Neural Computing and Applications (2020) 32:14579–14591
https://doi.org/10.1007/s00521-020-04941-4

S.I.: DEEP LEARNING APPROACHES FOR REALTIME IMAGE SUPER RESOLUTION (DLRSR)

GAN-Poser: an improvised bidirectional GAN model for human motion prediction

Deepak Kumar Jain[1] · Masoumeh Zareapoor[2] · Rachna Jain[3] · Abhishek Kathuria[3] · Shivam Bachhety[3]

Received: 16 June 2019 / Accepted: 8 April 2020 / Published online: 29 April 2020

Abstract
A novel method called GAN-Poser has been explored to predict human motion in less time given an input 3D human skeleton sequence based on a generator–discriminator framework. Specifically, rather than using the conventional Euclidean loss, a frame-wise geodesic loss is used for geometrically meaningful and more precise distance measurement. In this paper, we have used a bidirectional GAN framework along with a recursive prediction strategy to avoid mode-collapse and to further regularize the training. To be able to generate multiple probable human-pose sequences conditioned on a given starting sequence, a random extrinsic factor Θ has also been introduced. The discriminator is trained in order to regress the extrinsic factor Θ, which is used alongside with the intrinsic factor (encoded starting pose sequence) to generate a particular pose sequence. In spite of being in a probabilistic framework, the modified discriminator architecture allows predictions of an intermediate part of pose sequence to be used as conditioning for prediction of the latter part of the sequence. This adversarial learning-based model takes into consideration of the stochasticity, and the bidirectional setup provides a new direction to evaluate the prediction quality against a given test sequence. Our resulting novel method, GAN-Poser, achieves superior performance over the state-of-the-art deep learning approaches when evaluated on the standard NTU-RGB-D and Human3.6 M dataset.

Keywords Human motion · GAN · Probability theory · Pose estimation · Sequence model · 3D model

1 Introduction

An accurate and short (several seconds) predictions of what is going to happen within the world given past events may be an elementary and helpful human ability. Such ability is important for daily activities, social interactions and ultimately survival. As an example, driving needs predicting alternative cars' associated pedestrians' motions so as to avoid an accident; greeting needs predicting the situation of the opposite person's hand, and taking part in sports needs predicting other players' reactions. So as to form a model that may act seamlessly with the world, it desires the same ability to grasp the dynamics of the human world and to predict probable futures supported learned history and therefore the immediate gift. However, the long run is not settled, thus predicting the long run cannot be settled, except within the terribly short term. Due to the fact that the predictions extend additional into the long run, uncertainty becomes higher. Individuals walking might flip or fall; individuals throwing a ball might drop it instead. However, some predictions are additional plausible than others and have a better chance. Recently, deep learning is used in medical image processing, inspired by hopeful results in computer vision and medical imaging. The usage of learning-based techniques in image recording, however,

✉ Deepak Kumar Jain
deepak@cqupt.edu.cn

[1] Key Laboratory of Intelligent Air-Ground Cooperative Control for Universities in Chongqing, College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China

[2] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

[3] Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India

has been imperfect. Some standard-space recording tasks are agreeable to learning and may deliver momentous enhancement over approaches such as iterative optimization or grid exploration when the choice of plausible pose/alignment is wide, challenging a large capture range [1]. Under these circumstances, a human witness can find the estimated pose of 3D objects rapidly and carry them into rough alignment without resolving an iterative optimization via feature recognition. In our work, we tend to concentrate on making a model that may predict a plausible future human (skeleton) pose from a given past. The number of poses taken from the immediate past, and therefore, the foreseen range of 3D poses within the future, which might be unrestricted, is fed as parameters to the model. Here, we tend to address this by adding a geodesic loss so as to stabilize and improve the training. To quantitatively assess the standard of the nondeterministic predictions, we tend to simultaneously train a motion-quality-assessment model that learns the chance that a given skeleton sequence may be a real human motion.

We tend to take a look at our motion prediction model on a massive dataset captured with a special modality. However, GANs have weaknesses. They will be troublesome to coach and unstable in their learning, their loss value does not essentially indicate the standard of the generated sample, and therefore, the coaching will collapse simply. Recent literature tries to enhance GAN coaching and supply a theoretical warranty for its convergence. We tend to train our model on all actions right away; thus, its output is not conditioned on any specific act. Our model takes as input a 3D sequence of previous poses and a random vector z from the reduced sequence area that samples attainable future poses. For every such z value, the model generates a special output sequence of attainable future poses. We tend to associate RNN for the generator as RNNs are a category of neural networks designed to model sequences, particularly variable-length sequences. Our main contributions are: (1) the model tends to propose a completely unique human motion model that may predict attainable future from the past. (2) It also tends to propose a motion-quality-assessment model to quantitatively assess our results. (3) We have used bidirectional GANs with geodesic loss and recursive prediction strategy to reduce overfitting and take into consideration of the stochasticity in the prediction of future pose sequence.

1.1 Organization of paper

Section 2 explains the related research in GANs and other deep learning techniques as per works done by different authors in human motion estimation. Section 3 describes the proposed approach that was used during this research. Section 4 includes the methodology of the model and its parameters in the training. It also describes the dataset used. Section 5 shows the evaluation tests and results that were obtained with respect to other models of RNN and similar techniques. The last section presents the conclusion with accuracies and error rates in model and possible directions of future work.

2 Related works

To accomplish the pose prediction, an improved Wasserstein generative adversarial network (WGAN-GP) [2] was formulated with a custom loss perform that took into account of the human motion and human anatomy. The generator was a novel adaptation of sequence-to-sequence model [3] of poses derived from a recurrent neural network (RNN), and therefore the critic and discriminator are a multilayer network (MLP). It intended to use the critic network to coach the generator, and therefore, the soul network to be told identifying between true sequences of poses from a faux one. In essence, there was a mix of some sides of the initial GAN [4] with WGAN-GP [5]. This was successfully employed in computational linguistics [6], caption generation from pictures [7], video classification and action recognition [1, 3], action detection, video description [8] and sequence prediction [9].

