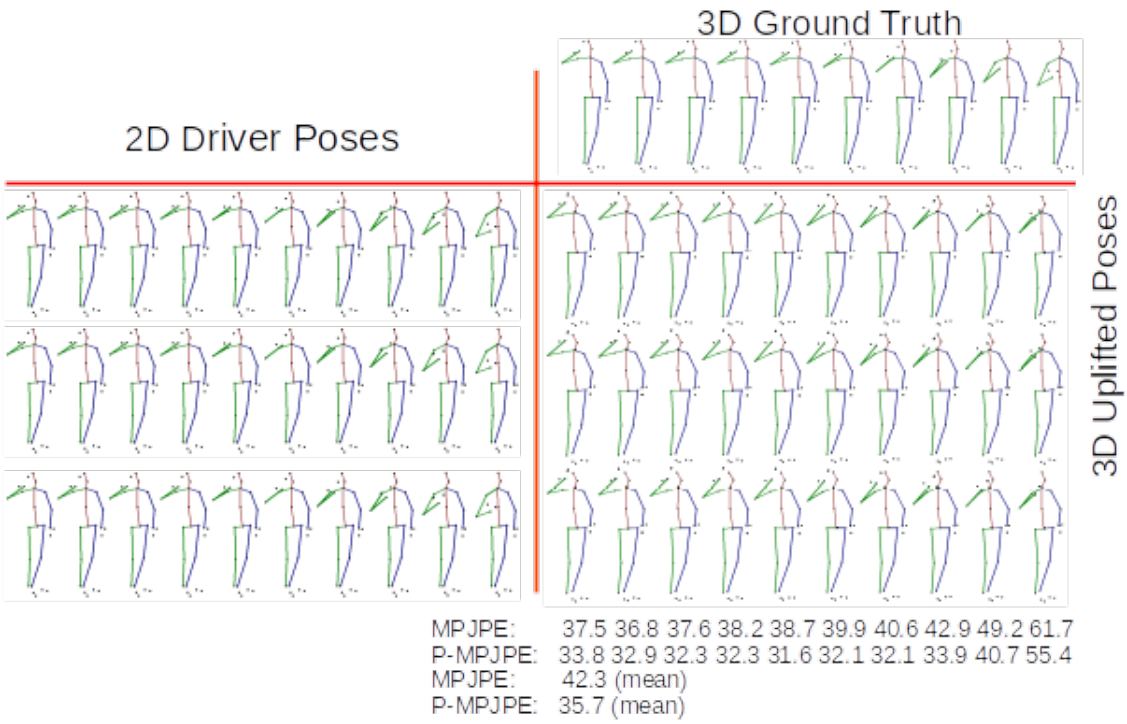


Figure 4.12: Subject 9, Activity 4 Greeting (H36M)

Figure 4.13 is a side on walking sequence with joint occlusions of the pelvis and knees. Poses 7 through 10 in the uplift has the pelvis rotated and the hips become visible, while in the ground truth hips remain occluded. In the same poses while the hips rotate the model is able to maintain the knee joint occlusion. Overall the MPJPE indicates the overall joint position is accurate through the uplifted poses.

