Human Motion Prediction via Spatio-Temporal Inpainting

A. Hernandez Ruiz[1] J. Gall[2] F. Moreno-Noguer[1]
[1]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
[2] Computer Vision Group, University of Bonn, Germany

Abstract

We propose a Generative Adversarial Network (GAN) to forecast 3D human motion given a sequence of past 3D skeleton poses. While recent GANs have shown promising results, they can only forecast plausible motion over relatively short periods of time (few hundred milliseconds) and typically ignore the absolute position of the skeleton w.r.t. the camera. Our scheme provides long term predictions (two seconds or more) for both the body pose and its absolute position. Our approach builds upon three main contributions. First, we represent the data using a spatio-temporal tensor of 3D skeleton coordinates which allows formulating the prediction problem as an inpainting one, for which GANs work particularly well. Secondly, we design an architecture to learn the joint distribution of body poses and global motion, capable to hypothesize large chunks of the input 3D tensor with missing data. And finally, we argue that the L2 metric, considered so far by most approaches, fails to capture the actual distribution of long-term human motion. We propose two alternative metrics, based on the distribution of frequencies, that are able to capture more realistic motion patterns. Extensive experiments demonstrate our approach to significantly improve the state of the art, while also handling situations in which past observations are corrupted by occlusions, noise and missing frames.

1. Introduction

Recent advances in motion capture technologies, combined with large scale datasets such as Human3.6M [16], have spurred the interest for new deep learning algorithms able to forecast 3D human motion from past skeleton data. State-of-the-art approaches formulate the problem as a sequence generation task, and solve it using Recurrent Neural Networks (RNNs) [8, 17], sequence-to-sequence models [24] or encoder-decoder predictors [5, 11]. While promising results, these works suffer from three fundamental limitations. First, they address a simplified version of the problem in which global body positioning is disregarded, either by parameterizing 3D body joints using position agnostic angles [8, 17, 24] or body centered coordinates [5]. Second, current methods require additional supervision in terms of action labels during training and inference, which limits their generalization capabilities. And third, most approaches aim to minimize the L2 distance between the ground truth and generated motions. The L2 distance, however, is known to be an inaccurate metric, specially to compare long motion sequences. In particular, the use of this metric to train a deep network favors motion predictions that converge to a static mean pose. Even though this issue has been raised in [11, 17] and is partially solved during training using other metrics (e.g. geodesic loss), the L2 distance is still being used as a common practice when benchmarking different methodologies. To our understanding, this practice compromises the progress in this field.

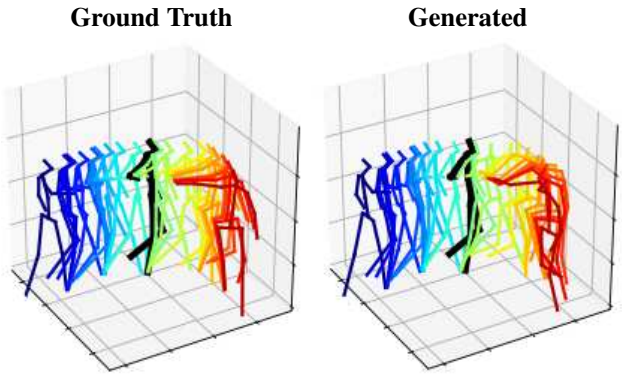

Figure 1. **Example result.** Our approach is the first in generating full body pose, including skeleton motion and absolute position in space. The predicted sequence starts from the skeleton marked in black. Note that the generated motion is somewhat different but semantically indistinguishable from the ground truth.

In this paper we tackle all three issues. Specifically, we design a novel GAN architecture that is conditioned on past observations and is able to jointly forecast non-rigid body pose and its absolute position in space. For this purpose we represent the observed skeleton poses (expressed in the camera reference frame) using a spatio-temporal tensor and formulate the prediction problem as an inpainting task, in which a part of the spatio-temporal volume needs

to be regressed. A GAN architecture consisting of a fully convolutional generator specially designed to preserve the temporal coherence, and three independent discriminators that enforce anthropomorphism of the generated skeleton and its motion, makes it possible to render highly realistic long-term predictions (of 2 seconds or more). Interestingly, the L2 loss is only enforced over the reconstructed past observations and not over the hypothesized future predictions. This way, the generation of future frames is fully controlled by the discriminators. In fact, our model does not require ground truth annotations of the generated frames nor explicit information about the action being performed.

We also introduce a novel metric for estimating the similarity between the generated and the ground truth sequences. Instead of seeking to get a perfect match for all joints across all frames (as done when using the L2 distance), the metric we propose aims to estimate the similarity between distributions over the human motion manifold.

In the experimental section we show that our approach, besides yielding full body pose, orientation and position, is also robust to challenging artifacts including missing frames and occluded joints on the past skeleton observations. Fig. 1 shows an example result of our approach.

2. Related Work

Deep Learning for motion prediction. Most recent deep learning approaches build upon the problem formulation proposed by [31] in which input motion sequences are represented by 3D body joint angles in a kinematic tree. Motivated by their success in machine translation problems [6, 18, 30], RNNs are then used to predict motion sequences of body joint angles. For instance, Fragkiadaki *et al*. [8] introduce an Encoder-Recurrent-Decoder (ERD) in combination with a Long Short-Term Memory (LSTM) for this purpose. Jain *et al*. [17] introduced structural RNNs, an approach that exploits the structural hierarchy of the human body parts. Martinez *et al*. [24] develop a sequence-to-sequence architecture with a residual connection that incorporates action class information via one-hot vectors. While well-suited for the particular motion they are trained for, these approaches do not generalize to other actions. More importantly, these models are only effective at predicting in the short- and mid-range time horizons, and are often surpassed by a simple zero velocity baseline model. This is in part due to the use of the L2 metric both for training and evaluation. More recent approaches have different strategies to address this problem.