Since the introduction of the Kinect sensor [10], many works on recognizing human action and predicting human poses from skeleton data have been done. For example, prediction of human poses trained on previous poses using deep RNNs [11, 12] as these large human motion datasets [13, 14] are available. Adversarial learning was first described by Goodfellow [15]. This was followed up by the deep convolutional GAN [16] which popularized the technique with realistic visualizations and stabilized training efforts. While Pix2Pix forms the backbone for one of the proposed methods for our study for conditional generation of video frames, there were prior studies GANs conditioned on various forms of supervision. Generative adversarial networks have shown impressive performance in image generation [17], video generation [18, 19] and other domains [20–22]. The key idea in GANs is an adversarial loss that forces the generator to fool the discriminator. Instead of developing new GAN objective functions as is normally the case, our goal here to investigate how to improve human motion prediction by leveraging the GAN framework. Recent human motion prediction, which relies on deep RNNs [11, 12, 23] or deep neural networks [24], is primarily deterministic. In [11], the authors mix both deterministic and probabilistic human motion predictions. Their deterministic aspect is based on a modified RNN called recurrent decoder (RD) that adds fully connected layers before and after an LSTM [25] layer

and minimizes a Euclidean loss. Their probabilistic aspect uses a Gaussian mixture model (GMM) with five mixture components and minimizes the GMM negative log-likelihood. In [12], Butepage developed a general framework that converts a structure graph to an RNN, called a structure RNN (S-RNN). They test their framework on different problem sets including human motion prediction and showed that it outperforms the current state of the art. However, they need to design the structure graph manually and task specifically. The authors examine recent deep RNN methods [7] for human motion prediction and show that they achieve state-of-the-art results with a simpler model by proposing three simple changes to RNN. On the other hand, Martinez [24] used an encoder–decoder network based on a feed-forward network and compare the results of three different such architectures: symmetric, time-scale and hierarchical.

3 GAN-Poser: the proposed approach

To predict human motion, a sequence of human poses is given as input to the system to predict future poses that are valid. Our goal is to observe the probability of the future sequence based on input sequence $P(z|x)$ where the sequence of input poses is $x = \{x_1, x_2, \ldots, x_m\}$ and the sequence of predicted poses is $z = \{z_1, z_2, \ldots, z_n\}$ given that each z_i and x_j represent a single pose. Our proposed model, GAN-Poser, uses a bidirectional GAN framework in which the encoder and the generator are not connected with a compressed code. We also use the recursive prediction strategy along with the geodesic loss to reduce the overfitting and take into consideration of the stochastic nature of the model. The given prediction model is a revised version of the sequence-to-sequence network. A sequence-to-sequence network has 2 parts: an encoder and a decoder where a giant network is created by two completely different networks. While this model besides taking a sequence of human poses as input also takes a z vector which is Gaussian distribution [26]. After drawing and mapping z, it is added to the encoder states. It is all done in the same space as the output states of the encoder. The result is used as the initial state of the decoder. This means that the last output of the encoder becomes the first input of the decoder. GRU is used for our sequence-to-sequence network.

For a given input pose x, each value of z provides a different and valid future pose. It also depends on the network parameters that the system needs to learn. Over the years, the problem of human motion prediction is viewed as a regression problem. Even the most recent deep RNNs view it in the same light. But this approach has its flaws. Because it only learns one outcome at a time, with an increase in the length of the future sequence, the probability of this particular outcome decreases. Moreover, image enhancement operators are trained in a weakly supervised method via adversarial learning motivated by aesthetic decision.

3.1 Architecture

Figure 1 depicts the architecture of our model, GAN-Poser. Initially, a 3D input image dataset is fed into the model.

After that, the preprocessing step is performed wherein we use a helper function for splitting the data and a filter function which is used a refractor filter. Finally, a vectorized dataset is obtained. The next step in the architecture of our proposed model, GAN-Poser, is the usage of a random vector z which is fed to the generator network. The discriminator uses the vectorized form of the dataset and gives the function $D(x)$ which helps in the identification of the plausible and the non-plausible human action poses across the whole network. The cost is determined and by using the stochastic gradient functions of the discriminator and the generator, the model is trained. We also use the recursive prediction strategy to train the generator and the discriminator. Finally, the model predicts future actions based on the input by calculating the sampling as well as the geodesic loss.

3.2 GANs training algorithm

The GANs training algorithm consists of two parts: (1) when the discriminator is trained at that time the generator is at rest. In this stage, the system is only forward propagated and no back-propagation is ready. Real Data are used to train the discriminator for n epochs, to make it correctly predict them, and it is also trained on the false produced data from generator and to make it predict it as false. (2) in the next phase, The generator is trained and by the time discriminator is at rest. The results of training the discriminator on fake data by generator are used to train the generator and to get better at each step and fool the discriminator. The algorithm is repeated for a few epochs and then automatically checked the false data if it seems unpretentious. The training is stopped as soon as the data are seen acceptable; otherwise, the epochs continue to go on. The GANs are represented as a minimax algorithm, where the discriminator is demanding to minimalize its reward $V(D, G)$ and the generator is requiring reducing the discriminator's return or in other words, getting the best out of its loss. It can be mathematically defined by Eq. 1:

$$\min_G \max_D V(D, G) \tag{1}$$

where $V(D, G)$ is defined in Eq. 2 as:

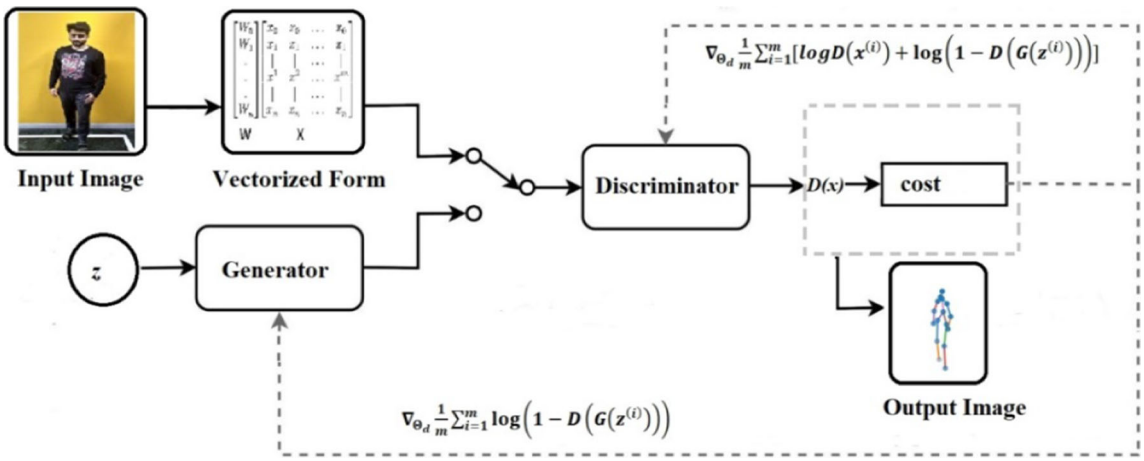

Fig. 1 Architecture of the GAN-Poser model for human pose prediction

$$V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z))] \quad (2)$$

here G, D stands for generator and discriminator, respectively. Moreover, $p_{\text{data}}(x)$ is the distribution of the real data, and $p_z(z)$ is the distribution of the generator network. The variables x and z are the samples from $p_{\text{data}}(x)$ and $p_z(z)$, respectively. The functions $D(x)$ and $G(z)$ are the network distributions for the discriminator and the generator, respectively. Now, the algorithm for Minibatch stochastic gradient descent for the training of the GANs has been discussed as follows:

for numbers of iterations **do**:

for k steps **do**:

- sample minibatch of p noise samples $\{z_1, \ldots ..z_p\}$ from noise prior p_g (z)
- sample minibatch of p examples $\{x_1, \ldots, x_p\}$ from data distribution p_{data} (x)
- update discriminator by rising its stochastic gradient:

$$\nabla_{\Theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \quad (3)$$

end for

- sample minibatch of p noise samples $\{z_1, \ldots, z_p\}$ from noise prior p_g (z)
- update generator by rising its stochastic gradient:

$$\nabla_{\Theta_d} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \quad (4)$$

end for

- update using any standard gradient based rule learning.