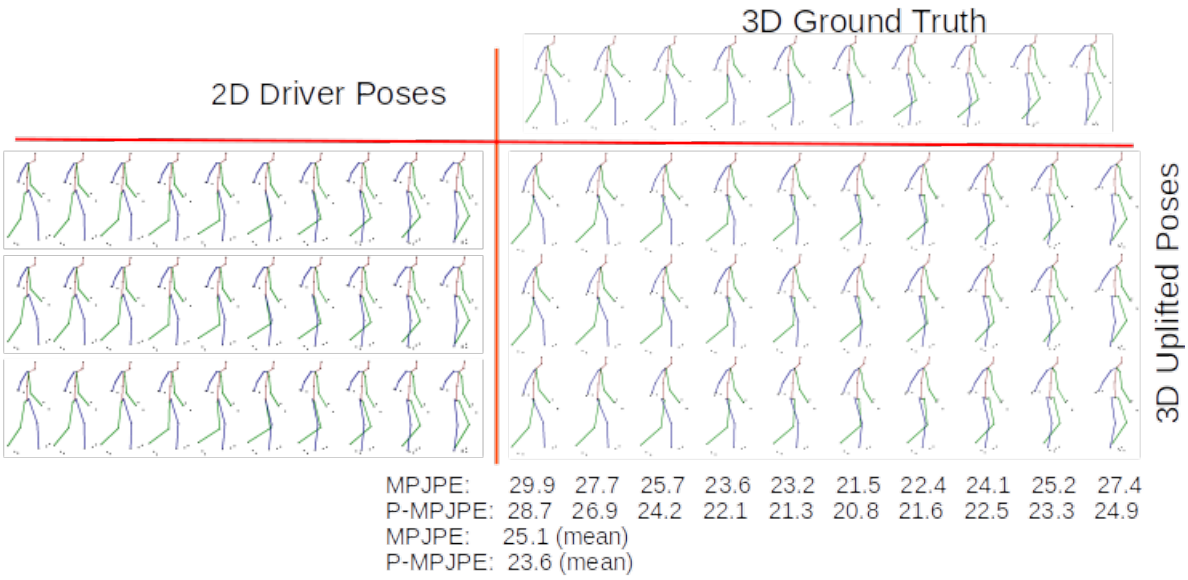


Figure 4.13: Subject 11, Activity 15 Walking (H36M)

Comparison with State-of-the-Art Methods:

Table 4.3 is a quantitative comparison of this method with other state-of-the-art methods using 2D ground truth poses. While not out performing on all actions it does achieve better than state-of-the-art for actions Directions, photo, sitting and Walking the dog. On average this model out performs PoseAug and Cascade and is within 3.6mm of best in class MHFormer.

Method	Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
VPose (CVPR 19) [37]	35.2	40.2	32.7	35.7	38.2	45.5	40.6	36.1	48.8	47.3	37.8	39.7	38.7	27.8	29.5	37.8
PoseAug (CVPR'21)[16]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.2
Cascade (CVPR'20)[28]	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Anatomy3D(TCSVT'21)[9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
SRNet (ECCV'20)[51]	34.8	32.1	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
PoseFormer (ICCV'21)[55]	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	**38.6**	31.7	31.5	29.0	23.3	23.1	31.3
MHFormer (CVPR'22) [29].	**27.7**	32.1	**29.1**	**28.9**	**30.0**	33.9	**33.0**	**31.2**	37.0	39.3	**30.0**	**31.0**	29.4	**22.2**	**23.0**	**30.5**
This method	29.1	**25.2**	35.8	32.9	30.6	**30.3**	43.6	43.3	**28.1**	**33.7**	**27.7**	**25.6**	**21.9**	23.8	27.3	30.6

Table 4.3: Comparison with state-of-the-art methods on Human3.6M under Protocol 1, (mode 1) in millimeters. Best results in bold.

2D to 3D Uplift and Prediction

This section contains mode 2 results, a 2D input sequence generating a 3D uplifted future prediction sequence. As with model 1 analysis, actions are selected to demonstrate the models capability on actions of different dynamic levels and occlusion. In this series of figures (4.14, 4.16, 4.18, 4.15, 4.17, 4.19), the top line is the 3D ground truth. Below the red line and left of the vertical line is the 2D prior sequence and to the right is the 3D uplifted prediction. Each line is an uplifted prediction with a different random z drawn from a uniform distribution. Figures (4.14, 4.16, 4.18) are trained and tested on the NTU dataset. Figures (4.15, 4.17, 4.19) results are trained and tested on the Human3.6M dataset.

Visual Analysis: Consistent with the finding in mode 1, static sequences are accurately 3D uplifted and the prediction accuracy consistent through the sequence. The "Drink Water" activity does lose right arm and hand joint positioning in the later poses, while the leg and torso positioning does remain accurate. The loss of accuracy is largely due to the arm positioning.