Li *et al*. [22], propose a model with an auto-regressive CNN generator, and combine the L2 loss with an adversarial loss. Gui *et al*. [11] fully eliminate the L2 loss at training, and utilize an ERD model with a combination of an adversarial and geodesic losses. These works, however, still perform evaluation according to the L2 metric, which fails to capture the semantics of the motion, specially for long term predictions. Furthermore, since motion is parameterized in terms of the joint angles, the rotation and translation of the body in space is not estimated.

Sequence completion and image inpainting. Completing missing data within a sequence has been traditionally addressed using low-rank matrix factorization [1, 32]. Deep Learning approaches have also been used for this purpose *e.g*., through RNNs [21, 23]. These works, however, are not designed for future prediction.

Image inpainting is a very related problem. In the deep learning era, Denoising AEs [3] and Variational AEs [20] have become prevalent frameworks for denoising and completing missing data and image inpainting. These baselines, however, cannot handle large missing portions of structured data. State of the art has been significantly pushed by GANs conditioned with partial or corrupted images [27, 33, 34]. As we shall see, our approach draws inspiration on this idea.

Metrics for evaluating human motion prediction. The fact that L2 is not appropriate for measuring similarity between human motion sequences has been recently discussed and addressed in different works. Coskun *et al*. [7] use deep metric learning and a contrastive loss to learn the metric directly from the data. This is arguably the best way to compare the motion sequences semantically. Nonetheless, the problem with this approach is that once the metric is trained, it is hard to apply to different models, since the metric is trained with a specific setup. An alternative for sequences that does not require training, would be to use frequency based metrics. In [10] a metric based in power spectrum is proposed. This metric shows interesting properties, and seems suitable to compare actions with periodic motion such as walking. The main drawback of this approach is that it compares sequence by sequence, which in our view is undesirable. We, instead, would like to compare distributions of sequences.

For image generation, there are recent works that propose measuring the fitness of the models based on properties of the distribution of the generated data. The Inception Score [29] measures the entropy of the label outputs of the inception network on the generated images. The Frechet Inception Distance [14] (FID) propose instead to fit two multivariate Gaussians to the activations of the inception network, when the real and generated samples respectively. Then the FID is obtained by measuring the distance between the gaussian models. Following these works, we propose new metrics based on the distribution of the frequencies of the generated samples. These metrics have the advantage of being easy to implement and replicate, and they measure the general fitness of a model by taking into account a distribution of sequences.

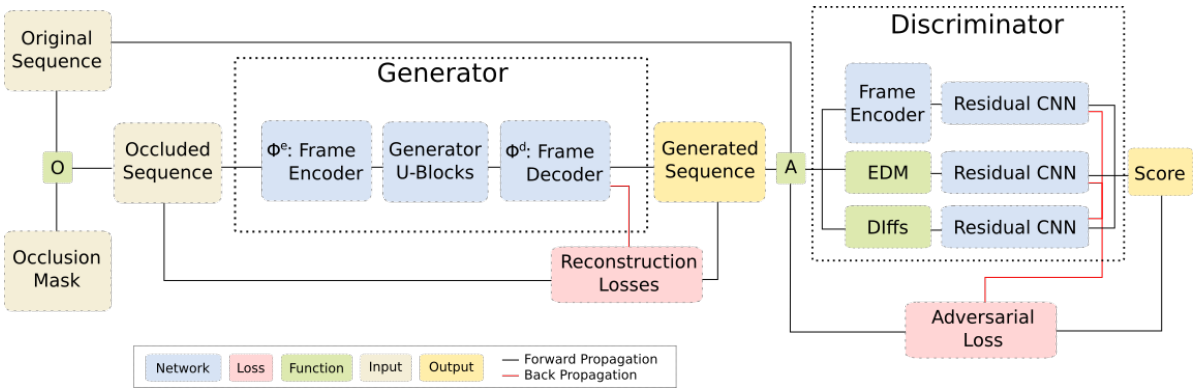

Figure 2. Overview of our architecture. An input masked sequence of 3D joint coordinates is fed into a fully convolutional and time preserving generator. The output sequence is controlled by a number of geometric constraints, including losses applied to the generator output and adversarial losses of three independent discriminators.

3. Problem Formulation

We represent human pose with a J-joint skeleton, where each joint consists of its 3D Cartesian coordinates expressed in the camera reference frame. Rotation and translation transformations are inherently encoded in such coordinates. A motion sequence is a concatenation of F skeletons, which we shall represent by a tensor $\mathbf{S} \in \mathbb{R}^{\mathrm{F}\times\mathrm{J}\times 3}$. Let us define an occlusion mask as a binary matrix $\mathbf{M} \in \mathbb{B}^{\mathrm{F}\times\mathrm{J}\times 3}$, which determines the part of the sequence that is not observed and is applied onto the sequence by performing the element-wise dot product $\mathbf{S} \circ \mathbf{M} \equiv \mathbf{S}^{\mathrm{m}}$. Our goal is then to estimate the 3D coordinates of the masked joints. Note that depending on the pattern used to generate the occlusion mask $\mathbf{M}$, we can define different sub-problems. For instance, by masking the last frames of the sequence we can represent a forecasting problem; if instead we mask specific intermediate joints, then we represent random joints occlusions, structured occlusions or missing frames. Our model can tackle any combination of these sub-problems.

