# 3D Skeleton-based Human Motion Prediction with Manifold-Aware GAN

Baptiste Chopin, Naima Otberdout, Mohamed Daoudi, *Senior, IEEE,* Angela Bartolo

**Abstract**—In this work we propose a novel solution for 3D skeleton-based human motion prediction. The objective of this task consists in forecasting future human poses based on a prior skeleton pose sequence. This involves solving two main challenges still present in recent literature; (1) discontinuity of the predicted motion which results in unrealistic motions and (2) performance deterioration in long-term horizons resulting from error accumulation across time. We tackle these issues by using a compact manifold-valued representation of 3D human skeleton motion. Specifically, we model the temporal evolution of the 3D poses as trajectory, what allows us to map human motions to single points on a sphere manifold. Using such a compact representation avoids error accumulation and provides robust representation for long-term prediction while ensuring the smoothness and the coherence of the whole motion. To learn these non-Euclidean representations, we build a manifold-aware Wasserstein generative adversarial model that captures the temporal and spatial dependencies of human motion through different losses. Experiments have been conducted on CMU MoCap and Human 3.6M datasets and demonstrate the superiority of our approach over the state-of-the-art both in short and long term horizons. The smoothness of the generated motion is highlighted in the qualitative results.

**Index Terms**—Human motion prediction, manifold-valued representation, manifold-aware Wasserstein GAN.

## 1 Introduction

The problem of forecasting future human motion play a vital role in many applications in computer vision and robotics, such as human-robot interaction [1], autonomous driving [2] and computer graphics [3]. In this work, we propose a predictive model for short and long-term future 3D skeleton poses given an initial prior history. Addressing this issue involves two main challenges: How to represent the temporal evolution of the human motion to ensure the smoothness of the predicted sequences? and how to take the spatial correlations between human joints into account to avoid implausible poses?

Because of the explosion of deep learning and the availability of large scale datasets for human motion analysis, deep learning models have been widely exploited to address the problem of human motion prediction and especially Recurrent Neural Networks (RNN) [4], [5], [6], [7]. Indeed, RNN-based approaches achieved good advance in term of accuracy, however, the motions predicted with these methods present significant discontinuities due to the frame-by-frame regression process that discourage the global smoothness of the motion. In addition, recurrent models suffer from error accumulation across time, which increase error and worsen long-term forecasting performance. To remedy this, more recent works avoid these models and explore feed-forward networks instead. Including CNN [8], GNN [9] and fully-connected networks [10]. Thanks to their hierarchical structure, feed-forward networks can better deal with the spatial correlations of human joints than RNNs. However, an additional strategy is required to encode the temporal information when using these models. To face this issue, an interesting idea was to model the human motion as trajectory [11], [12].

In this work, we follow the idea of modeling motions as trajectories in time but in a different context from the previous work. Among the benefits of our representation, the possibility to map these trajectories to single compact points on a manifold, which helps preserving the continuity and the smoothness of the predicted motions. Besides, the compact representation avoids the problem of the error accumulation across time and makes our approach suitable for long-term prediction as illustrated in Figure 4. Nevertheless, the challenge here is that the resulting representations are manifold-valued data that cannot be manipulated with traditional generative models in a straightforward manner. To face this challenge, we introduce in this paper, a manifold-aware Wasserstein Generative Adversarial Networks (WGAN) that predict future skeleton poses given the input prior motion sequence that is encoded as a manifold-valued data. The spatial dependencies between human joints are taken into consideration in our method through additional loss functions that add more constraints on the predicted skeleton poses to ensure their plausibility. An overview of our prediction process is illustrated in Figure 1.

The contribution of this work can be summarized as follows: (1) To the best of our knowledge, we are the first to propose an approach that exploits compact manifold-valued representation for human motion prediction. By doing so, we model both the temporal and the spatial dependencies involved in human motion, resulting in smooth motions and plausible poses in long-term horizons. (2) We propose a predictive manifold-aware WGAN for motion prediction. (3) We propose a new loss function based on Gram matrix of the 3D poses that avoids predicting implausible poses. (4) Experimental results on Human 3.6M and the CMU MoCap datasets show quantitatively and visually the effectiveness


- *B. Chopin is with Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France. E-mail: baptiste.chopin@univ-lille.fr*
- *N. Otberdout is with Ai movement - University Mohammed VI Polytechnic, Rabat, Morocco, E-mail: naima.otberdout@um6p.ma*
- *M. Daoudi is with IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France, and Univ. Lille, CNRS, Centrale Lille, Institut Mines-Télécom, UMR 9189 CRIStAL, F-59000 Lille, France, E-mail: mohamed.daoudi@imt-nord-europe.fr*
- *Angela Bartolo is with Univ. Lille, CNRS, UMR 9193 SCALab, F-59000 Lille, France. E-mail: angela.bartolo@univ-lille.fr*

of our method for short-term and long-term prediction.

We presented some preliminary ideas of this work in [13]. With respect to [13], this paper provides more theoretical details about the proposed approach, it includes also new figures and more discussion. Furthermore, we present in this paper more results that further demonstrate the superiority of our solution over the state-of-the-art. While in [13], we compared the approaches based on joints position, we present here a new metric to evaluate the smoothness and the temporal evolution of the predicted motion. We provide a new qualitative evaluation for our ablation study to highlight the importance of the different losses of our method. We also demonstrate in this paper the ability of our method to predict longer sequences by recursive generation.

## 2 Related Work

**Human Motion Prediction with Deep Learning**. Given that the task of human motion prediction is a temporal dependent problem, recurrent models (RNN) were the first potential solution to be investigated, hence several works applied RNN and their variants to tackle this task. In [4], the authors proposed a model that incorporates a nonlinear encoder and decoder before and after recurrent layers. Their approach suffers from error accumulation and discontinuity between the last frame of the prior and the first frames of the generated sequence. Moreover, their approach only capture the temporal dependencies but ignore the spatial correlations between articulations. To deal with this problem, [5] proposed a Structural-RNN model relying on high-level spatio-temporal graphs. [6] take a different direction to minimize the error accumulation effect in RNNs; they used a feed forward network for pose filtering and a RNN for temporal filtering. However, this strategy only minimizes the accumulated error that still exists and deteriorates the performance of recurrent models in long-term prediction. Alternatively, more recent works exploit feed-forward networks. To model the temporal evolution with these models, various strategies have been suggested. In [8], [10], convolution across time was exploited to model the temporal dependencies with convolution networks, while [11] adopt Discrete Cosine Transform to encode the motion as trajectory. Graph neural networks were also applied for motion prediction [5], [9] as a suitable tool to model the spatial correlations involved between the articulations.
In this paper, we take a completely different direction and we propose to deal with human motion by exploiting a manifold-valued representation with generative adversarial models.