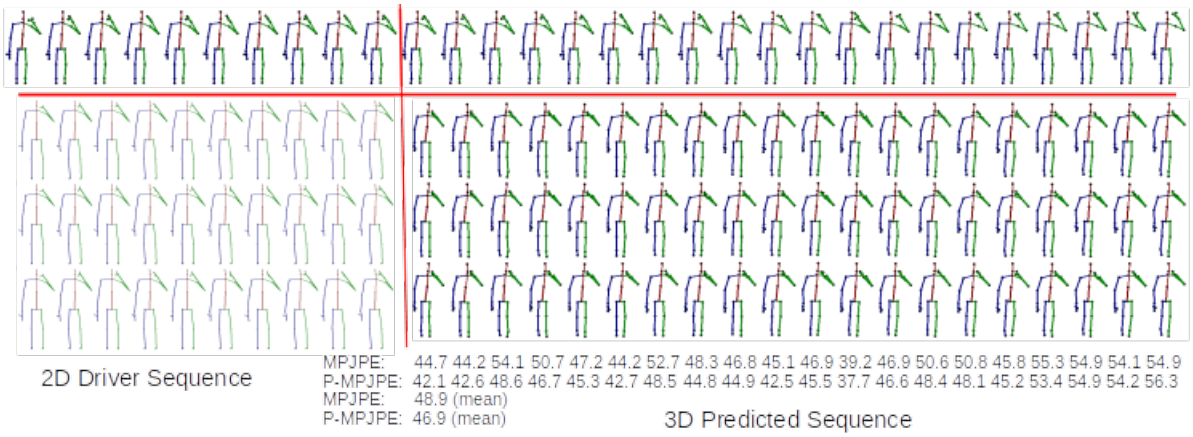


Figure 4.14: Subject 16, Activity 1 Drink Water (Prediction), (NTU)

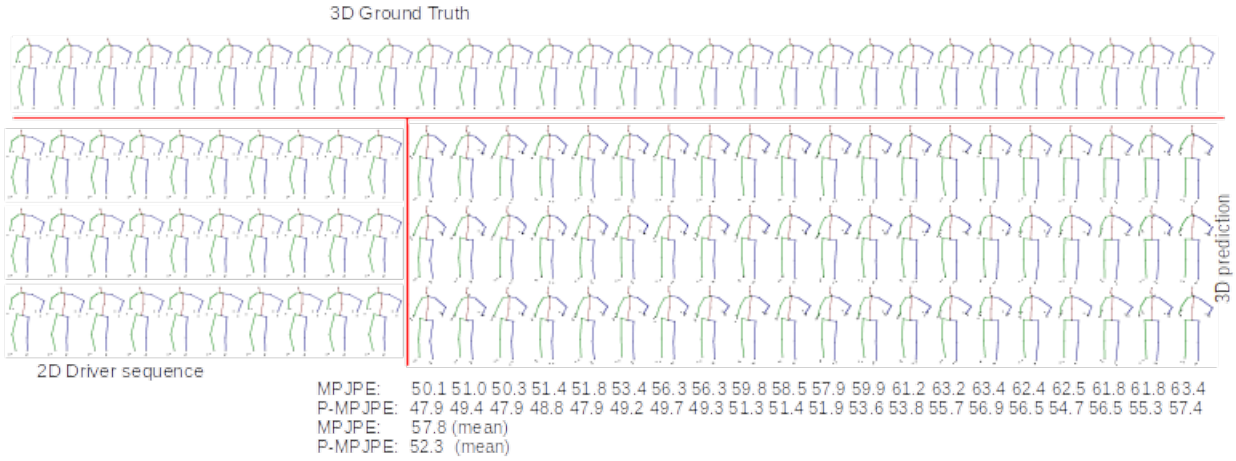


Figure 4.15: Subject 11, Activity 2 Directions (Prediction), (H36M)

Figures 4.16 and 4.17 are predictions on dynamic actions. In both sequences the first 3 to 4 predicted poses provide a good representation and the accuracy decreases for the following poses. Activity walking dog fails to predict the later poses where the subject crouches and the joint position becomes increasingly in inaccurate. "Cheer up" later poses in the prediction do visually resemble the ground truth, which is likely due to the less dynamic nature of the latter part of the sequence. The latter poses maintain joint accuracy.