3.3 Bidirectional generative adversarial networks

Generative adversarial network (GAN) is an unsupervised learning technique which is motivated by the minimax theorem. In this, both the participant networks namely the generator and the discriminator networks compete and try to outperform each other. The training itself vacillates between both networks [27]. According to the original paper [7], the function of the generator is to learn to generate images close to real images while and the discriminator distinguishes between the generated image and the real image. In a steady-state, there is a likelihood of 50% of the discriminator for predicting whether the generator network generates an image or not.

In our proposed model, we have used the bidirectional GAN [28–30] or BI-GAN along with the geodesic loss. The bidirectional GAN is, in simple words, a higher version of the autoencoder at its core. In an autoencoder, an image is rendered to the encoder which produces the in-between compressed code, 'c', which is then sent to the decoder as an input to reconstruct the original image. Whereas, in the bidirectional GAN, the encoder is supplied with a 3D image x and the encoder generates a compressed code that is denoted by 'c'. Then, this compressed code is supplied into the decoder which produces a reconstructed image $\bar{x}$. The only difference between the autoencoder and the bidirectional GAN is that in the Bi-GAN; the encoder and the decoder are not connected by a single compressed code c. The input image provided to the encoder in case of bidirectional GAN is sampled from the probability distribution function $p_x(x)$, and the compressed code c is sampled from the probability distribution function $p_c(c)$. This probability distribution function uses the latent code distribution. But the main predicament arises that since the encoder and the decoder are not connected, we cannot train

our bidirectional GAN model. Hence, we use a discriminator for this purpose. It takes in a pair of images and the compressed code. Now, this can come from either decoder or encoder; therefore, if the discriminator generates an output value of 1, this means that the input is obtained from the encoder and similarly, if the out generated by the discriminator is 0, the then input is obtained by the decoder. In the original GAN algorithm, Jensen–Shannon (JS) divergence is used as its loss function which degrades its efficiency and makes it difficult to train. The presence of its ratio between two probabilities that might not overlap initially can cause the JS to be zero or infinity. It can cause vanishing gradients. It uses the same input poses to learn multiple possible futures poses using different z values. JS distance is replaced with the Earth Mover Distance (EMD), which efficiently keeps a balance between training the discriminator versus the generator.

The discriminator in WGAN discriminates between neither synthetic inputs nor real neither input nor provides a probability as output. The properties of an adversarial training scheme such as it allows the generation of multiple futures from a single past compelled us to use it. Apart from this, without explicitly using the ground truth of the real future, the generator can still be trained to give predictions based on data where the cost function is learnt implicitly.

3.4 Objective function for bidirectional GAN

For the generator, we take an assumption that we have a prior belief on where the latent space z lies, that is $P_c(C)$. We draw from the latent space generator G which gives the synthetic output. This is shown by Eq. 5:

$$G(c, \theta_G) : c \rightarrow x_{\text{synthetic}} \tag{5}$$

In above Eq. 5, the parameters of the generator network G are given by θ_G which are the variable parameters. These variable parameters will be optimized during the back propagation of the neural network. When the input c is given, the network produces the output $x_{\text{synthetic}}$ as a synthetic image. The encoder is just the inverse of the generator. If given a draw from the data space $P_x(x)$, the output of the encoder is a real image. This is shown by the following equation:

$$E(x, \theta_E) : x \rightarrow c \tag{6}$$

In Eq. 6, when x is parameterized by θ_E and the encoder E takes x as an input, it generates a real image. Here, c denotes the real encoding.

Next is the discriminator which aims to classify if the sample is real or synthetic. This means that it specifies if a sample is from the real distribution $P_x(x)$ or the synthetic data distribution $P_G(x/c)$. Moreover, it also aims to classify if a encoding is real $P_E(c/x)$ or synthetic $P_c(C)$.

The objective function of the bidirectional GAN is given as follows:

$$\begin{aligned} V(G, D, E) = {} & E_{x \sim p_z}[\log D(x, E(x))] \\ & + E_{z \sim p_z}[\log(1 - D(G(z), z))] \end{aligned} \tag{7}$$

In Eq. 7, D and E refer to the decoder and the encoder, respectively, and G is the generator network. $E(x)$ maps the examples from the data space, denoted by x, to the latent space, denoted by z. The main aim for using the objective function to train is to solve the min–max problem which is given by Eq. 8:

$$\min_{G,E} \max_{D} V(G, D, E) \tag{8}$$

As it can be clearly seen in Eq. 8, we are using the generator and the encoder block for minimizing as they are trying to fool the discriminator. Moreover, we are maximizing the discriminator block which is denoted by D. In our model, GAN-Poser, the following optimized loss function is used for updating the parameters of the discriminator D which is shown by Eq. 9:

$$L_D = E_{x \sim p_x}[\log D(x, E(x)))] + E_{z \sim p_z}[\log(1 - D(G(z), z))] \tag{9}$$

Furthermore, for the updation of the parameters of the generator and the encoder, the following loss function has been optimized which is shown by Eq. 10:

$$L_{EG} = E_{z \sim p_z}[\log D(G(z), z)] + E_{x \sim p_x}[\log(1 - D(x, E(x)))]. \tag{10}$$

3.5 Recursive prediction strategy

To further regularize the training, we introduce a recursive prediction strategy [31, 32]. In spite of being in a probabilistic framework, the enhanced discriminator architecture concedes predictions of an intermediate part of pose sequence to be used as conditioning for prediction of the latter part of the sequence. To accomplish this task, we have used the recursive error prediction algorithm. For calculation, first, the prediction error-index is calculated by the following formula:

$$J_{t|\theta} = \frac{1}{2} \sum_{k=1}^{t} \left[\tilde{z}_{k|\theta}^{T} \Lambda_{k|\theta}^{-1} \tilde{z}_{k|\theta} + \log \det \Lambda_{k|\theta} \right] \tag{11}$$