4. Model

4.1. STMI-GAN architecture

Fig. 2 shows an overview of the GAN we propose, in which we pose the human motion prediction problem as an inpainting task in the spatio-temporal domain. We denote our network as STMI-GAN (Spatio-Temporal Motion Inpainting GAN). We next describe its main components.

Generator. The rationale for the design holds in that convolutional GANs have been successful in image inpainting problems, which is similar to ours. A masked human motion sequence $\mathbf{S}^{\mathrm{m}}$, however, cannot be directly processed by a convolutional network because the dimension corresponding to the joints (J) does not have a spatial meaning, in contrast to the temporal (F) and Cartesian coordinates dimensions. That is, neighboring joints along this dimension do not correspond to neighboring joints in 3D space[1]. To alleviate this lack of spatial continuity problem, the generator is placed in-between a frame autoencoder, namely a frame encoder Φ^e and a frame decoder Φ^d, which are symmetrical networks. The frame encoder, operates over the J-dimension and projects each frame $\mathbf{S}^{\mathrm{m}}_{i::} \in \mathbb{R}^{\mathrm{J}\times 3}$ of the sequence to a one-dimensional vector $\mathbf{S}^{\mathrm{e}}_{i::} = \Phi^e(\mathbf{S}^{\mathrm{m}}_{i::}) \in \mathbb{R}^{\mathrm{H}\times 1}$, where H is the dimension of the space for the pose embedding[2]. To project each frame, the encoder does not use information from neighboring frames, being thus time invariant. We denote the encoded sequence as $\mathbf{S}^e \in \mathbb{R}^{\mathrm{F}\times\mathrm{H}\times 1}$.

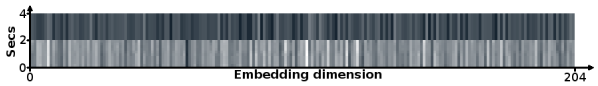

Figure 3. **Motion embedding.** A motion sequence of the H3.6M dataset passed through the frame encoder. The sequence is occluded from the half onwards, with the goal of motion prediction.

As we can observe in Fig.3, the frame encoder learns to represent the sequence as a 2D matrix, in an arbitrary space. Although the learnt space has no clear interpretation, we can observe from the sample that it retains certain properties, such as the temporal ordering, and a constant "zero" value for the occluded frames. This encoded sequence is then passed through a series of generator blocks Φ^g, which produces a new sequence $\mathbf{S}^g \in \mathbb{R}^{\mathrm{F}\times\mathrm{H}\times 1}$ in the embedded space. The blocks of the generators are CNNs that process the sequence in both temporal and spatial dimensions. Further details are explained in Section 6. Finally the decoder network Φ^d maps back the the transformed sequence $\mathbf{S}^g$ to the output sequence in the original shape $\mathbf{S}^{out} \in \mathbb{R}^{\mathrm{F}\times\mathrm{J}\times 3}$.

Discriminator. To capture the complexity of the human motion distribution we split the discriminator into three

[1] For instance, the joint #0 is normally the hip, and its neighbors in the body graph are the joints #1 (left hip), #5 (right hip) and #9 (spine).

[2] $\mathbf{S}_{i::}$ denotes the i-th element of $\mathbf{S}$ along the first axis.

branches, capturing different aspects of the generated sequence. Each discriminator is a Residual CNN classifier that serves as a feature extractor. These features are linearly combined to obtain a probability of the sequence of being real. We next describe the main blocks of our model. Details of the underlying architectures are detailed later in Sect. 6.

Base discriminator. The same architecture of the frame encoder Φ^e, with independent parameters, is used to process the generated sequence $\mathbf{S}^{out}$. The reason to reuse such architecture is that we want a discriminator to be applied directly on the non Euclidean representation used by the CNN blocks of the generator, in order to boost its performance.

EDM discriminator. We introduce a geometric discriminator that evaluates the anthropomorphism of the generated sequence $\mathbf{S}^{out}$ via the analysis of its Euclidean Distance Matrix, computed as $\text{EDM}(\mathbf{S}^{out}) \equiv \mathbf{D} \in \mathbb{R}^{\text{J}\times\text{J}}$, where $\mathbf{D}_{ij}$ is the Euclidean distance between joints i and j of $\mathbf{S}^{out}$. This is a rotation and translation invariant representation [13, 26], allowing to focus the attention of the discriminator into the shape of the skeleton.

Motion discriminator. the Base discriminator sees the sequences as absolute coordinates of joints in the space, and the EDM discriminator sees them as relative coordinates w.r.t the other joints. But these discriminators are missing the joint correlations between the absolute motion and their relative (articulated) counterpart. Thus, we consider a third discriminator that operates over the concatenation of both, the temporal differences of absolute coordinates $\|\mathbf{S}^{out}(t) - \mathbf{S}^{out}(t-1)\|_1$ and the temporal differences of EDM representations $\|\text{EDM}(\mathbf{S}^{out}(t)) - \text{EDM}(\mathbf{S}^{out}(t-1))\|_1$, where $\mathbf{S}^{out}(t)$ indicates generated the sequence at time t.