**Generative Adversarial Networks (GANs)**: GANs have been also exploited to address the problem of human motion prediction in [14] and [15], however, in order to model the temporal dependencies involved, they build their generator on RNN structures. In this way, the error accumulation problem is present in their model which may deteriorate its performance in the long-term. In our work we completely discard recurrent models by adopting a compact representation of the human motion.
Motivated by the interest of manifold-valued images in a variety of applications, [16] proposed manifold-aware WGAN. Inspired from this work, we build a manifold-aware WGAN that predict the future points of a poses trajectory given previous pose sequence. However, our model is different from the one proposed in [16] in two ways. Firstly, instead of unsupervised image generation from a vector noise, our model addresses the problem of predicting future manifold-valued representations from a manifold-valued inputs. In addition, we propose different objective functions to train our model on the task at hand.

**Modeling Human Motions as Trajectories on a Riemannian Manifold**: While our present work is the first that explores the benefit of manifold-valued trajectories for human motion prediction, representing 3D human poses and their temporal evolution as trajectories on a manifold was adopted in many recent works for action recognition. Different manifolds were considered in different studies [17], [18], [19]. More related to our work, in [20], a human action is interpreted as a parametrized curve and is seen as a single point on the sphere by computing its Square Root Velocity Function (SRVF). Accordingly, different actions were classified based on the distance between their associated points on the sphere. All papers mentioned above show the effectiveness of motion modeling as a trajectory in action recognition. Motivated by this fact, we show in this paper the interest of using such representation to address the recent challenges that still encountered in human motion prediction.

## 3 Human Motion Modeling

Two 3D skeleton representations were adopted for human motion prediction; angles based and 3D coordinates based representations. The first one models each joint by its rotation in term of Euler angles, while the second representation uses the 3D coordinates of the joints. More recently, [9], showed in their experiments that the angles based representation where two different sets of angles can represent the exact same pose, leads to ambiguous results and cannot provide a fair and reliable comparison. Motivated by this, we use 3D joint coordinates to represent our skeleton poses. The proposed approach relies on the representation of the human motions as points belonging to a hypersphere. However, the challenge encountered when working with such representation is the non-linearity of the hypersphere space which is a manifold. Accordingly, and following the state-of-the-art [16], [21] we used the term "manifold-valued data" to refer to our human motion representation and "manifold-aware GAN" to refer to the version of the generative adversarial model that learns to generate points on the hypersphere manifold.

### 3.1 Representation of Pose Sequences as Trajectories in $\mathbb{R}^n$

Let $k$ be the number of joints that compose the skeleton, we represent $P_t$ the pose of the skeleton at frame $t$ by a n-dimensional tuple: $P_t = [x_1(t), y_1(t), z_1(t) \dots x_k(t), y_k(t), z_k(t)]^T$, The pose $P_t$ encodes the positions of $k$ distinct joints in 3 dimensions. Consequently, an action sequence of length $T$ frames, can be described as a sequence $\{P_1, P_2 \dots, P_T\}$, where $P_i \in \mathbb{R}^n$ and $n = 3 \times k$.
This sequence represents the evolution of the action over time and can be considered as a result of sampling a continuous curve in $\mathbb{R}^n$. Based on this consideration, we model in what follows, each pose sequence of a skeleton, as a continuous curve in $\mathbb{R}^n$ that describes the continuous evolution of the sequence over time.
Let us represent the curve describing a pose sequence by a continuous parameterized function $\alpha(t) : I = [0, 1] \rightarrow \mathbb{R}^n$. In

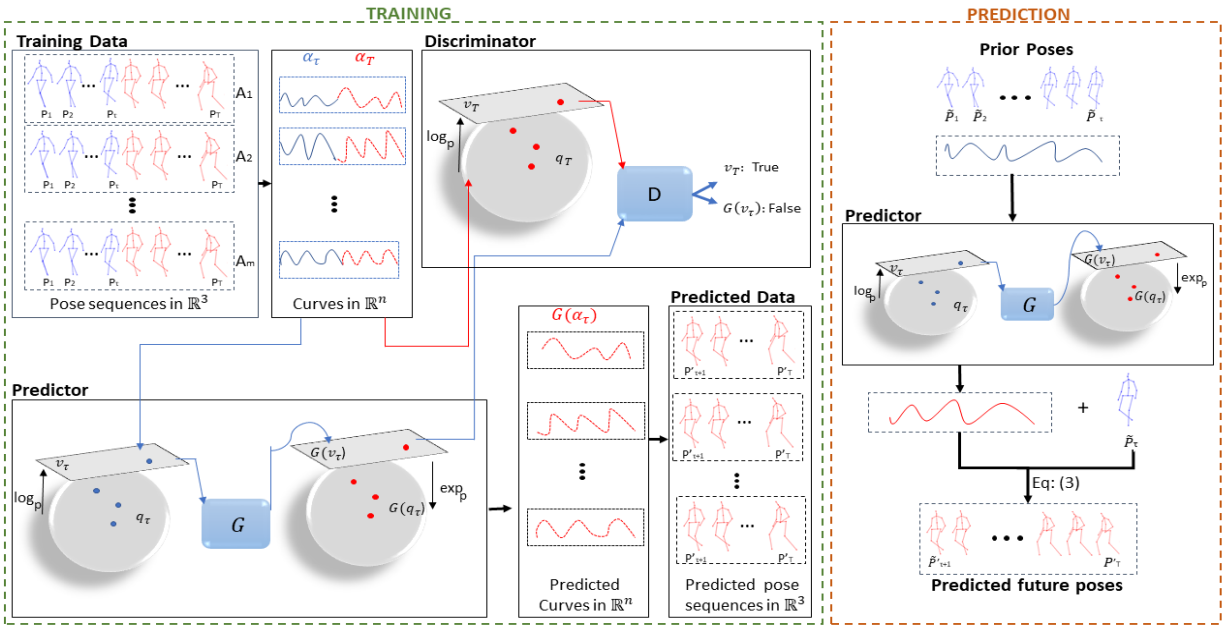



Fig. 1: Overview of the human motion training and prediction processes. Given a pose sequence history represented as a curve, then mapped to a single point in a hypersphere. The predictor maps the input point to a tangent space, then feeds it to the network $\mathcal{G}$ that predicts the future motion as a vector in $T_\mu(\mathcal{C})$. During training a discriminator is used to compare the mapped points from the ground truth to the generated ones. Exponential operator maps this vector to $\mathcal{C}$, before transforming it to a curve representing a motion. The predicted motion is transformed into a 3D human pose sequence corresponding to the future poses of the prior ones.

this work, we formulate the problem of human motion prediction given the first consecutive frames of the action as the problem of predicting the possible next points of the curve describing these first frames. More formally, the problem of predicting the future poses $\{P_{\tau+1}, P_{\tau+2}, \ldots, P_T\}$, given the first $\tau$ consecutive skeleton poses $\{P_1, P_2, \ldots, P_\tau\}$, where $\tau < T$, is formulated as the problem of predicting $\alpha(t)_{t=\tau+1\ldots T}$ given $\alpha(t)_{t=1\ldots\tau}$, such that, $\alpha(t)$ is the continuous function representing the curve associated to the pose sequence $\{P_1, P_2, \ldots, P_T\}$.