In Eq. 11, θ is the extrinsic factor, $(\tilde{z}_{k|\theta})$ is the prediction error estimate for the next step and $\Lambda_{k|\theta}$ is the weight matrix. In a special case, the prediction error estimate for the next step is used from the Kalman filter as $z_{k|k-1,\theta} = E\{z_0, z_1, \ldots, z_{k-1}, \theta\}$. In a case where $\Lambda_{k|\theta}$ is independent

of the extrinsic factor θ, the simplified function comes out to be as follows:

$$\bar{J}_{t|\theta} = \frac{1}{2}\sum_{k=1}^{t}\left[\tilde{z}_{k|\theta}^{T}\Lambda_{k}^{-1}\tilde{z}_{k|\theta}\right] \tag{12}$$

This function which is shown in Eq. 12 is used to keep in check the errors when the recursive algorithm is applied where the immediate predicted part is used for the prediction of the rest of the sequence. In our research, we have trained the discriminator to regress the above-mentioned extrinsic factor θ, which is eventually used to generate a particular pose sequence.

3.6 Sampling-based loss

The sampling-based loss consistently achieves motion prediction error viable with or better than the state of the art. Moreover, the model has been trained to reduce the error over a 1-second time domain as the network retains the capability to produce probable motion in the long term. Since the proposed sampling-based loss does not require any hyper-parameter tuning, we can infer that it is a fast training technique to previous work for long-term motion generation using GANs. Importance sampling has been used in deep learning mainly in the form of manually tuned sampling schemes. Bengio [33] manually designed a sampling scheme inspired by the perceived way that human children learn; in practice, they provide the network with examples of increasing difficulty in an arbitrary manner. Diametrically opposite, it is common for deep embedding learning to sample hard examples because of the plethora of easy non-informative ones [34]. For the explanation of the sampling based loss, the input is taken as a_i and the output is taken as b_i. The main aim is to minimize the loss function used in the following Eq. 13:

$$\bar{\Theta} = \arg\min_{\Theta}\frac{1}{n}\sum_{i=1}^{n}\zeta(\psi(a_i,\Theta),b_i) \tag{13}$$

In Eq. 13, ζ refers to the loss function which has to be minimized and ψ refers to the deep learning model which has been parameterized by the vector Θ. Here, n refers to the number of examples used in the training set and $\bar{\Theta}$ is the optimal parameter vector. For finding the parameter vector for the iteration $t+1$, we use the stochastic gradient descent procedure. The iteration t will depend upon the sampling distribution $\{p_1^t,\ldots,p_n^t\}$ and the rescaling coefficients $\{w_1^t,\ldots,w_n^t\}$. The parameter vector for the iteration $t+1$ is calculated by the formula used in Eq. 14:

$$\Theta_{t+1} = \Theta_t - \eta w_{I_t}\nabla_{\Theta_t}\zeta(\psi(a_{I_t},\Theta_t),b_{I_t}) \tag{14}$$

here I_t is the data point which is sampled at each step of the iteration.

3.7 Geodesic loss

The proposed deep learning model with geodesic loss minimization can attain precise outcomes with a wide capture array in real-time. The loss function for the training purpose is defined as:

$$L_{\text{total}} = L_{\text{rotation}} + \mu L_{\text{translation}} \tag{15}$$

where μ is a hyper-parameter to balance between the rotation loss L_{rotation} in the range from 0 to π, and the translation loss $L_{\text{translation}}$. It is the mean-squared error (MSE) between the actual and predicted translation vectors. For the training stage, we have used the geodesic loss compared to MSE loss. The MSE loss is defined as:

$$L_{\text{MSE}} = \|v-u\|^2 \tag{16}$$

In Eq. 16, u and v are the output of the rotation head and the actual rotation, respectively. MSE helps to reduce the search space for pose calculation but has less accuracy for rotation between distances. The distance between two 3D rotations is geometrically interpreted as the geodesic distance between two points on the unit sphere. It is the radian angle between two viewpoints or the shortest distance which has an exponential form. Let R_i and R_j be two rotation matrices to measure a distance between two points, that is, the 3D angle between these rotations. The amount of rotation needs to be applied on rotation matrix R_i to reach rotation matrix R_j and is calculated in Eq. 17 as:

$$d(R_i,R_j) = \left\|\log\left(R_i^{\mathrm{T}}R_j\right)\right\|F \tag{17}$$

here F is the Frobenius norm and $\left\|\log\left(R_i^{\mathrm{T}}R_j\right)\right\|$ is the matrix logarithm of a rotation matrix. The distance between R_i and R_j can be represented as rotation matrix as given in Eq. 18:

$$\mathrm{tr}(R) = 1 + 2\cos(\theta) \tag{18}$$

where θ is equal to $\cos^{-1}\left[\frac{\mathrm{tr}\left(R_i^{\mathrm{T}}R_j\right)-1}{2}\right]$.

Therefore, the geodesic loss which is defined as the distance between two rotation matrices can be written as given in Eq. 19.

$$L_{\text{geodesic}} = d(R_iR_j) = \cos^{-1}\left[\frac{\mathrm{tr}\left(R_i^{\mathrm{T}}R_j\right)-1}{2}\right] \tag{19}$$

This is a natural Riemannian metric [35] to calculate the geodesic loss.

4 Methodology

4.1 Preprocessing

A helper function was created which split the training data and generated a CSV file containing the list for the training data. The path of each of the input file which contained the video clips was obtained, and the list of all the folders and sub-folders was created. Moreover, each file name was taken without any extension and only the first names of each file were considered. Then, the data were randomly split into training, testing and validating datasets and the target CSV file was generated using file generator function. A filter data function was also used which was used as a refactor filter. After that, we normalized the x, y, z values of each joint in the range -1 and $+1$. We obtained the range of the raw data by finding the minimum and maximum values on each dimension and then computing the minimum and maximum values over all the dimensions. Then, we first used it for the training data. In the next set, to avoid ambiguities between the camera and 3D pose rotation, all the scaling components from the 3D poses are excluded. This is done by aligning every 3D pose to a template pose. We do this by assessing the ideal scale for the corresponding shoulder and hip joints, and the resulting transformation is applied to all joints. The residual scale variations are compensated by the camera scale component. In contrast to the example given in [16], we do not need to know the mean and standard deviation of the training set. This allows for an easy transfer of our method to a different domain of 3D poses.