4.2. Losses

To train our network we use two main losses: 1) The reconstruction losses, that encourage the generator to preserve the information from the visible part of the sequence; 2) The GAN loss, which guides the generator to inpaint the sequences by learning and reproducing the motion in the dataset. For all the following formulae, let $\mathbf{S}$ be the input motion sequence, $\mathbf{M}$ the occlusion mask, $\mathbf{S}^{out}$ the generated sequence, F number of frames and J number of joints.

Reconstruction Loss. Our default reconstruction loss computes the L2 norm over the generated sequence w.r.t. the visible portion of the ground truth.

$$\mathcal{L}_{rec} = \|(\mathbf{S}^{out} \circ \mathbf{M}) - (\mathbf{S} \circ \mathbf{M})\|_2 \tag{1}$$

This loss is only applied over the visible part of the original and generated sequences. By doing this, we penalize deviations from the visible part of the sequences, while avoiding to penalize the different possible completions of the sequence.

Limb Distances Loss. [13] showed that most common actions can be recognized from just the relative distance between the extremities, *i.e*. hands, feet and head. We therefore add a loss that explicitly enforces the correct distance between these semantically important joints. Since this loss looks at the relative distance, instead of the absolute position, it provides different gradients to the reconstruction loss, and encourages the network to learn a more precise location for the limbs. Formally, if we denote by $\mathcal{E} = \{i, j\}$ the set of limb pairs , the loss $\mathcal{L}_{limb}$ is computed as:

$$\sum_{f=1}^{F} \sum_{\{i,j\}\in\mathcal{E}} \|\|\mathbf{S}^{m}_{fi:} - \mathbf{S}^{m}_{fj:}\|_2 - \|\mathbf{S}^{m,out}_{fi:} - \mathbf{S}^{m,out}_{fj:}\|_2\|_2 \tag{2}$$

where $\mathbf{S}^{m,out} = \mathbf{S}^{out} \circ \mathbf{M}$, denoting again that this loss is only computed over the visible part of the original sequence.

Bone Length Loss. We also enforce constant bone length of the whole generated sequence. Its main goal is to discourage the generator to explore solutions where the skeleton is not well formed. If we denote by $\overline{\mathcal{B}} = \{\bar{l}_1, \ldots, \bar{l}_B\}$ the mean length of the B body bones computed over the visible part of the sequence, and by $\mathcal{B}_f = \{l_{f1}, \ldots, l_{fB}\}$ the length of the bones at frame f, this loss is computed as

$$\mathcal{L}_{bone} = \sum_{f=1}^{F} \sum_{b=1}^{B} \|\bar{l}_b - l_{fb}\|_2 \tag{3}$$

Regularized Adversarial Loss. Our adversarial loss is based on the original GAN loss [9], with the R1 regularization described in [25]. Let G_θ be the generator network, parameterized by the variable θ, D_ψ be the discriminator network, parameterized by the variable ψ, and $\mathbb{P}_o$ the distribution of input motion sequences. We can then write the Discriminator Loss as:

$$\begin{aligned}\mathcal{L}_D =& \mathbb{E}_{\mathbf{S}^{out}\sim\mathbb{P}_o}\left[\log(1 - D_\psi(G_\theta(\mathbf{S}\circ\mathbf{M})))\right] \\ &+ \mathbb{E}_{\mathbf{S}\sim\mathbb{P}_o}\left[\log(D_\psi(x))\right] + \frac{\gamma}{2}\mathbb{E}_{\mathbf{S}\sim\mathbb{P}_o}\left[\|\nabla D_\psi(x)\|^2\right]\end{aligned} \tag{4}$$

The Generator Loss is as follows:

$$\mathcal{L}_G(\theta, \psi) = \mathbb{E}_{\mathbf{S}^{out}\sim\mathbb{P}_o}\left[\log(D_\psi(G_\theta(\mathbf{S}\circ\mathbf{M})))\right] \tag{5}$$

Full Loss. The full loss $\mathcal{L}$ consists of a linear combination of all previous partial losses:

$$\mathcal{L} = \lambda_r\mathcal{L}_{rec} + \lambda_l\mathcal{L}_{limb} + \lambda_b\mathcal{L}_{bone} + \lambda_D\mathcal{L}_D + \lambda_G\mathcal{L}_G \tag{6}$$

where λ_r, λ_l, λ_b, λ_D and λ_G are the hyper-parameters that control the relative importance of every loss term. Finally, we can define the following minimax problem:

$$G^\star = \arg\min_G \max_{D\in\mathcal{D}} \mathcal{L}\ , \tag{7}$$

where $G^\star$ draws samples from the data distribution.