### 3.2 Representation of Human Motions as Elements in a Hypersphere $\mathcal{C}$

For the purpose of modeling and studying our curves, we adopt square-root velocity function (SRVF) proposed in [22]. It was successfully exploited for human action recognition [20], 3D face recognition [23] and facial expression generation [21]. Conveniently for us, this function maps each curve $\alpha(t)$ to one point in a hypersphere which provides a compact representation of the human motion. Specifically, for a given curve $\alpha(t) : I \to \mathbb{R}^n$, the square-root velocity function (SRVF) $q(t) : I \to \mathbb{R}^n$ is defined by the formula

$$q(t) = \begin{cases} \frac{\dot{\alpha}(t)}{\sqrt{\|\dot{\alpha}(t)\|}}, & \text{if } \|\dot{\alpha}(t)\| \neq 0 \\ 0, & \text{if } \|\dot{\alpha}(t)\| = 0 \end{cases} \tag{1}$$

where, $\|\cdot\|$ is the Euclidean 2-norm in $\mathbb{R}^n$. We can easily recover the curve (*i.e*, pose sequence) $\alpha(t)$ from the generated SRVF (*i.e*, dynamic information) $q(t)$ by,

$$\alpha(t) = \int_0^t \|q(s)\| q(s) ds + \alpha(0) \ , \tag{2}$$

where $\alpha(0)$ is the skeleton pose at the initial time step which corresponds in our case to the final time step of the history. In order to remove the scale variability of the curves, we scale them to be of length 1. Consequently, the SRVF corresponding to these curves are elements of a unit hypersphere in the Hilbert manifold $\mathbb{L}^2(I, \mathbb{R}^n)$ as explained in [22]. We will refer to this hypersphere as $\mathcal{C}$, such that, $\mathcal{C} = \{q : I \to \mathbb{R}^n | \ \|q\| = 1\} \subset \mathbb{L}^2(I, \mathbb{R}^n)$ . Each element of $\mathcal{C}$ represents a curve in $\mathbb{R}^n$ associated with a human motion. As $\mathcal{C}$ is a hypersphere, the geodesic length between two elements $q_1$ and $q_2$ is defined as:

$$d_{\mathcal{C}}(q_1, q_2) = \cos^{-1}(\langle q_1, q_2 \rangle) \ . \tag{3}$$

## 4 Architecture and Loss Functions

Given a set of $m$ action sequences $\{\{P_1, P_2, \ldots P_T\}_i\}_{i=1}^m$ of $T$ consecutive skeleton poses. Let us consider the first $\tau$ poses $(\tau < T)$ as the actions history represented by their corresponding SRVFs $\{q_\tau^i\}_{i=1}^m$, and the last $(T - \tau)$ skeleton configurations as the future poses $\{q_T^i\}_{i=1}^m$ to be predicted.

Motivated by the success of generative adversarial networks, we aim to exploit these generative models to learn an approximation of the function $\Phi : \mathcal{C} \to \mathcal{C}$ that predicts the $(T - \tau)$ future poses from their associated $\tau$ prior ones. This can be achieved by learning the distribution of SRVFs data corresponding to future poses, on their underlying manifold *i.e.*, hypersphere. As stated earlier, SRVFs representations are manifold-valued data that cannot be used directly by classical GANs. This is due to the fact that the distribution of data having values on a manifold is quite different from the distribution of those lying on Euclidean space. [16], exploited the tangent space of the involved manifold and propose a manifold-aware WGAN that generates random data on a manifold. Inspired from this work, we propose a manifold-aware WGAN for motion prediction, to which we refer as PredictiveMA-WGAN, that can predict the future poses from the past ones. This

is achieved by using the prior poses as input condition to the MA-WGAN. This condition is also represented by its SRVF; as a result PredictiveMA-WGAN takes manifold-valued data as input to predict its future, which is also a manifold-valued data.

### 4.1 Network Architecture

PredictiveMA-WGAN consists of two networks trained in an adversarial manner: the predictor $\mathcal{G}$ and the discriminator $\mathcal{D}$. The first network $\mathcal{G}$ adjust its parameters to learn the distribution $\mathbb{P}_{q_T}$ of the future poses $q_T$ conditioned on the input prior ones $q_\tau$, while $D$ tries to distinguish between the real future poses $q_T$ and the predicted ones $\hat{q}_T$. During the training of these networks, we iteratively map the SRVF data back and forth to the tangent space using the exponential and the logarithm maps, defined in a particular point on the hypersphere.

The predictor network is composed of multiple upsampling and downsampling blocks. It takes as input the prior poses $q_\tau$ and output the predicted future poses $\hat{q}_T$. A fully connected layer with $36864$ output channels and five upsampling blocks with $512$, $256$, $128$, $64$ and $1$ output channels, process the input prior pose. These upsampling blocks are composed of the nearest-neighbor upsampling followed by a $3 \times 3$ stride $1$ convolution and a Relu activation. The Discriminator $D$ contains three downsampling blocks with $64$, $32$ and $16$ output channels. Each block is a $3 \times 3$ stride $1$ Conv layer followed by batch normalization and Relu activation. These layers are then followed by two fully connected (FC) layers of $1024$ and $1$ outputs. The first FC layer uses Leaky ReLU and batch normalization.

### 4.2 Loss Functions

In general, the objective of the training consists in minimizing the Wasserstein distance between the distribution of the predicted future poses $\mathbb{P}_{\hat{q}_T}$ and that of the real ones $\mathbb{P}_{q_T}$ provided by the dataset. Toward this goal we make use of the following loss functions:

**Adversarial loss** – We propose an adversarial loss for predicting manifold-valued data from their history. The predictor takes a manifold-value data $q_\tau$ as input rather than a random vector as done in [16], which requires to map these data to a tangent space using the logarithm map before feeding them to the network. Our adversarial loss is the following:

$$\begin{aligned}\mathcal{L}_a =& \mathbb{E}_{q_T \sim \mathbb{P}_{q_T}}\left[\mathcal{D}\left(\log_\mu(q_T)\right)\right] \\ & -\mathbb{E}_{\mathcal{G}(log_\mu(q_\tau)) \sim \mathbb{P}_{\hat{q}_T}}\left[\mathcal{D}\left(\log_\mu\left(\exp_\mu(\mathcal{G}(\log_\mu(q_\tau)))\right)\right)\right] \\ & +\lambda \mathbb{E}_{\widetilde{q} \sim \mathbb{P}_{\widetilde{q}}}\left[\left(\|\nabla_{\widetilde{q}} \mathcal{D}(\widetilde{q})\| - 1\right)^2\right],\end{aligned} \tag{4}$$

where the exponential map, $\exp_\mu(.)$: $T_\mu(C) \mapsto \mathcal{C}$ has a simple expression:

$$\exp_\mu(s) = cos(\|s\|)\mu + sin(\|s\|)\frac{s}{\|s\|},$$

and the inverse exponential map also called logarithm map $\log_\mu(q)$: $\mathcal{C} \mapsto T_\mu(C)$ is given by:

$$\log_\mu(q) = \frac{d_\mathcal{C}(q,\mu)}{sin(d_\mathcal{C}(q,\mu))}(q - cos(d_\mathcal{C}(q,\mu))\mu)$$

where $d_\mathcal{C}(.,.)$ is the geodesic distance defined by (3). The last term of $\mathcal{L}_a$ represents the gradient penalty proposed in [24]. $\widetilde{q}$ is a random sample following the distribution $\mathbb{P}_{\widetilde{q}}$, which is sampled uniformly along straight lines between pairs of points sampled from the real distribution $\mathbb{P}_{q_T}$ and the generated distribution $\mathbb{P}_{\hat{q}_T}$. It is given by: $\widetilde{q} = (1-a)\log_\mu(q_T) + a\log_\mu(\exp_\mu(\mathcal{G}(\log_\mu(q_\tau))))$, where $\nabla_{\widetilde{q}} D(\widetilde{q})$ is the gradient with respect to $\widetilde{q}$, and $0 \leqslant a \leqslant 1$.