4.2 Evaluation metrics

Evaluation of the performance comparison is done using the mean angle error measurement as seen in [27, 36], which used Euclidean distance between the predicted motions and ground truth motion in angle space. Rotation and translation of the whole body are excluded since this information does not depend on the actions. We have used the predictions frame by frame for the visualization of the model. Moreover, as most of the papers which have been discussed in the literature survey discuss the comparison of the mean per joint positioning error (MPJPE), we have also obtained better results with our algorithm, that is, GAN-Poser, regarding the mean per joint positioning error. In this, the subjects utilized for the training purpose are 1, 5, 6, 7, 8 and for the testing purpose, custom real-time images of a person are used. The choice of the subjects used for training is taken so that our results could be comparable to the other state-of-the-art methods. MPJPE has two pars; in the first part, MPJPE is directly computed, and in the other, MPJPE is computed by applying the rigid alignment between the poses.

4.3 Dataset

For the comparison of the results, we have taken the two datasets that are the NTURGB-D and the Human3.6 M dataset. For the NTURGB-D dataset, we have taken the four actions namely walking, discussion, greeting and taking photo, for the comparison with other state-of-the-art methods, whereas for the Human3.6 M dataset, we have compared our result regarding the four actions namely walking, eating, smoking and discussion. The datasets have been explained as follows:

4.3.1 NTURGB-D dataset

To authenticate the model potential, we run multiple experiments on the largest human motion datasets: a Microsoft Kinect dataset NTURGB-D [27]. The poses in the NTURGB-D dataset are inferred from Kinect skeleton data and have objections in the direct application due to occlusions, or different posture behavior. However, even with noisy skeletons, our model generalizes well on this dataset. The NTURGB-D action recognition dataset consists of 56,880 actions, and each action comes with the corresponding RGB video, depth map sequence, 3D skeletal data and infrared video. We use only the 3D skeleton data. They contain the 3D locations of 25 major body joints at each frame, as defined by the Microsoft Kinect API. NTURGB-D has 60 action classes and 40 different subjects, and each action was recorded by three Kinects from different viewpoints. It is more difficult for training on the human pose angle, which has fewer degrees of freedom. We train directly on the joint positions and use the same pipeline for NTU-RGB-D to have a more generic model.

4.3.2 Human3.6 M dataset

It is the biggest benchmark dataset that contains images that are aligned in accordance with 2D and 3D correspondence. The Human3.6 M dataset [13] contains 3.6 million human action poses which have been collected from 4 digital cameras. The data have been organized into 15 motions, and the dataset consists of fewer clips as compared to the NTURGB-D dataset [27]. The motions contain the walking action along with various asymmetries such as walking with a hand in the pocket, walking with a bag on the shoulder and different sitting, laying and waiting poses. To use the same pipeline for both the dataset, we have split the clips in the Human3.6 M dataset into shorter segments and have used every other frame for the training.

4.4 Training

The training of the model has been performed on a GTX 1060 which contains 1200 CUDA cores and a 6 GB DDR5 memory. In the training loop, we have iterated over the network for n number of times on the critic network and once on the decoder as well as the discriminator network. Here, we have used various iteration values of n and have also tried to dynamically update the iteration based on the sampling and the geodesic losses. Since none of the methods have shown any improvement, we have taken the value of n as 10. The mean time for each of the epoch is 230 s. For training the Human3.6 M dataset, we have used the subjects 1, 5, 6, 7, 8 for training and the real-time 3D images of a person have been used for the testing. We have also trained our model on the NTURGB-D dataset similar to the Human3.6 M dataset. For bringing the stability in the discriminator training procedure, we have reduced the learning rate by half as compared to the generator network. We have used the ADAM optimizer for training all the networks and have set the learning rate to 5e−1 for the generator network. Hence, the learning rate for the discriminator network used is 2.5e−0.5.

5 Results and evaluation

For the Human3.6 M dataset [13], we used the subjects 1, 5, 6, 7, 8 for training our model and custom images to test the model and demonstrate the predicted pose as well as the performance of the model. Figure 2 clearly depicts the performance of our model on the sample images. Two real-life images of the person walking in a room is given as the input to our model, GAN-Poser. The model predicts the output for the given images which is given as a 3D joint pose. Similarly, Fig. 3 clearly depicts the performance of our model on the sample images. Two real-life images of the person eating in a room are given as the input to our model, GAN-Poser. The model predicts the output for the given images which is given as a 3D joint pose.

5.1 Zero velocity baseline

One of the most striking results for the good performance comparison of the baselines is the zero-velocity one [37].

They evidently perform better than state-of-the-art outcomes, highlighting the stringency of the discontinuities between training and prediction in previous work. By visualizing the performance of the baselines, it can be determined that deterministic losses are not appropriate to calculate motion forecasting with a long time frame. For the comparison of the result, we will use the NTURGB-D [27] dataset. Here, we have considered the four actions namely walking, discussion, greeting and taking photo for the comparison of the results with our model. The results have been compared to each category wise in the following tables along with the summary of results using methods such as ERD, LSTM-3LR and SRNN, as well as a zero-velocity baseline.