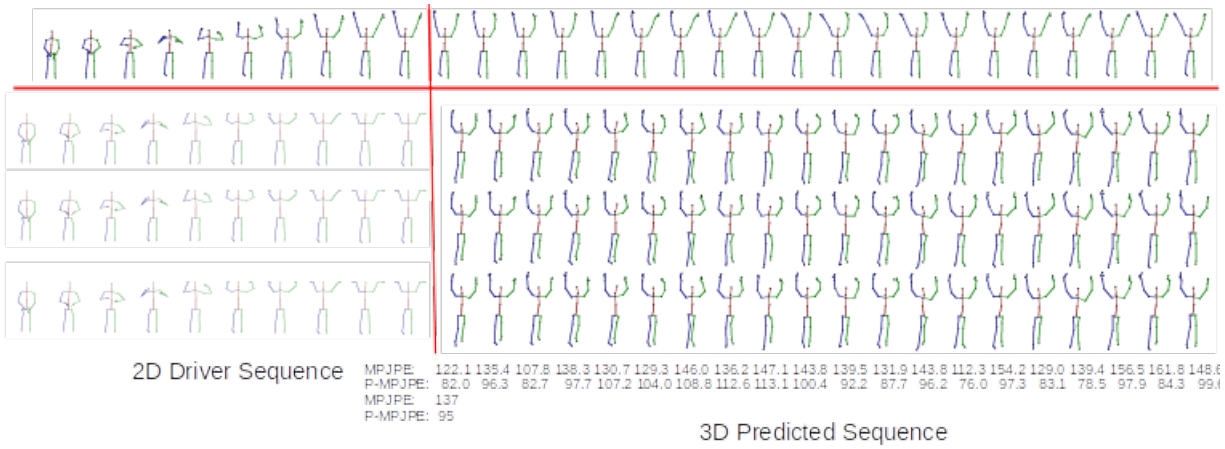


Figure 4.16: Subject 1, Activity 22 Cheer up (Prediction), (NTU)

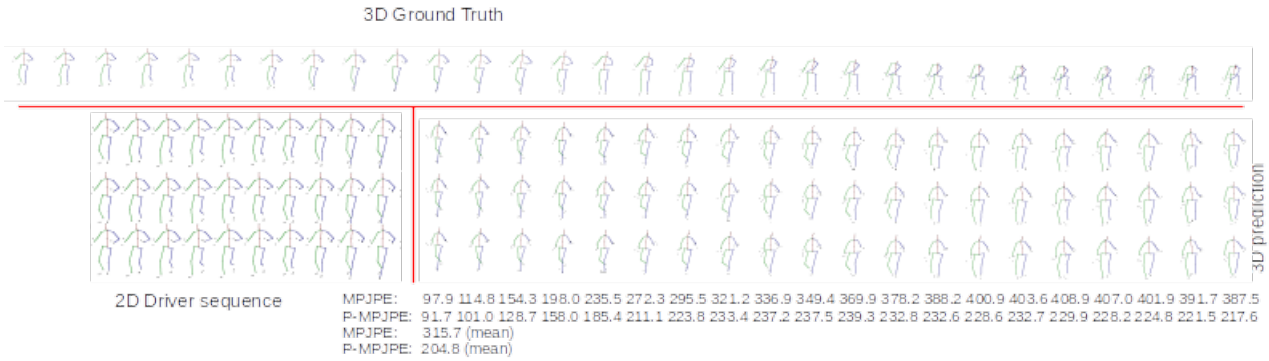


Figure 4.17: Subject 11, Activity 13 Walking Dog (Prediction), (H36M)

In action "take off jacket", the legs and arm joints are occluded. As in mode 1, the occlusions affect the models learning of movement in the legs and arms. The torso and head positions are predicted accurately while the arms and legs fail to keep position and orientation from the beginning of the prediction. Figure 4.19 contains a prediction of

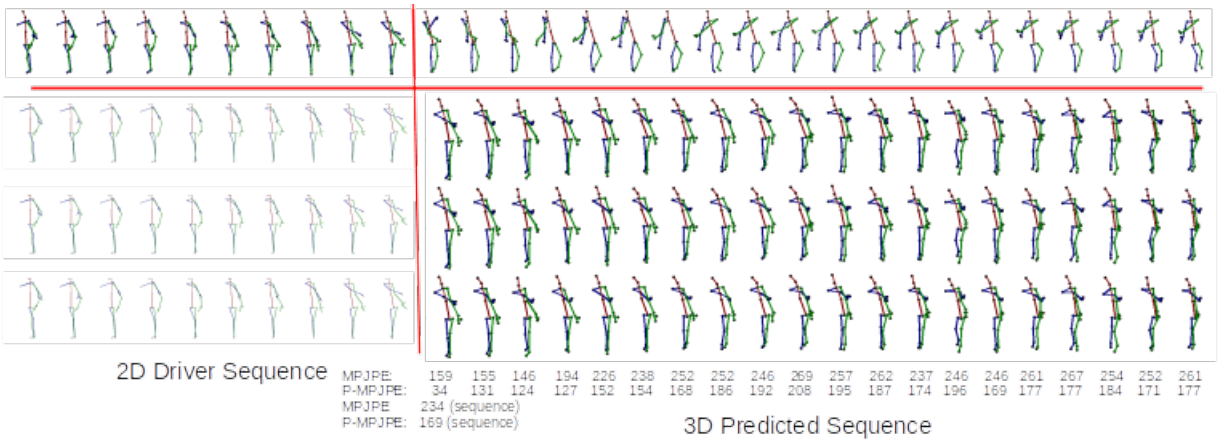


Figure 4.18: Subject 16, Activity 15 Take off jacket (Prediction), (NTU)