The reference point $\mu$ of the tangent space used in our training is set to the mean of the training data. For a given set of training trajectories $q_1, \ldots, q_m$. The mean is given by the Karcher mean [25] in $\mathcal{C}$,

$$\mu = \underset{q_i \in \mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{m} d^2_\mathcal{C}(\mu, q_i) \tag{5}$$

where $\{q_i\}_{i=1}^m$ is $m$ training data. We present a commonly used algorithm for finding Karcher mean for a given set of curves [26]. This approach, presented in Algorithm 1. This computation is based on an iterative calculation which converges to the optimal solution which is the mean.

**Algorithm 1:** Karcher mean on $\mathcal{C}$

**Input:** Given SRVFs $\{q_1, q_2 \cdots q_N\}$,
$\epsilon = 0.9$, $\tau$: threshold which is a very small number
**Output:** $\mu_j$ : mean of $\{q_i\}_{i=1:N}$

**1-** $\mu_0$: initial estimate of Karcher mean, for example one could just take $\mu_0 = q_1$, j=0
**repeat**
  **for** $i \leftarrow 1$ **to** $N$ **do**
    **2-** Compute $v_i = \frac{\theta_i}{sin(\theta_i)}(q_i^* - cos(\theta_i)\mu_j)$, where $cos(\theta_i) = \langle \mu_j, q_i^* \rangle$
    **3-** Compute the average direction $\overline{v} = \frac{1}{n}\sum_{i=1}^n v_i$
    **4-** Move $\mu_j$ in the direction of $\overline{v}$ by $\epsilon$: $\mu_{j+1} = cos(\epsilon\|\overline{v}\|)\mu_j + sin(\epsilon\|\overline{v}\|)\frac{\overline{v}}{\|\overline{v}\|}$
  **5-** j=j+1
**until** $\|\overline{v}\| < \tau$;

**Reconstruction loss** – In order to predict motions close to their ground truth, we add a reconstruction loss $\mathcal{L}_r$. This loss function quantifies the similarities in the tangent space $T_\mu(\mathcal{C})$ between the tangent vector $\log_\mu(q_T)$ of the ground truth $q_T$ and its associated reconstructed vector $\log_\mu(\exp_\mu(\mathcal{G}(\log_\mu(q_\tau))))$. It is given by,

$$\mathcal{L}_r = \|\log_\mu(\exp_\mu(\mathcal{G}(\log_\mu(q_\tau)))) - \log_\mu(q_T)\|_1 \ , \tag{6}$$

where $\|.\|_1$ denotes the $L_1$-norm.

**Skeleton integrity loss** – We propose a new loss function $\mathcal{L}_s$ that minimizes the distance between the predicted poses and their ground truth as a remedy to the generation of abnormal skeleton poses. Indeed, the aforementioned loss functions rely only on the SRVF representations, which imposes constraints only on the dynamic information. However, to capture the spatial dependencies between joints that avoid implausible poses, we need to impose constraints on the predicted poses directly instead of their motions. By doing so, we predict dynamic changes that fit the initial pose and result in a long-term plausibility. The proposed loss function is based on the Gram matrix of the joint configuration $P$, $G = PP^T$, where $P$ can be seen as $k \times 3$ matrix. Let $G_i, G_j$ be two Gram matrices, obtained from joint poses $P_i, P_j \in \mathbb{R}^{k \times 3}$.

The distance between $G_i$ and $G_j$ can be expressed [27, p. 328] as:

$$\Delta(G_i, G_j) = \mathrm{tr}\,(G_i) + \mathrm{tr}\,(G_j) - 2\sum_{i=1}^{3}\sigma_i\ , \tag{7}$$

where $\mathrm{tr}(.)$ denotes the trace operator, and $\{\sigma_i\}_{i=1}^{3}$ are the singular values of $P_j^T P_i$. The resulting loss function is,

$$\mathcal{L}_s = \frac{1}{m}\frac{1}{\tau}\sum_{i=1}^{m}\sum_{t=1}^{\tau}\Delta(P_{i,t}, \hat{P}_{i,t})\ , \tag{8}$$

where $m$ represents the number of training samples, $\tau$ is the length of the predicted sequence, $P$ is the ground truth pose and $\hat{P}$ is the predicted one.

**Bone length loss** – To ensure the realness of the predicted poses, we impose further restrictions on the length of the bones. This is achieved through a loss function that forces the bone length to remain constant over time. Considering $b_{i,j,t}$ and $\hat{b}_{i,j,t}$ the $j$-th bones at time $t$ from the ground truth and the predicted $i$-th skeleton, respectively, we compute the following loss :

$$\mathcal{L}_b = \frac{1}{m}\frac{1}{\tau}\frac{1}{B}\sum_{i}^{m}\sum_{t=1}^{\tau}\sum_{j}^{B}\|b_{i,j,t} - \hat{b}_{i,j,t}\|\ , \tag{9}$$

with B the number of bones in the skeleton representation.

**Global loss** – PredictiveMA-WGAN is trained using a weighted sum of the four loss functions $\mathcal{L}_a$, $\mathcal{L}_r$, $\mathcal{L}_s$ and $\mathcal{L}_b$ introduced above, such that,

$$\mathcal{L} = \beta_1\mathcal{L}_a + \beta_2\mathcal{L}_r + \beta_3\mathcal{L}_s + \beta_4\mathcal{L}_b. \tag{10}$$

The parameters $\beta_i$ are the coefficients associated to different losses, they are set empirically in our experiments.

The algorithm 2 summarizes the main steps of our approach. It is divided in two stages, first we outline the steps needed to train our model, then we present the prediction stage, where the trained model is used to predict future poses of a given sequence.

**Algorithm 2:** PredictiveMAWGAN algorithm

```
// Training
```
**Data:** $\{q_\tau^i\}_{i=1}^m$: SRVFs of training prior poses, $\{q_T^i\}_{i=1}^m$: real future poses, $\theta_0$ : initial parameters of $\mathcal{G}$, $\eta_0$ : initial parameters of $\mathcal{D}$, $\epsilon$: learning rate, $K$: batch size, $\lambda$: balance parameter of gradient penalty, $\zeta$: iterations number.
**Result:** $\theta$: generator learned parameters.