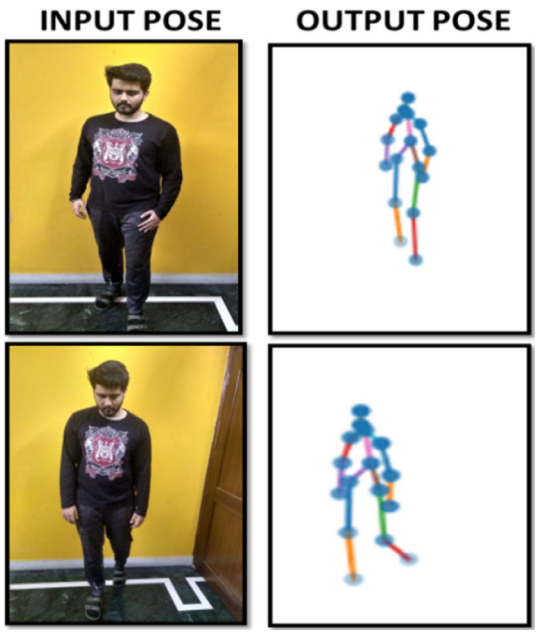

Fig. 2 Prediction of walking action pose

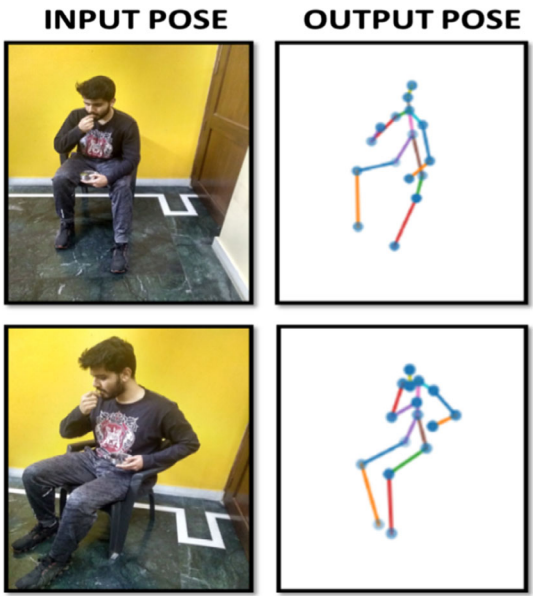

Fig. 3 Prediction of eating action pose

5.1.1 Walking

5.1.2 Discussion

5.1.3 Greeting

5.1.4 Taking photo

It is evident from Tables 1, 2, 3 and 4 that our model, GAN-Poser, is comparable to the state-of-the-art methods. Moreover, it can be seen from Table 1 that in the walking category, our model performs the best for 80 ms. We can also see from Table 2 that GAN-Poser surpasses all the state-of-the-art methods in short-term as well as the long-term human prediction that is, in the case of 80 ms, 400 ms and 1000 ms. Tables 3 and 4 also show that GAN-Poser gives the comparable results to the state-of-the-art methods on the NTURGB-D dataset.

Table 1 Comparison of performance for mean angle error for both short-term and long-term human prediction on the NTURGB-D dataset for the walking action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
ERD [8]	0.77	0.90	1.12	1.25	1.44
LSTM-3LR [8]	0.73	0.81	1.05	1.18	4.36
Res-GRU [24]	0.27	**0.47**	**0.68**	**0.76**	**1.06**
Zero-velocity [37]	0.39	0.68	0.99	1.15	1.32
GAN-Poser	**0.25**	0.66	0.82	1.13	1.77

The best performance has been highlighted in the bold

Table 2 Comparison of performance for mean angle error for both short-term and long-term human prediction on the NTURGB-D dataset for the discussion action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
ERD [8]	0.76	0.96	1.17	1.24	2.04
LSTM-3LR [8]	0.71	0.84	1.02	1.11	1.99
Res-GRU [24]	0.31	0.69	1.03	1.12	1.87
Zero-velocity [37]	0.31	**0.67**	**0.97**	1.04	1.96
GAN-Poser	**0.24**	0.88	1.01	**1.03**	**1.69**

The best performance has been highlighted in the bold

Table 3 Comparison of performance for mean angle error for both short-term and long-term human prediction on the NTURGB-D dataset for the greeting action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
ERD [8]	0.85	1.09	1.45	1.64	1.98
LSTM-3LR [8]	0.80	0.99	1.37	1.54	1.85
Res-GRU [24]	**0.52**	**0.86**	1.30	**1.47**	1.96
Zero-velocity [37]	0.54	0.89	1.30	1.49	**1.80**
GAN-Poser	0.53	0.88	**1.22**	1.56	1.91

The best performance has been highlighted in the bold

Table 4 Comparison of performance for mean angle error for both short-term and long-term human prediction on the NTURGB-D dataset for the taking photo action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
ERD [8]	0.70	0.78	0.97	1.09	1.39
LSTM-3LR [8]	0.63	0.64	0.86	0.98	1.30
Res-GRU [24]	0.29	0.58	0.90	1.04	1.47
Zero-velocity [37]	0.25	0.51	**0.79**	**0.92**	**1.27**
GAN-Poser	**0.21**	**0.30**	0.86	1.14	1.54

The best performance has been highlighted in the bold

5.2 Comparison with HP-GAN

HP-GAN [38] is a sequence-to-sequence model. Here, a similar approach is proposed as compared to our model. HP-GAN uses a combination of WGAN-GP and a custom loss function to predict the human motion pose. It includes learning from the previously predicted poses which can indicate that it is also using the recursive prediction strategy. Moreover, it also predicts the future sequence poses while carrying in the same sequence of input but using a different value of vector z. But the HP-GAN model fails to incorporate the long-term pose prediction which is extremely essential for forecasting motion poses from the actual human dynamics perspective. Our model, GAN-Poser, addresses this issue by taking into consideration the bidirectional nature of the GAN so that it can efficiently predict the long-term human poses. The HP-GAN uses a simple probabilistic approach for the human motion prediction which predicts multiple plausible future human poses from the same input. In our model, we have trained the discriminator to regress the extrinsic factor Θ, which is eventually used to generate a particular pose sequence.

5.3 Comparison with BiHMP-GAN

The BiHMP-GAN [39] uses a bidirectional framework for prediction of the human motion to avoid the mode collapse. This model also uses the random extrinsic factor Θ to generate multiple sequences of the human poses from a given starting pose sequence. In general, it is a critic model as compared to the HP-GAN, which was a probabilistic model. It also focuses on the long-term human motion prediction just like our model and performs better as compared to the HP-GAN. Our model, GAN-Poser, incorporates the geodesic loss, which is used for bringing the stability in the model. This has further provided us with better results in terms of short-term as well as long-term predictions as compared to the HP-GAN and BiHMP-GAN.

The following tables demonstrate the results for the different actions.

5.3.1 Walking

It is evident from Table 5 that our model, GAN-Poser, produces the results which are comparable with the state-of-the-art methods. For the short-term predictions, the mean angle error of our model is 0.71, 0.74, 84 and 1.23 for 8 ms, 160 ms, 320 ms and 400 ms, respectively. The best model for the short-term prediction is the BiHMP-GAN model. For the long-term predictions, our model performed better than the HP-GAN model by obtaining the mean angle error of 1.87 for 1000 ms.

5.3.2 Eating

As seen from Table 6, GAN-Poser produces better results than HP-GAN for the short-term predictions with the mean angle error of 0.80, 0.94, 0.97 and 1.06 for 80 ms, 160 ms, 320 ms and 400 ms, respectively. Our model achieved better results than the HP-GAN and the RRNN model for the long-term predictions by obtaining the mean angle error value of 1.25 for 1000 ms.