interest, and the prediction represents the turn of the subject to a side on stance, but while the ground truth stops walking (legs stop moving) the prediction keeps walking in a plausible sequence with a natural looking gait. A possible explanation for this is as the Human3.6M action sequences are long, in the order of 2,500 frames the data sampling is random, is it possible parts of a long sequence are not sampled and go

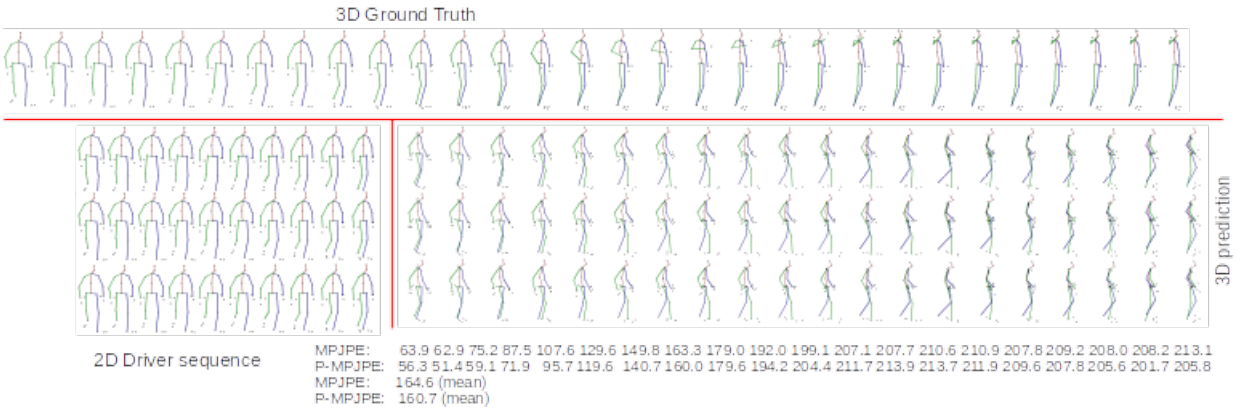


Figure 4.19: Subject 9, Activity 4 Greeting (Prediction), (H36M)

"unseen". As a result, the prediction is learned from part of the same or another sequence creating a synthetic or hybrid sequence.

4.1.6 Ablation Study

To determine the effects of various components of the model, a number of ablation experiments are conducted on mode 1 (2D to 3D uplift) with the Human3.6M dataset using protocol 1. The generator loss function (4.7) consists of three components, WGAN-GP, bone length loss and joint position loss. To determine the effect of bone loss and position loss, the model is trained first with bone loss removed, then with joint position loss removed, and again with both removed. Results in table 4.4. "All Losses" is the full loss function (4.7). Mean best and worst are MPJPE scores of the 10 best and 10 worst uplifts respectively. Removal of both the bone and position loss increases MPJPE by 1.6mm, 5.2% above the all losses result. While this does not seem large, given state-state-of-the-art, small improvements are needed. Bone and position loss contribute most on the worst performing uplifts. Without these losses, "mean worst" MPJPE increases by 4.1mm. A review of the worst performing uplifts will prove very

Generator Loss	MPJPE	P-MPJPE	Mean best	Mean worst
All Losses	30.6	26.2	11.4	110.6
Bone Loss, removed	31.8	26.6	12.4	121.4
Position Loss, removed	31.6	26.5	11.6	114.3
No Bone or Position Loss	32.2	26.8	12.6	114.7

Table 4.4: Ablation study results

useful further work, it may be possible to add additional loss to improve the worst uplifts. Between position and bone loss, position loss contributes most, and removal of position loss increases MPJPE results (less accurate) on both best and worst uplifts.