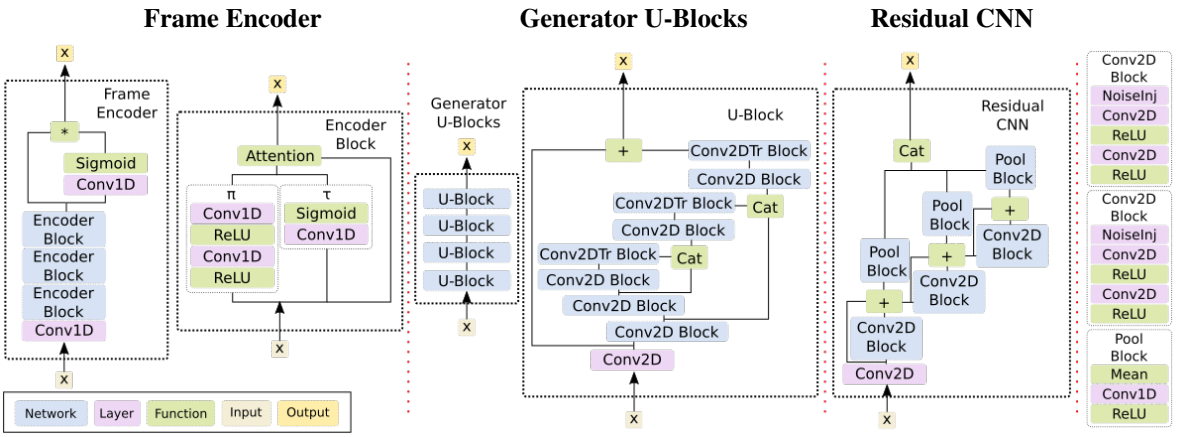

Figure 4. **Details of the architecture.** From left to right: Frame Encoder; Generator U-Blocks; Residual CNN. In each case we plot the general view of the block (left) and the fine detail of the structural elements (right). The Attention function is defined as: $att(x, \pi, \tau) = \pi\tau + x(1 - \tau)$. The Conv2DTr denotes Convolution 2D Transpose, also called deconvolution.

5. Metrics for motion prediction

Our goal then is to analyze the distribution generated by our model, for which we propose metrics that are analogue to the Inception Score [29] and the Frechet Inception Distance [14]. With this in mind we propose the following metrics:

PSEnt measures the entropy in the power spectrum of a dataset. This metric can give us a rough estimate of the fitness of the model. First we compute the power spectrum of the dataset independently per each joint and axis. Each joint-axis combination is considered a distinct feature of a sequence. Formally, the power spectrum of a feature is computed as: $PS(s_f) = \|FFT(s_f)\|^2$. We can then compute the Power Spectrum Entropy over a dataset:

$$PSEnt(D) = \frac{1}{S}\sum_{s\in D}\frac{1}{F}\sum_{f=1}^{F} - \sum_{e=1}^{E}\|PS(s_f)\| * log(\|PS(s_f)\|) \tag{8}$$

where D is a dataset, s is a sequence, f is a feature, and e is frequency.

A common characteristic of the generative models trained with the L2 loss is that they have the tendency to regress to the mean, lowering the entropy of the generated sequences. An entropy value lower than the expected is a telltale sign of a biased model, whereas a higher entropy value points to a rather noisy and maybe inaccurate or unstable model.

PSKL measures the distance (in terms of the KL divergence) between the ground truth and generated datasets:

$$PSKL(C, D) = \sum_{e=1}^{E}\|PS(C)\| * log(\frac{\|PS(C)\|}{\|PS(D)\|}) \tag{9}$$

where C and D are datasets, s is a sequence, f is a feature, and e is frequency. The KL divergence is asymmetric, so we compute both directions PSKL(GT, Gen) and PSKL(Gen, GT) to have the complete picture of the divergence. If both directions are roughly equal, it would mean that the datasets are different but equally complex. On the other hand if the divergences are considerably different, it would mean that one of the datasets has a biased distribution.

L2 based metrics. We also measure the distance between the ground truth sequence s_{gt} and the generated sequence s_{gen}, considering each joint (j) as an independent feature vector.

$$L2(s_{gt}, s_{gen}) = \frac{1}{J}\sum_{f=1}^{J}\|s_{gt} - s_{gen}\|_2 \tag{10}$$

In [8, 24] s is represented in Euler angles, but in our work s is in coordinates, making it readable in millimeters. The mean is used to obtain a measure for the complete dataset.

6. Implementation Details

We next describe the blocks of our architecture. Code Available at: https://github.com/magnux/MotionGAN

Frame Autoencoder. The Frame Encoder (Fig. 4-left) is a fully connected network with identical sequential blocks. Each block contains two consecutive fully connected layers, and an attention mechanism [2] at the end. The fully connected layers can also be seen as 1D convolutions with kernel size 1 along the time dimension. The attention is performed by applying a mask over the output of the block. The mask is a linear transformation of the input to the block, followed by a sigmoid activation. After the blocks, a final linear transformation and an attention are applied. The architecture is similar to a VAE [20], but without the Gaussian constrains over the output of the encoder. The decoder network is a symmetrical network to the encoder, with the same number of blocks and layers, but independent parameters.

Generator U-Blocks. The goal of the generator is to produce an output that should be indistinguishable from an unmasked input, while preserving the shape of the sequence. To accomplish this, we use U-blocks [28] (see Fig. 4-center) with convolutions that halve the spatial resolution of the input in each layer, until it reaches a small representation. Then a transposed convolution is used to double the resolution until it reaches again the same dimensions as the input. A key component in this architecture are the skip connections that connect the output of the convolutional layers to the input of the deconvolutional layers. We can think of this architecture as an iterative refinement, in which the output of a block is refined by the next block to produce a better final output. Following [19], we also incorporate a noise injection layer into our convolutional blocks which makes the model prediction non-deterministic and enriches it.

Residual CNN. We designed the architecture of our discriminator inspired on ResNet [12] and DenseNet [15]. Our network (Fig. 2) branches in three discriminators, and each discriminator has a classifier, with the same architecture but separate parameters. Their architecture (Fig. 4-right) consists of several consecutive blocks with two convolutional layers and additive residual connections, similar to ResNet. The outputs of each block are also transformed and then concatenated into a tensor. The final output is the concatenation of the outputs of all blocks. This output is finally passed onto a fully connected network that assigns a score.