Table 5 Comparison of performance for mean angle error for both short-term and long-term human prediction on the Human3.6 M dataset for the walking action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
Conv-motion	**0.33**	0.54	0.68	0.73	0.92
RRNN [41]	**0.33**	0.56	0.78	0.85	1.14
HP-GAN [38]	0.95	1.17	1.69	1.79	2.47
BiHMP-GAN [39]	**0.33**	**0.52**	**0.64**	**0.69**	**0.88**
GAN-Poser	0.71	0.74	0.84	1.23	1.87

The best performance has been highlighted in the bold

Table 6 Comparison of performance for mean angle error for both short-term and long-term human prediction on the Human3.6 M dataset for the eating action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
Conv-Motion	0.22	0.36	0.58	**0.71**	1.24
RRNN [41]	0.26	0.43	0.66	0.81	1.34
HP-GAN [38]	1.28	1.47	1.70	1.82	2.51
BiHMP-GAN [39]	**0.21**	**0.33**	**0.55**	**0.71**	**1.20**
GAN-Poser	0.80	0.94	0.97	1.06	1.25

The best performance has been highlighted in the bold

Table 7 Comparison of performance for mean angle error for both short-term and long-term human prediction on the Human3.6 M dataset for the smoking action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
Conv-Motion	0.26	0.49	0.96	**0.92**	1.62
RRNN [41]	0.35	0.64	1.03	1.15	1.83
HP-GAN [38]	1.71	1.89	2.33	2.42	3.2
BiHMP-GAN [39]	0.26	0.49	0.91	0.88	1.12
GAN-Poser	**0.25**	**0.29**	**0.87**	1.02	**1.06**

The best performance has been highlighted in the bold

5.3.3 Smoking

As seen from Table 7, GAN-Poser produces better results than HP-GAN, RRNN, Conv-Motion and BiHMP-GAN for the short-term predictions with the mean angle error of 0.25, 0.29, 0.87 for 80 ms, 160 ms and 320 ms, respectively. For 400 ms, our model performed better than HP-GAN and RRNN with the error of 1.02. Our model surpassed all the state-of-the-art methods for the long-term predictions by obtaining the mean angle error value of 1.06 for 1000 ms.

Table 8 Comparison of performance for mean angle error for both short-term and long-term human prediction on the Human3.6 M dataset for the discussion action pose

Model	80 ms	160 ms	320 ms	400 ms	1000 ms
Conv-Motion	**0.32**	0.67	0.94	**1.01**	1.86
RRNN [41]	0.37	0.77	1.06	1.10	1.79
HP-GAN [38]	2.29	2.61	2.79	2.88	3.67
BiHMP-GAN [39]	**0.32**	**0.65**	**0.92**	9.98	**1.78**
GAN-Poser	0.33	0.70	1.11	1.96	2.83

The best performance has been highlighted in the bold

5.3.4 Discussion

It is evident from Table 8 that our model, GAN-Poser, produces the results which are comparable with the state-of-the-art methods. For the short-term predictions, the mean angle error of our model is 0.33, 0.70, 1.11 and 1.96 for 80 ms, 160 ms, 320 ms and 400 ms, respectively. For the long-term predictions, our model performed better than the HP-GAN model by obtaining the mean angle error of 2.83 for 1000 ms.

5.4 Comparison with RepNet

RepNet [40] or Reprojection Network focuses on addressing the issue that the reprojection constraint is sensitive to overfitting. But the RepNet model does not uses the actual predicted 3D joint poses for future predictions. Hence, to overcome this problem and improve the model, we have used the recursive prediction strategy which we have already discussed in Sect. 3.4. Moreover, for the comparison of results with the RepNet and other state-of-the-art methods, we have divided our training into two parts. In the first part, we have trained the model using a non-rigid alignment and obtained the results for the four actions, which are walking, eating, sleeping and discussion. The metrics used for the evaluation is the mean per joint positioning error (MPJPE), and the results are shown in the following table:

As it is evident from Table 9 that our model outperforms most of the state-of-the-art methods and it presents the results which are comparable to the results for RepNet. In the second part, we have trained the model using the poses having rigid alignment and the mean per joint positioning error (MPJPE) is compared with RepNet and other state-of-the-art methods (Table 10).

In the table, our model obtained the minimum error of 61.7 for the discussion action and maximum error of 88.0 for the smoking action. The table further shows that the results obtained by our model, if not the best, are comparable to the other state-of-the-art methods.

Table 9 Comparison of GAN-Poser with RepNet and other state-of-the-art methods for the mean per join positioning error (MPJPE) using the Human3.6 M dataset which is trained using the non-rigid alignment of poses

Model	Walking	Eating	Smoking	Discussion
LinKDE [13]	177.1	132.3	162.1	183.6
Tekin [42]	126.3	88.8	118.4	147.2
Zhou [43]	114.2	87.1	107.4	109.3
Du [44]	137.4	104.9	120.0	112.7
Park [45]	131.9	90.0	105.8	116.2
Martinez [46]	**50.9**	**62.9**	**69.1**	**60.8**
RepNet [40]	72.6	82.7	88.0	85.2
GAN-Poser	87.5	90.3	89.9	86.1

The best performance has been highlighted in the bold

Table 10 Comparison of GAN-Poser with RepNet and other state-of-the-art methods for the mean per join positioning error (MPJPE) using the Human3.6 M dataset which is trained using the rigid alignment of poses

Model	Walking	Eating	Smoking	Discussion
Akther [47]	198.6	161.8	177.8	177.6
Ramakrishna [48]	174.8	141.6	160.4	149.3
Zhou [23]	110.41	87.91	106.0	95.8
Bogo [49]	79.7	67.8	83.4	60.2
Martinez [46]	**35.9**	**44.4**	**54.0**	**52.0**
RepNet [40]	63.2	59.6	66.6	58.3
GAN-Poser	69.1	66.8	88.0	61.7