**for** $i = 1 \dots \zeta$ **do**
- Sample a mini-batch of $K$ random prior poses $\{q_\tau^j\}_{j=1}^K \sim \mathbb{P}_{q_\tau}$;
- Sample a mini-batch of $K$ real future poses; $\{q_T^j\}_{j=1}^K \sim \mathbb{P}_{q_T}$;
- $D_\eta \leftarrow \Delta_\eta(\mathcal{L})$, $\mathcal{L}$ is given by Eq. 10;
- $\eta \leftarrow \eta + \epsilon.AdamOptimizer(\eta, D_\eta)$;
- Sample a mini-batch of $K$ random prior poses; $\{q_\tau^j\}_{j=1}^K \sim \mathbb{P}_{q_\tau}$;
- Compute $\{\mathcal{G}_\theta(\log_\mu(q_\tau^j))\}_{j=1}^K$;
- $G_\theta \leftarrow \Delta_\theta(-D_\eta\left(\log_\mu\left(\exp_\mu(\mathcal{G}_\theta(\log_\mu(q_\tau)))\right)\right))$
- $\theta \leftarrow \theta + \epsilon.AdamOptimizer(\theta, G_\theta)$;

```
// Prediction
```
**Data:** $\theta$: generator learned parameters, $\{P_i\}_{i=1}^\tau$: Prior poses of a testing sequence.
**Result:** $\{\hat{P}_i\}_{i=\tau+1}^T$: Predicted future poses.

Compute $q_\tau$ from $\{P_i\}_{i=1}^\tau$ with Eq. 1;
Compute $\hat{q}_T = \exp_\mu(\mathcal{G}_\theta(\log_\mu(q_\tau)))$ using the learned parameters $\theta$;
Transform $\hat{q}_T$ into pose sequence $\{\hat{P}_i\}_{i=\tau+1}^T$ using Eq. 2, with $\alpha(0) = P_\tau$

## 5 Experiments

We evaluate the proposed approach with extensive experiments on two popular datasets. In this part we show and discuss our results.

### 5.1 Datasets and Pre-processing

**Human 3.6M** [28]. it is a database that contains 11 subjects performing 15 different actions (Walking, Phoning, Taking photos…). It is one of the largest dataset and the most commonly used for evaluating human motion prediction with 3D skeletons. Following the protocol set by previous approaches [7], [29] we train our model on 6 subjects and test it on the specific clips of the 5th subject. In the same way as [29] out of the 32 skeletal joints we only use 17, we remove the joints that correspond to duplicate joints, hands and feet.

For Human3.6M we take the database processed by [5] formatted in exponential map and we use their code to convert them to Cartesian coordinates. During our preprocessing step we down sample the sequence from 50 fps to 25 fps and then perform a normalization by subtracting the mean, dividing by the norm and subtracting the coordinates of the root joint (hips). In the dataset proposed by [5] each class of each subject is composed of 2 long sequences. We divide those into smaller sequence for short term prediction (60 frames) and long term prediction (75 frames), following [8]. When generating these smaller sequence we avoid overlap, *e.g.* when generating sequence for long term prediction (75 frames) the first sequence contains the frames 1 to 75, the second frames 76 to 150 and so on. This leaves us with 3480 training samples and 812 testing samples for short-term prediction and 2769 training samples and 644 testing samples for long-term prediction.

**CMU Motion Capture** (CMU MoCap). CMU Mocap dataset [1] is a database that contains 5 categories of motion, each containing several actions. Following [8], we keep only 8 actions: 'basketball', 'basketball signal', 'directing traffic','jumping', 'running', 'soccer', 'walking' and 'washing window'. We keep the same joint configuration as for Human3.6M and preprocess the data the same way. This leads to 2871 training samples and 704 test samples for short-term prediction and 2825 training samples and 677 test samples for long-term prediction.

### 5.2 Implementation Details

We train our network with a batch size of 64 on 500 epochs and with a learning rate of $10^{-4}$ using the Adam optimizer [30]. We use $\beta_1 = 1$, $\beta_2 = 1$, $\beta_3 = 10$ and $\beta_4 = 10$ for the loss coefficients. Our Implementation run on a PC with a Nvidia Quadro RTX 6000 GPU, two 2.3Ghz processors and 64Go of RAM using Tensorflow 2.2.

1. http://mocap.cs.cmu.edu

### 5.3 Evaluation Metrics and Baselines

We use state-of the art methods for motion prediction that were based on 3D coordinate representation for our comparison. This includes RNN based method (Residual sup). [7], CNN based method (ConvSeq2Seq) [8] and graph models; (FC-GCN) [9] and (LDRGCN) [29].

The zero velocity baseline introduced by [7] is a very simple baseline that use the last observed frame at $t = \tau$ as the value for all the predicted frames, we also compare ourselves to this baseline. The result of LDRGCN are those reported by the authors for the method trained with data in 3D coordinate space. Concerning FC-GCN, ConvSeq2Seq and Residual sup., the results are those reported by [9] using 3D coordinate data for training. We report the results presented by [29] for long-term prediction (1000ms) results on Human 3.6M, since they are not provided in [9]. The long term results for Residual sup. are not available, we did not include it in our results.

We base our quantitative evaluation on the Mean Per Joint Position Error (MPJPE) [28] in millimeter following the state-of-the-art [29]. The metric compare the 3D coordinates of the ground truth with the predicted motions. It is given by,

$$MPJPE = \sqrt{\frac{1}{\Delta t}\frac{1}{k}\sum_{t=\tau+1}^{\tau+\Delta t}\sum_{j=1}^{k}\|p_{t,j} - \hat{p}_{t,j}\|^2}\ , \qquad (11)$$

where $p_{t,j} = [x_j(t), y_j(t), z_j(t)]$ are the coordinates of joint $j$ at time $t$ from the ground truth sequence, $\hat{p}_{t,j}$ the coordinates from the generated sequence, $k$ the total number of joints in the skeleton, $\tau$ the number of frames in prior sequence and $\Delta t$ the number of predicted frames at which the sequence is evaluated.

While MPJPE evaluates the generated samples based on joints positions, it is not enough to assess the evolution of the motion. To complete our assessment we further compare our method with the other approaches based on the evolution along time of the speed of the predicted sequences, we refer to this metric as MPJS (Mean Per Joint Speed). It is computed as follows,

$$MPJS(t) = \frac{1}{k}\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{k}\|p_{i,t-1,j} - p_{i,t,j}\|\ , \qquad (12)$$

with $p_{i,t-1,j}$ and $p_{i,t,j}$ the the coordinates of joint j at time t-1 and t respectively, $k$ the number of joint in the skeleton and M the total number of samples in the test set.