4.1.7 Additional Results

Real world applications of 2D to 3D uplift generally do not have skeleton data available without first processing imagery to extract the skeleton. For instance, an autonomous vehicle will observe pedestrians via the vehicles cameras in 2 dimensions, and the pedestrians 2D skeleton is estimated from the video. This estimation adds noise to the skeleton accuracy over observations made in controlled motion capture environments. To assess the models ability in a noisy environment, 5% uniformly distributed noise is injected into the test data. The model is trained on unmodified training data, as it is expected in a real world situation training data would be well curated. Table 4.5 compares the results between added noise and those without noise. The results on both MPJPE and P-MPJPE on average is 0.8mm and 0.6mm difference respectively, which indicates the model will perform well on skeleton data collected from video sources.

Method	Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
MPJPE																
No Noise.	29.1	25.2	35.8	32.9	30.6	30.3	43.6	43.3	28.1	33.7	27.7	25.6	21.9	23.8	27.3	30.6
5% noise.	29.8	25.9	36.5	33.8	31.5	31.0	44.3	43.4	29.4	34.6	28.5	26.4	22.7	24.7	28.1	31.4
Difference	+0.7	+0.7	+0.7	+0.9	+0.9	+0.7	+0.7	+0.1	+1.3	+0.9	+0.8	+0.8	+0.8	+0.9	+0.8	+0.8
P-MPJPE																
No noise.	25.6	22.0	31.4	27.8	25.3	27.1	44.5	33.1	25.8	27.8	22.9	20.6	17.1	18.5	23.0	26.2
5% noise.	26.1	22.5	32.0	28.6	25.9	27.6	44.9	33.3	26.9	28.5	23.7	21.3	17.9	19.3	23.7	26.8
Difference	+0.5	+0.5	+0.6	+0.8	+0.6	+0.5	+0.4	+0.2	+1.1	+0.7	+0.8	+0.7	+0.8	+0.8	+0.7	+0.6

Table 4.5: Added noise comparison on Human3.6M (mode 1) in millimeters

Table 4.6 compares the MPJPE results for mode 2, the values for the future 20 3D uplifted and predicted poses. As with the mode 1 results, the error difference in predicted sequences is small at 0.6mm, which indicates the model will work well on skeletons lifted from video.

Method	Dir	Disc	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
MPJPE																
No noise	131.3	130.5	115.7	151.3	118.8	152.1	136.3	125.2	110.8	146.5	128.8	130.8	163.5	116.7	126.0	132.3
5% noise	132.1	131.0	116.8	151.6	119.4	152.3	137.3	126.0	111.7	146.6	129.7	131.7	163.8	117.4	126.5	132.9
Difference	0.2	0.5	0.9	0.3	0.6	0.2	1.0	0.8	0.9	0.1	0.9	0.9	0.3.	0.7	0.5	0.6
P-MPJPE																
No noise	115.8	106.9	96.2	128.5	102.4	123.4	117.8	99.7	92.7	113.5	106.3	106.0	124.5	100.5	107.4	109.4
5% noise	116.6	107.6	96.8	128.9	103.0	123.7	118.7	100.2	92.7	113.7	106.9	106.4	124.6	101.1	107.6	109.9
Difference	0.8	0.7.	0.6	0.4	0.6.	0.3	0.9.	0.5	0.0	0.2	0.6	0.4.	0.1	0.6.	0.2	0.5

Table 4.6: Added noise comparison on Human3.6M (mode 2, prediction) in millimeters

4.1.8 Smallest and Largest MPJPE

This series of figures presents a sample of the best (4.20, 4.22) and worst (4.21, 4.23) performing uplifts and predictions by MPJPE result for both mode 1 and mode 2

using the Human3.6M dataset. The top line of each uplift is the ground truth, and below the red line, the 2D to 3D uplifted sequence. The best performing uplifts (4.20) have average joint errors in the order of 11.1mm. These uplifts are typically static, however as can been observed in the walking example the leg movement, is well uplifted. The poorest performing uplifts 4.21 are consistent with earlier observations and are sequences with joint occlusion and or dynamic sequences.

Figure 4.20: Mode 1 (uplift) sequences with smallest MPJPE on Human3.6M

Figure 4.21: Mode 1 (uplift) sequences with largest MPJPE on Human3.6M

Figure 4.22: Mode 2 (prediction) sequences with smallest MPJPE on Human3.6M

Figure 4.23: Mode 2 (predicted) sequences with largest MPJPE on Human3.6M

Chapter 5

Conclusion

Human pose estimation and prediction has been a focus of considerable research, experiencing continual progression in approaches and improvement in accuracy. The methods are diverse including temporal, body models and various deep learning methods utilising single poses and sequences of poses. More recently, generative adversarial network methods have found a place in this research also with a diversity of approaches.