The best performance has been highlighted in the bold

6 Conclusion and future work

The paper concludes a novel GAN model, GAN-Poser, to improve predictions of motion sequences from a global outlook. A discriminator is projected to model the sequence-level dependability of the predicted sequences. We have used the bidirectional GAN with geodesic loss along with the recursive prediction strategy to reduce the over fitting and took into consideration the stochasticity for the prediction of future pose sequence. However, using the proposed method and all the modifications in the model that we have implemented, there still no silver bullet to confirm if the training has converged. Even if that works, the training can even diverge after it is already converged in the training loop. Therefore, the future work lies in looking for more stable converging methods. Another research work is considering the semantic and gap of the z vector. If we can evaluate the reverse mapping criteria for sequences of z values, we can utilize it further for classification. In the future, the work can also be extended looking for multiple subject detection and orientation in space.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Shamsolmoali P, Zareapoor M, Zhou H, Yang J (2020) AMIL: Adversarial Multi-instance Learning for Human Pose Estimation. ACM Trans Multimedia Comput Commun Appl (TOMM) 16(1s):1–23
2. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. CoRR arXiv:1701.07875
3. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: Human behavior understanding—2nd international workshop, HBU 2011, Amsterdam, The Netherlands, 16, 2011. Proceedings, pp 29–39
4. Bütepage J, Black MJ, Kragic D, Kjellström H (2017) Deep representation learning for human motion prediction and classification. CoRR arXiv:1702.07486
5. Chen B, Wang W, Wang J, Chen X (2017) Video imagination from a single image with transformation generation. CoRR arXiv:1706.04124
6. Chung J, Gülçehre Ç, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR arXiv:1412.3555
7. Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Darrell T, Saenko K (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, pp 2625–2634
8. Fragkiadaki K, Levine S, Felsen P, Malik J (2015) Recurrent network models for human dynamics. In: 2015 IEEE international conference on computer vision, ICCV 2015, Santiago, Chile, December 7–13, 2015, pp 4346–4354
9. Graves A (2013) Generating sequences with recurrent neural networks. CoRR arXiv:1308.0850
10. Pöhlmann STL, Harkness EF, Taylor CJ, Astley SM (2016) Evaluation of Kinect 3D sensor for healthcare imaging. J Med Biol Eng 36:857–870
11. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
12. Butepage J, Black MJ, Kragic D, Kjellström H (2017) Deep representation learning for human motion prediction and classification. CoRR arXiv:1702.07486
13. Ionescu C, Papava D, Olar V, Sminchisescu C (2014) Human3.6 m: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans Pattern Anal Mach Intell 36(7):1325–1339
14. Jain A, Zamir AR, Savarese S, Saxena A (2016) Structuralrnn: deep learning on spatio-temporal graphs. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 5308–5317
15. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27:2672–2680
16. Denton EL, Chintala S, Fergus R et al (2015) Deep generative image models using a Laplacian pyramid of adversarial networks. In: NIPS
17. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV
18. Vondrick C, Pirsiavash H, Torralba A (2016) Generating videos with scene dynamics. In: NIPS
19. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. In: ICML
20. Shamsolmoali P, Zareapoor M, Wang R, Jain DK, Yang J (2019) G-GANISR: gradual generative adversarial network for image super resolution. Neurocomputing 366:140–153
21. Zareapoor M, Zhou H, Yang J (2019) Perceptual image quality using dual generative adversarial network. Neural Comput Appl. https://doi.org/10.1007/s00521-019-04239-0
22. Ng JY, Hausknecht M, Vijayanarasimhan S, Oriol Vinyals RM, Toderici G (2016) Beyond short snippets: deep networks for video classification. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR, pp 4594–4602
23. Zhou X, Zhu M, Leonardos S, Daniilidis K (2017) Sparse representation for 3D shape estimation: a convex relaxation approach. IEEE Trans Pattern Anal Mach Intell 39(8):1648–1661
24. Martinez J, Black MJ, Romero J (2017) On human motion prediction using recurrent neural networks. In: CVPR
25. Ionescu C, Li F, Sminchisescu C (2011) Latent structured models for human pose estimation. In: International conference on computer vision
26. Bouhlel N, Dziri A (2019) Kullback–Leibler divergence between multivariate generalized gaussian distributions. IEEE Signal Process Lett 26(7):1021–1025
27. Daskalakis C, Papadimitriou CH (July 2009) On a network generalization of the minmax theorem. In: International colloquium on automata, languages, and programming. Springer, Berlin, pp 423–434
28. Zhang Z, Liu S, Li M, Zhou M, Chen E (Oct 2018) Bidirectional generative adversarial networks for neural machine translation. In: Proceedings of the 22nd conference on computational natural language learning, pp 190–199
29. Berglund M, Raiko T, Honkala M, Kärkkäinen L, Vetek A, Karhunen JT (2015) Bidirectional recurrent neural networks as generative models. In: Advances in neural information processing systems, pp 856–864
30. Jaiswal A, AbdAlmageed W, Wu Y, Natarajan P (Dec 2018) Bidirectional conditional generative adversarial networks. In: Asian conference on computer vision. Springer, Cham, pp 216–232
31. Moore JB, Weiss H (1979) Recursive prediction error methods for adaptive estimation. IEEE Trans Syst Man Cybern 9(4):197–205
32. Wigren T (2004) Recursive prediction error identification of nonlinear state space models. Technical Reports from the Department of Information Technology, 4
33. Bengio Y (2009) Learning deep architectures for AI. Found Trends® Mach Learn 2(1):1–127
34. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
35. Ollivier Y (2015) Riemannian metrics for neural networks I: feedforward networks. Inf Inference J IMA 4(2):108–153
36. Shahroudy A, Liu J, Ng T-T, Wang G (June 2016) Ntu rgb + d: a large scale dataset for 3D human activity analysis. In: The IEEE conference on computer vision and pattern recognition (CVPR)
37. Tang Y, Ma L, Liu W, Zheng W (2018) Long-term human motion prediction by modeling motion context and enhancing motion dynamic. Preprint arXiv:1805.02513
38. Barsoum E, Kender J, Liu Z (2018) HP-GAN: probabilistic 3D human motion prediction via GAN. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 1418–1427
39. Kundu JN, Gor M, Babu RV (2019, July) Bihmp-gan: bidirectional 3D human motion prediction Gan. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 8553–8560
40. Wandt B, Rosenhahn B (2019) RepNet: weakly supervised training of an adversarial reprojection network for 3D human pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7782–7791

41. Bitzer S, Kiebel SJ (2012) Recognizing recurrent neural networks (rRNN): Bayesian inference for recurrent neural networks. Biol Cybern 106(4–5):201–217
42. Tekin B, Rozantsev A, Lepetit V, Fua P (2016) Direct prediction of 3D body poses from motion compensated sequences. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 991–1000
43. Zhou X, Zhu M, Leonardos S, Derpanis KG, Daniilidis K (June 2016) Sparseness meets deepness: 3D human pose estimation from monocular video. In: The IEEE conference on computer vision and pattern recognition (CVPR)
44. Du Y, Wong Y, Liu Y, Han F, Gui Y, Wang Z, Kankanhalli M, Geng W (2016) Marker-less 3D human motion capture with monocular image sequence and height-maps. In: European conference on computer vision, pp 20–36. Springer, Berlin
45. Park S, Hwang J, Kwak N (2016) 3D human pose estimation using convolutional neural networks with 2D pose information. In: Computer vision—ECCV 2016 workshops—Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, proceedings, Part III, pp 156–169
46. Martinez J, Hossain R, Romero J, Little JJ (2017) A simple yet effective baseline for 3D human pose estimation. In: ICCV
47. Akhter I, Black MJ (June 2015) Pose-conditioned joint angle limits for 3D human pose reconstruction. In: IEEE conference on computer vision and pattern recognition (CVPR 2015), pp 1446–1455
48. Ramakrishna V, Kanade T, Sheikh YA (Oct 2012) Reconstructing 3D human pose from 2D image landmarks. In European conference on computer vision (ECCV)
49. Bogo F, Kanazawa A, Lassner C, Gehler P, Romero J, Black J (Oct 2016) Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In: Computer vision—ECCV 2016, lecture notes in computer science. Springer, London