### 5.4 Quantitative Comparison

#### 5.4.1 Joints position-based evaluation

To be consistent with recent works, the result are reported for short term prediction and long term prediction. For short term prediction we predict 10 future frames within 400ms given 10 historical frames while we predict 25 in 1s based on 25 prior frames for long term prediction. In Table 1 we show the comparison of our results with recent methods that use 3D joint coordinates representation. This representation as been proven to provide a more reliable comparison than the angle based representation by [9]. The results in the table show the clear superiority of our method over methods from the state-of-the-art on both datasets. We highlight that our approach is very competitive with the LDRGN approach for very short term prediction (80ms and 160ms) while outperforming it for longer prediction (320ms, 400ms and 1s). This demonstrate that it is robust when predicting long term motions that stay close to the ground truth.

| | Human3.6M average | | | | |
|---|---|---|---|---|---|
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 19.6 | 32.5 | 55.1 | 64.4 | 107.9 |
| Residual sup. | 30.8 | 57.0 | 99.8 | 115.5 | - |
| convSeq2Seq | 19.6 | 37.8 | 68.1 | 80.3 | 140.5 |
| FC-GCN | 12.2 | 25.0 | 50.0 | 61.3 | 114.7 |
| LDRGCN | **10.7** | **22.5** | 43.1 | 55.8 | 97.8 |
| Ours | 12.6 | **22.5** | **41.9** | **50.8** | **96.4** |
| | CMU MoCap average | | | | |
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 18.4 | 31.4 | 56.2 | 67.7 | 130.5 |
| Residual sup. | 15.6 | 30.5 | 54.2 | 63.6 | 96.6 |
| convSeq2Seq | 12.5 | 22.2 | 40.7 | 49.7 | 84.6 |
| FC-GCN | 11.5 | 20.4 | 37.8 | 46.8 | 96.5 |
| LDRGCN | **9.4** | 17.6 | 31.6 | 43.1 | 82.9 |
| Ours | **9.4** | **15.9** | **29.2** | **38.3** | **80.6** |

TABLE 1: Average error over all actions of Human3.6M and CMU MoCap. The short-term in $80$,$160$,$320$,$400ms$, and long-term in $1s$.

In Table 2 and 3 we report the results for the literature and for our method on all action classes of Human3.6M and CMU Mocap datasets respectively. The baseline methods adopt a protocol that consist in reporting he average error on eight randomly sampled test sequences. We found that this random sampling can significantly affect the error and makes it hard to present a fair comparison. To avoid this, we decided to report to run the experiment on 8 randomly selected test sequences 100 times, we then report the average error and the standard deviation for these 100 runs for the results of our model. With the standard deviation we can have a better measurement of the general performance of our architecture on different test sequences.

According to Tables 2 and 3, our method perform better than the state-of the art, especially when dealing with long term prediction, these results are consistent with the average error over all actions classes. Interestingly our results also show that the simple zero velocity baseline sometimes outperforms the state of the art approach on long term prediction (*e.g,* Photo, Sitting and Walking dog for Human3.6H, Soccer and Jumping for CMU MoCap). On the other hand for short term prediction it is always outperformed by the predictions methods. This may be an indication that the MPJPE is not the best suited metric for the problem and a motivation to find a better more representative metric in future works. The results show that the previous approaches performance decrease over time, while ours proves more robust in long term horizons, we are show to perform better than both the zero velocity baseline and the literature. We can notice than some classes present a very large variance (*e.g,* jumping) while for other the variance is very low (*e.g,* running). This is due to the number of samples which can be be very different from a class to another but also to the high diversity of samples for some classes. Other classes that present less variability (*e.g,* walking) have a reduced variance.

#### 5.4.2 Motion-based evaluation

To further assess the generated sequences, we evaluate their motion based on the MPJS introduced before. By looking at the evolution of this metric, we can compare our generated motion with the ground truth ones and evaluate the ability of our model to predict motion in long term prediction. To this end we show