Spatial Alignment. Since we are working over an absolute coordinate system, the sequences have a wide range of values (from mm to m). To improve the robustness of the generator we subtract the position of the hip joint in the first frame to all the joints in the sequence. Then the skeleton is rotated to always face in the same direction. The alignment is performed by a custom layer in the network before the frame encoder, and is reversed just after the frame decoder.

7. Experiments

Datasets. In the experimental section we mainly use the Human3.6M [16] dataset. We follow the same split used in [8, 24].

7.1. Motion Prediction

In this section we compare our approach to [24], one of the baseline works in the state of the art. From this work, we are using the residual supervised model, which is a sequence-to-sequence model with residual connections and uses the labels as part of the inputs. Since our model is based on Cartesian coordinates, we compute the joint angle representation equivalent to that used by the Residual supervised (Res.sup.) [24] model. This transformation allows us for a consistent comparison between the two.

We next run an ablation study, using always the same generator network, but training it with different discriminator networks. As we argued in previous sections, we aim to capture the distribution of the Ground Truth (GT) data, and each component in the architecture was designed with this purpose. Our hypothesis is that an adversarial loss is better than L2 and geometric losses to train a generative network. Even more, we argue that the complexity of the discriminator network should be correlated with a better result in the generated sequences.

Model	PSEnt	PSKL(GT,Gen)	PSKL(Gen,GT)
0 to 1 second			
Org.Data. (Val vs Train)	0.67990	0.00590	0.00572
Res.sup. [24]	0.37492	0.03293	0.04524
NoGAN	0.44363	0.03729	0.05040
Base disc	0.73626	0.01198	0.01149
+EDM disc	0.57045	0.01801	0.02131
+Motion disc	0.72617	0.01220	0.01141
STMI-GAN	**0.68099**	**0.01090**	**0.01125**
1 to 2 seconds			
Org.Data. (Val vs Train)	0.67749	0.00628	0.00611
Res.sup. [24]	0.20975	0.10188	0.17004
NoGAN	0.27969	0.07989	0.13743
Base disc	0.60450	0.01559	0.01766
+EDM disc	0.49198	0.02546	0.03315
+Motion disc	0.72963	0.01223	0.01129
STMI-GAN	**0.68328**	**0.01041**	**0.01010**
2 to 3 seconds			
Org.Data. (Val vs Train)	0.67391	0.00640	0.00620
Res.sup. [24]	0.12752	0.17402	0.33566
NoGAN	0.34717	0.06099	0.09562
Base disc	0.60804	0.01396	0.01611
+EDM disc	0.45627	0.03368	0.04596
+Motion disc	0.72368	0.01312	0.01201
STMI-GAN	**0.71778**	**0.01306**	**0.01213**
3 to 4 seconds			
Org.Data. (Val vs Train)	0.67891	0.00590	0.00566
Res.sup. [24]	0.09333	0.18692	0.37605
NoGAN	0.26750	0.08672	0.15567
Base disc	0.50224	0.02646	0.03460
+EDM disc	0.41653	0.04516	0.06541
+Motion disc	0.76111	0.01436	0.01275
STMI-GAN	**0.70985**	**0.01108**	**0.01024**
0 to 4 seconds			
Org.Data. (Val vs Train)	1.65373	0.01225	0.01227
Res.sup. [24]	0.85732	0.13320	0.15644
NoGAN	1.07468	0.10245	0.12508
Base disc	1.58270	0.02197	0.02274
+EDM disc	1.22901	0.08416	0.09894
+Motion disc	1.77806	0.02434	0.02270
STMI-GAN	**1.69147**	**0.01888**	**0.01801**

Table 1. **Ablation Study.** Power Spectrum based metrics for different configurations of our model.

Our tested models are: NoGAN: generator network trained with the reconstruction losses over the whole sequence, and no adversarial loss. Base disc: is the same network, but using the encoder discriminator as loss for the generated part. +EDM disc: the network is trained using both the base and the EDM discriminators. +Motion disc: the network is trained using the base and motion discriminator. STMI-GAN: is the full network, trained with all joint discriminators.

Every discriminator seems to be adding information to the generated distribution. We can observe this qualitatively, and we can confirm it with the proposed metrics.

Entropy Analysis and KL distance Analysis. We should first note that the PSEnt in the original distribution is almost identical in every one second window, at approximately 0.678. This number represents the entropy of the uniform distribution, which means that the short term frequencies are fairly uniform. The PSEnt raises to 1.65, when we consider a 4 second time window. Such raise means that the long term motion has a biased, and more complex frequency distribution, which is not uniform but denser in some parts of the spectrum.

We can observe in Tab.1 that the Res.sup. [24] and NoGAN baselines decay in entropy as the seconds pass. Also we see that the KL divergence grows rapidly for the baselines and is around an order of magnitude higher than any of the GAN models. The GAN models all seem to have a good behavior, with PSEnt values close to the GT distribution. The Base disc model is already stable, but has some decay in entropy towards the end of the sequence. The Motion disc model seems to be pretty stable but it consistently overshoots on the entropy. This may be interpreted as the model overemphazising in moving. The EDM disc model seems to be harming at a first glance, since it considerably lowers the PSEnt, but the main point of adding this discriminator is to prevent unexpected poses from happening. It is a regularizer by design.