To apply pose estimation and prediction of human movement in real world applications such as autonomous vehicles, 2D representations collected from video needs to be lifted into a 3D representation. This thesis builds upon the 3D to 3D generative pose prediction approach of HP-GAN. Utilising HP-GAN's recurrent encoder/decoder GAN structure and adding 2D to 3D uplift by introducing the noise vector as the z joint position dimension and a loss function for constraining the location of joint locations and bone lengths.

This approach achieves better than state-of-the-art in seven of fourteen activity types on the Human3.6M dataset and within 0.1mm of state-of-the-art on average over all activity types. Further the same model also generates realistic future pose sequences representing the future human motion.

5.0.1 Future work

During ablation testing it was observed some uplifts experience high MPJPE rates, this is visually noted in figure 4.21. While these uplifts maintain visually recognisable actions the joint accuracy is low. Improving these activities with high MPJPE results is likely to reduce the overall MPJPE to better than state-of-the-art. Viable approach's to investigate are data sampling approaches, additional loss function to correct for the error in these cases, and swapping the RNN for a transformer.

Bibliography

[1] Cmu graphics lab motion capture database.

[2] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. Applications of generative adversarial networks (gans): An updated review. *Archives of computational methods in engineering*, 28(2):525–552, 2021.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[4] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[5] Neeraj Battan, Yudhik Agrawal, Sai Soorya Rao, Aman Goel, and Avinash Sharma. Glocalnet: Class-aware long-term human motion synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 879–888, January 2021.

[6] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[7] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks, 2017.

[8] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning, 2017.

[13] Dylan Drover, Rohith MV, Ching-Hang Chen, Amit Agrawal, Ambrish Tyagi, and Cong Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[14] Xiaoxiao Du, Ram Vasudevan, and Matthew Johnson-Roberson. Bio-lstm: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction. *IEEE Robotics and Automation Letters*, 4(2):1501–1508, 2019.

[15] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8575–8584, June 2021.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 2014.

[18] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. Best of Automatic Face and Gesture Recognition 2008.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local

nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[22] Jonathan Schwarz Joe Yearsley Ikhsanul Habibie, Daniel Holden and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In Gabriel Brostow Tae-Kyun Kim, Stefanos Zafeiriou and Krystian Mikolajczyk, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 119.1–119.12. BMVA Press, September 2017.

[23] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[24] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks, 2017.

[25] Brendan F. Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[26] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. *arXiv preprint arXiv:2111.12707*, 2021.

[30] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis, 2018.

[31] Shengyuan Liu, Pei Lv, Yuzhen Zhang, Jie Fu, Junjin Cheng, Wanqing Li, Bing Zhou, and Mingliang Xu. Semi-dynamic hypergraph neural network for 3d pose estimation. In *IJCAI*, pages 782–788, 2020.

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[33] Diogo C. Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers Graphics*, 85:15–22, 2019.

[34] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks, 2017.

[35] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning, 2019.

[37] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael 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.

[38] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[40] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[41] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.

[43] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019. IEEE, 2016.

[44] Jun Sun, Mantao Wang, Xin Zhao, and Dejun Zhang. Multi-view pose generator based on deep learning for monocular 3d human pose estimation. *Symmetry*, 12:1116, 07 2020.

[45] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *In Advances in Neural Information Processing Systems 27*, volume 27, pages 3104–3112, 2014.

[46] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N. Metaxas. Cr-gan: Learning complete representations for multi-view generation, 2018.

[47] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] Zhiyong Wang, Jinxiang Chai, and Shihong Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. *IEEE transactions on visualization and computer graphics*, 27(1):14–28, 2021.

[49] Yuan Xue, Tao Xu, Han Zhang, L. Rodney Long, and Xiaolei Huang. Segan: Adversarial network with multi-scale l1 loss for medical image segmentation. *Neuroinformatics*, 16(3):383–392, 2018.

[50] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016.

[51] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 507–523, Cham, 2020. Springer International Publishing.

[52] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7354–7363. PMLR, 09–15 Jun 2019.

[53] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, 2012.

[54] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris 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.

[55] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, October 2021.

[56] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. Pose-invariant embedding for deep person re-identification. *IEEE Transactions on Image Processing*, 28(9):4500–4509, 2019.

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.