| | Directions | | | | | Discussion | | | | | Eating | | | | | Greeting | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 16.0 | 27.1 | 46.4 | 53.9 | 83.9 | 17.8 | 29.7 | 51.0 | 59.8 | 103.1 | 13.5 | 21.9 | 37.0 | 43.9 | 83.3 | 26.4 | 43.7 | 70.1 | 80.5 | **124.9** |
| Residual sup. | 36.5 | 56.4 | 81.5 | 97.3 | - | 31.7 | 61.3 | 96.0 | 103.5 | - | 17.6 | 34.7 | 71.9 | 87.7 | - | 37.9 | 74.1 | 139.0 | 158.8 | - |
| convSeq2Seq | 22.0 | 37.2 | 59.6 | 73.4 | 118.3 | 18.9 | 39.3 | 67.7 | 75.7 | 123.9 | 13.7 | 25.9 | 52.5 | 63.3 | 74.4 | 24.5 | 46.2 | 90.0 | 103.1 | 191.2 |
| FC-GCN | 12.6 | 24.4 | 48.2 | 58.4 | 89.1 | 9.8 | 22.1 | 39.6 | 44.1 | 78.5 | 8.8 | 18.9 | 39.4 | 47.2 | 57.1 | 14.5 | 30.5 | 74.2 | 89.0 | 148.4 |
| LDRGCN | 13.1 | 23.7 | 44.5 | 50.9 | **78.3** | **9.4** | **20.3** | **35.2** | **41.2** | **67.4** | **7.6** | **15.9** | 37.2 | 41.7 | **53.8** | **9.6** | **27.9** | 66.3 | 78.8 | 129.7 |
| Ours | **11.1** | **20.9** | **38.8** | **47.0** | **83.5** | 11.9 | **22.7** | 44.8 | 54.6 | 102.2 | **9.0** | **15.9** | **29.1** | **35.0** | 65.3 | 19.6 | 35.1 | **64.0** | **78.2** | **126.8** |
| | ±2.7 | ±4.9 | ±8.4 | ±9.7 | ±15.3 | ±1.9 | ±3.4 | ±6.5 | ±7.7 | ±16.5 | ±1.5 | ±2.8 | ±4.8 | ±5.3 | ±6.8 | ±3.4 | ±6.8 | ±13.1 | ±16.1 | ±16.7 |
| | Phoning | | | | | Photo | | | | | Posing | | | | | Purchase | | | | |
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 15.8 | 26.5 | 43.7 | 51.0 | 92.3 | 16.9 | 28.4 | 49.2 | 58.3 | **98.8** | 20.4 | 34.7 | 61.5 | 73.3 | 136.1 | 22.1 | 36.5 | 61.8 | 72.2 | 126.3 |
| Residual sup. | 25.6 | 44.4 | 74.0 | 84.2 | - | 23.6 | 47.4 | 94.0 | 112.7 | - | 27.9 | 54.7 | 131.3 | 160.8 | - | 40.8 | 71.8 | 104.2 | 109.8 | - |
| convSeq2Seq | 17.2 | 29.7 | 53.4 | 61.3 | 127.5 | 14.0 | 27.2 | 53.8 | 66.2 | 151.2 | 16.1 | 35.6 | 86.2 | 105.6 | 163.9 | 29.4 | 54.9 | 82.2 | 93.0 | 139.3 |
| FC-GCN | 11.5 | 20.2 | 37.9 | 43.2 | 94.3 | **6.8** | 15.2 | 38.2 | 49.6 | 125.7 | 9.4 | 23.9 | 66.2 | 82.9 | 143.5 | 19.6 | 38.5 | 64.4 | 72.2 | 127.2 |
| LDRGCN | **10.4** | **14.3** | **33.1** | **39.7** | 85.8 | 7.1 | **13.8** | **29.6** | 44.2 | 116.4 | **8.7** | **21.1** | 58.3 | 81.9 | **133.7** | 16.2 | 36.1 | 62.8 | 76.2 | **112.6** |
| Ours | **11.7** | 19.4 | **34.9** | **42.3** | **81.8** | **8.8** | **16.0** | **32.4** | **40.9** | **98.9** | 13.7 | **25.9** | **50.0** | **61.1** | **137.7** | **14.2** | **26.5** | **48.3** | **58.1** | **120.8** |
| | ±2.2 | ±3.6 | ±6.4 | ±7.6 | ±9.8 | ±2.0 | ±3.5 | ±6.9 | ±8.6 | ±16.1 | ±3.3 | ±6.3 | ±11.0 | ±12.7 | ±12.8 | ±2.5 | ±4.8 | ±9.8 | ±12.5 | ±19.0 |
| | Sitting | | | | | Sitting Down | | | | | Smoking | | | | | Waiting | | | | |
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 14.6 | 23.9 | 40.9 | 48.4 | **94.7** | 19.5 | 32.4 | 53.5 | 61.8 | **112.2** | 14.9 | 24.6 | 41.7 | 49.3 | 84.0 | 17.0 | 28.2 | 48.9 | 57.8 | 99.4 |
| Residual sup. | 34.5 | 69.9 | 126.3 | 141.6 | - | 28.6 | 55.3 | 101.6 | 118.9 | - | 19.7 | 36.6 | 61.8 | 73.9 | - | 29.5 | 60.5 | 119.9 | 140.6 | - |
| convSeq2Seq | 19.8 | 42.4 | 77.0 | 88.4 | 132.5 | 17.1 | 34.9 | 66.3 | 77.7 | 177.5 | 11.1 | 21.0 | 33.4 | 38.3 | 52.2 | 17.9 | 36.5 | 74.9 | 90.7 | 205.8 |
| FC-GCN | 10.7 | 24.6 | 50.6 | 62.0 | 119.8 | 11.4 | 27.6 | 56.4 | 67.6 | 163.9 | 7.8 | 14.9 | 25.3 | 28.7 | 44.3 | 9.5 | 22.0 | 57.5 | 73.9 | 157.2 |
| LDRGCN | **9.2** | 23.1 | 47.2 | 57.7 | 106.5 | **9.3** | **21.4** | **46.3** | **59.3** | 144.6 | 8.1 | **13.4** | **24.8** | **24.9** | **43.1** | **9.2** | **17.6** | 47.2 | 71.6 | 127.3 |
| Ours | **10.4** | **17.9** | **33.1** | **40.7** | **97.7** | 15.8 | 28.2 | **52.9** | **64.5** | **125.2** | **7.9** | **14.3** | **25.2** | **30.4** | 63.4 | **11.4** | **20.3** | **38.8** | **47.2** | **94.0** |
| | ±2.8 | ±3.5 | ±5.3 | ±6.4 | ±14.0 | ±3.4 | ±5.1 | ±9.3 | ±11.5 | ±23.3 | ±1.6 | ±2.7 | ±4.5 | ±5.2 | ±9.7 | ±3.1 | ±4.3 | ±7.6 | ±9.0 | ±13.7 |
| | Walking Dog | | | | | Walking | | | | | Walking Together | | | | | Average | | | | |
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 26.9 | 42.3 | 69.2 | 79.5 | 119.2 | 28.1 | 49.2 | 86.0 | 100.3 | 149.1 | 23.5 | 39.2 | 65.4 | 75.6 | 111.3 | 19.6 | 32.5 | 55.1 | 64.4 | 107.9 |
| Residual sup. | 60.5 | 101.9 | 160.8 | 188.3 | - | 23.8 | 40.4 | 62.9 | 70.9 | - | 23.5 | 45.0 | 71.3 | 82.8 | - | 30.8 | 57.0 | 99.8 | 115.5 | - |
| convSeq2Seq | 40.6 | 74.7 | 116.6 | 138.7 | 210.2 | 17.1 | 31.2 | 53.8 | 61.5 | 89.2 | 15.0 | 29.9 | 54.3 | 65.8 | 149.8 | 19.6 | 37.8 | 68.1 | 80.3 | 140.5 |
| FC-GCN | 32.2 | 58.0 | 102.2 | 122.7 | 185.4 | **8.9** | 15.7 | 29.2 | 33.4 | 50.9 | 8.9 | 18.4 | 35.3 | 44.3 | 102.4 | 12.2 | 25.0 | 50.0 | 61.3 | 114.7 |
| LDRGCN | 25.3 | 56.6 | 87.9 | 99.4 | 143.2 | **8.9** | **14.9** | **25.4** | **29.9** | **45.8** | **8.2** | **18.1** | **31.2** | **39.4** | 79.2 | **10.7** | **22.5** | 43.1 | 55.8 | 97.8 |
| Ours | **19.3** | **34.2** | **65.6** | **77.5** | **117.8** | 12.0 | 21.1 | 35.6 | 42.4 | 68.2 | 11.6 | **19.7** | 34.5 | **41.8** | **63.4** | 12.6 | **22.5** | **41.9** | **50.8** | **96.4** |
| | ±5.9 | ±9.5 | ±17.8 | ±19.7 | ±23.7 | ±1.1 | ±1.7 | ±2.9 | ±3.8 | ±5.3 | ±1.1 | ±1.6 | ±3.0 | ±3.8 | ±6.4 | | | | | | |

TABLE 2: Motion prediction results measured with Equation 11 on Human3.6M dataset. Short-term results are reported within 80, 160, 320, 400ms, and long-term in 1s. Best results in bold while state-of-the-art best results that fit in our confidence interval are also written bold.