When we combine the three discriminators in the STMI-GAN model, the network approximates closely to the expected distribution. The STMI-GAN is stable both in PSEnt and PSKL and its performance does not decay as time passes. Indeed, it has a PSEnt close to the GT and a low PSKL, meaning that it is not only producing the same amount of motion but also the same kind of motion. We should note that the Human3.6M dataset has a considerable difference between the validation and train splits when following the standard protocol [24]. The PSKL between the validation and train splits for the whole sequence is around 0.012, almost symmetrical, and the PSKL between the generated distribution of STMI-GAN and validation is around 0.018, also symmetrical. This means that the distribution produced by the GAN is almost as close as the training and validation sets are.

L2 metric experiments. To demonstrate the point that L2 metric is not correlated with a realistic generation, we use a subset of the 120 test sequences of [8, 24], concretely the #8, #26, #27, #88. Fig. 5 shows the results of our approach and Res.sup. [24] on these sequences. Note that Res.sup. has a tendency to converge to the same pose (see the red skeleton in the center column), which is very close to the mean pose in the dataset. There is also the tendency to produce very small motions (see black and red at the bottom center frame). Indeed, [24] shows that the zero velocity baseline is often better than their model, specially for the class 'discussion' which has high uncertainty (see last row in Fig.5).

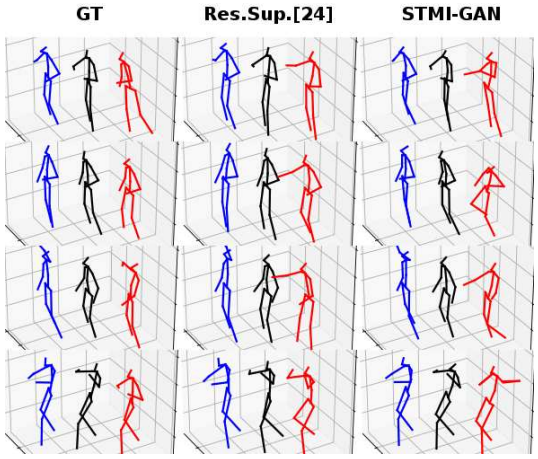

Figure 5. **H3.6 Examples.** Blue is first frame, black is first predicted frame, red is last predicted frame. Total length is 4 secs, 2 seed + 2 predicted.

When computing the L2 metric over angles as in [24] we obtain the following results: L2 Res.sup. $\rightarrow$ (0.69, 0.36, 0.64, 0.25); L2 STMI-GAN $\rightarrow$ (1.09, 0.74, 1.33, 0.96). Despite the baseline model has a lower L2, the sequences generated by our approach seem more diverse and realistic. These effects derive from the objectives used to train each model. The baseline models aim to minimize the spatial distance through the L2 loss. In our work we seek to reproduce the the distribution of human motion, and we use a GAN for this purpose. These objectives are not always aligned, and the L2 metric often fails to grasp the complexity of realistic human motions.

Noise Injection. It may seem that the noise injection would cause big differences in the output of the network, but actually the expected difference in the prediction is around $0.81mm$ per joint. This means that when called with the same seed sequence, the network produces almost identical sequences, only tweaking minor aspects of the sequence. This result confirms the effects of the noise reported in [19]. It is also interesting to note that the difference increases with the length of the prediction, meaning that indeed the injected noise is solving some level of uncertainty, but the maximum difference that we have measured predicting 4 seconds is $3.02mm$ per joint.

Qualitative Evaluation. We conducted evaluation with 15 person and four distinct surveys, all of them following the same scheme: a prediction model vs the ground truth. In the first two surveys we tested the baseline Res.sup. and STMI-GAN, to perform relative motion prediction. In the case of our model we removed the translation from the prediction to make it comparable. The last two surveys assess the absolute motion prediction. We use our NoGAN model as baseline vs our STMI-GAN model. In these surveys our goal is to obtain a 50% chance of being classified as real.

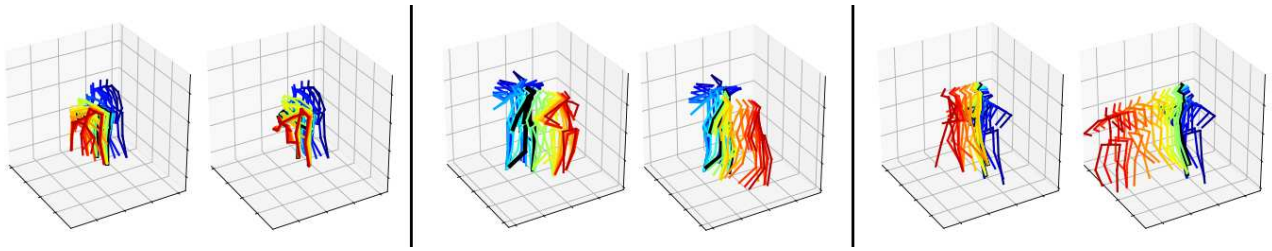

Figure 6. **Example result.** Three examples of the predicted motions (left: ground truth, right: predicted). The bluish colors are the part of the sequence that is observed. The prediction starts after the black skeleton and corresponds to the yellow-reddish colors.

Model	Motion	Min	Avg	Max
Res.sup [24]	Rel.	6.25%	31.88%	40.63%
STMI-GAN	Rel.	25.00%	33.54%	40.63%
NoGAN	Abs.	9.38%	31.46%	62.50%
STMI-GAN	Abs.	15.63%	**38.39**%	62.50%

Table 2. **Human Evaluation.** Percentage of times that a human evaluator thought a generated sequence was real. The min is the score of the "hardest" evaluator, who was fooled the less by the generative models. The max is the score of the "easiest" evaluator, who was confused more often by the model.

More than 50% would mean that the model is "more realistic" than the ground truth. As we can observe in Tab. 2 the results have a wide range of values. This is due to the fact that the survey was sent to a diverse audience. We can see that the baselines perform in a similar range of average values as our model, our model being bit better. However, the baselines perform very poorly to the trained eye, as the min score tells us. It is worth to note that while the relative motion prediction is an easier problem compared to the absolute motion prediction, the average score of our STMI-GAN is lower in the relative setting. This suggests that the relative motion generation is both harder for machine learning models and for humans. Also, the highest score on the surveys is in the max of the STMI-GAN in absolute prediction. This implies that some individuals were very convinced by the results of our model.

Fig.6 shows three examples. The sequence on the left is easy to predict, just continuing the motion of picking up things off the floor. The sequence on the center is a bit harder, as it is easy to guess that the person will continue walking, but the person halts after a couple of steps, and this is unexpected. The sequence on the right is challenging, because just before the generator begins its forecast, the person stops. This makes it hard to predict as the uncertainty rises and many options are plausible. We can see that the generator in fact guessed the action (walking) and the correct direction, but the speed of the motion is not accurate. We consider it however a good guess given the input.

7.2. Occlusion Completion

Finally, in Table 3 we report the robustness to different types of occlusion. In every test we used 80% of occlusion, *i.e.* we are trying to recover a sequence conditioning only on 20% of the data. The generator model is particularly robust to structured occlusions, it produces good results even without the GAN, we hypothesize that this is because it was trained to produce anthropomorphic guesses. When the occlusions happen at random, linear interpolation is also a good approach, but depending on the nature of the occlusion we may need a more robust model.

Problem	Linear Int	LR-Kalman[4]	NoGAN	STMI-GAN
Joint Occl.	232.06	329.23	**96.52**	108.99
Limb Occl.	209.45	312.40	**123.07**	189.09
Missing Frames	**50.42**	123.05	72.67	102.03
Noisy Transm.	**94.54**	308.98	98.53	110.29

Table 3. **Occlusion Completion.** We test different types of occlusion, concretely: **Joint Occlusions:** joints occluded at random in each frame. **Limb Occlusions:** joint chains representing limbs are occluded at random in each frame. **Missing Frames:** entire frames are occluded at random. **Missing transmission:** data points in any dimension are occluded at random. The table reports the L2 metric over coordinates(See 5).

8. Conclusions

We have presented a novel GAN architecture to predict 3D human motion from historical 3D skeleton poses. We have extended existing works, by also forecasting (beyond 2 seconds) the absolute position of the body. We have formulated our problem as an inpainting task in spatio-temporal volumes. In order to capture the essence and semantics of the human motion, the training of our network has been mostly guided by three independent discriminators. They encourage the generation of motion sequences with a similar frequency distribution to that of the original dataset. Since L2 is known not to be adequate to compare generated sequences, we have also proposed new metrics that estimate the frequency distribution of the datasets, grasping the concept of multiple possible futures. Experimental results on Human3.6M show the effectiveness of our model to generate highly realistic human motion predictions.

Acknowledgments: This work is supported in part by an Amazon Research Award and by the Spanish MiNeCo under projects HuMoUR TIN2017-90086-R and María de Maeztu Seal of Excellence MDM-2016-0656. Also was funded by the Deutsche Forschungsgemeinschaft GA 1927/4-1 and the ERC Starting Grant ARCA (677650).

References

[1] Antonio Agudo and Francesc Moreno-Noguer. Dust: Dual union of spatio-temporal subspaces for monocular multiple object 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 2, 2017.

[2] Antonio Valerio Miceli Barone. Low-rank passthrough neural networks. *arXiv preprint arXiv:1603.03116*, 2016.

[3] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.

[4] M Burke and Joan Lasenby. Estimating missing marker positions using low dimensional kalman smoothing. *Journal of biomechanics*, 49(9):1854–1858, 2016.

[5] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human motion analysis with deep metric learning. *arXiv preprint arXiv:1807.11176*, 2018.

[8] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, C Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. *arXiv preprint arXiv:1809.03036*, 2018.

[11] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *European Conference on Computer Vision*, pages 823–842, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Alejandro Hernandez Ruiz, Lorenzo Porzi, Samuel Rota Bulò, and Francesc Moreno-Noguer. 3d cnns on distance matrices for human action recognition. In *2017 ACM on Multimedia Conference*, pages 1087–1095. ACM, 2017.

[14] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[16] 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.

[17] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[18] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Taras Kucherenko, Jonas Beskow, and Hedvig Kjellström. A neural network approach to missing marker reconstruction in human motion capture. *arXiv preprint arXiv:1803.02665*, 2018.

[22] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018.

[23] Utkarsh Mall, G Roshan Lal, Siddhartha Chaudhuri, and Parag Chaudhuri. A deep recurrent framework for cleaning motion capture data. *arXiv preprint arXiv:1712.03380*, 2017.

[24] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4674–4683. IEEE, 2017.

[25] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3478–3487, 2018.

[26] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

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

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[31] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2007.

[32] Guiyu Xia, Huaijiang Sun, Beijia Chen, Qingshan Liu, Lei Feng, Guoqing Zhang, and Renlong Hang. Nonlinear low-rank matrix completion for human motion recovery. *IEEE Transactions on Image Processing*, 27(6):3011–3024, 2018.

[33] Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 4, 2017.

[34] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018.