| | Basketball | | | | | Basketball signal | | | | | Directing traffic | | | | | Jumping | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 20.3 | 34.6 | 62.2 | 75.0 | 143.5 | 6.4 | 11.0 | 19.9 | 24.2 | 50.5 | 26.6 | 41.9 | 69.1 | 81.9 | 155.3 | 21.4 | 36.3 | 63.2 | 75.2 | 138.8 |
| Residual sup. | 18.4 | 33.8 | 59.5 | 70.5 | 106.7 | 12.7 | 23.8 | 40.3 | 46.7 | 77.5 | 15.2 | 29.6 | 55.1 | 66.1 | 127.1 | 36.0 | 68.7 | 125.0 | 145.5 | 195.5 |
| convSeq2Seq | 16.7 | 30.5 | 53.8 | 64.3 | **91.5** | 8.4 | 16.2 | 30.8 | 37.8 | 76.5 | 10.6 | 20.3 | 38.7 | 48.4 | **115.5** | 22.4 | 44.0 | 87.5 | 106.3 | 162.6 |
| FC-GCN | 14.0 | 25.4 | 49.6 | 61.4 | 106.1 | 3.5 | 6.1 | 11.7 | 15.2 | 53.9 | 7.4 | **15.1** | 31.7 | 42.2 | 152.4 | 16.9 | 34.4 | 76.3 | 96.8 | 164.6 |
| LDRGCN | 13.1 | 22.0 | 37.2 | 55.8 | 97.7 | 3.4 | 6.2 | **11.2** | **13.8** | 47.3 | **6.8** | 16.3 | **27.9** | **38.9** | 131.8 | 13.2 | 32.7 | 65.1 | 91.3 | 153.5 |
| Ours | **9.1** | **16.6** | **34.7** | **44.5** | 108.4 | **3.3** | **5.9** | **11.5** | **14.7** | **44.7** | **19.6** | **31.3** | **54.8** | **66.1** | **155.5** | **12.5** | **22.7** | **44.4** | **55.8** | **120.4** |
| | ±0.7 | ±1.5 | ±3.5 | ±4.4 | ±5.1 | ±1.1 | ±2.0 | ±3.7 | ±4.7 | ±15.0 | ±16.7 | ±23.2 | ±34.4 | ±37.4 | ±52.1 | ±2.0 | ±3.8 | ±7.4 | ±9.6 | ±21.0 |
| | Running | | | | | Soccer | | | | | Walking | | | | | Wash window | | | | |
| millisecond (ms) | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 | 80 | 160 | 320 | 400 | 1000 |
| Zero velocity | 30.6 | 52.8 | 94.1 | 112.2 | 242.6 | 10.3 | 17.5 | 31.8 | 39.0 | 79.4 | 18.3 | 31.2 | 55.1 | 66.2 | 137.7 | 12.3 | 21.1 | 37.8 | 45.7 | 90.9 |
| Residual sup. | 15.6 | 19.4 | 31.2 | 36.2 | 43.3 | 20.3 | 39.5 | 71.3 | 84.0 | 129.6 | 8.2 | 13.7 | 21.9 | 24.5 | **32.2** | 8.4 | 15.8 | 29.3 | 35.4 | 61.1 |
| convSeq2Seq | 14.3 | **16.3** | **18.0** | **20.2** | **27.5** | 12.1 | 21.8 | 41.9 | 52.9 | 94.6 | 7.6 | 12.5 | 23.0 | 27.5 | 49.8 | 8.2 | 15.9 | 32.1 | 39.9 | 58.9 |
| FC-GCN | 25.5 | 36.7 | 39.3 | 39.9 | 58.2 | 11.3 | 21.5 | 44.2 | 55.8 | 117.5 | 7.7 | 11.8 | 19.4 | 23.1 | 40.2 | 5.9 | 11.9 | 30.3 | 40.0 | 79.3 |
| LDRGCN | 15.2 | 19.7 | 23.3 | 35.8 | 47.4 | 10.3 | 21.1 | 42.7 | 50.9 | 91.4 | **7.1** | **10.4** | **17.8** | **20.7** | 37.5 | 5.8 | 12.3 | 27.8 | 38.2 | **56.6** |
| Ours | **12.4** | 19.7 | 32.3 | 39.0 | 68.9 | **4.9** | **7.9** | **14.2** | **18.0** | **53.1** | 8.1 | 13.6 | 22.1 | 26.1 | **32.4** | **5.5** | **9.8** | **19.2** | **24.3** | **61.3** |
| | ±0.0 | ±0.0 | ±0.0 | ±0.0 | ±0.0 | ±0.3 | ±0.6 | ±1.2 | ±1.6 | ±4.5 | ±0.0 | ±0.0 | ±0.0 | ±0.0 | ±1.8 | ±0.8 | ±1.4 | ±2.5 | ±3.1 | ±8.4 |

TABLE 3: Motion prediction results measured with Equation 11 on CMU dataset. Short-term results are reported within 80, 160, 320, 400ms, and long-term in 1s. Best results in bold while state-of-the-art best results that fit in our confidence interval are also written bold.

in Figures 2 the evolution of the MPJS over time steps on the Human3.6M dataset.

Results show that the average ground truth speed vary slightly around 9 *mm/frame*. The zero velocity baseline obviously show the worst results as no motion is being produced and the speed is always null. Both FC-GCN and ConvSeq2Seq speeds continuously decrease over time meaning that less motion is being produced on average for long term horizon. On the other hand our method show variations in the average produced speed up to $1s$ with even an increase during long term prediction. The result from Tables 2 and 1 show that the pose error is good for long time prediction, meaning that the increase of average speed do not correspond to a degradation in the quality of the prediction. This support our claim that our method is a good fit to predict long motion that keep their spatial and temporal coherency. We report in Figure 3 the evolution of MPJS for the CMU dataset in the same way as for Human3.6M. We do not report the results for FC-GCN as there was no pretrained model available and we were unable to train their model ourselves. The results are similar to those of Human3.6M: average speed of prediction method lower than the ground truth, constant decrease for ConvSeq2Seq, more variation for our method with a significant increase in long term horizons. However we see that this time for short term prediction ConvSeq2Seq produce an average speed closer to the ground truth than ours. This seems to confirm the MPJPE results for ConvSeq2Seq that show that it performs better on CMU than on HUMAN3.6M, we still perform better for long term prediction, showing that our method is robust to different kind of datasets. Interestingly however all evaluated methods have average speeds well below the values of the ground truth. While this may be

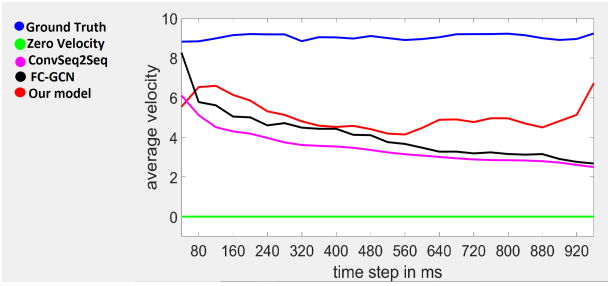



Fig. 2: The average speed (MPJS) evolution over 1000 *ms* of all action classes of the Human3.6M dataset.

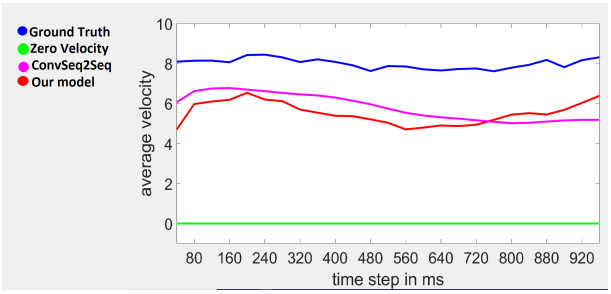



Fig. 3: The average speed (MPJS) evolution over 1000 *ms* of all action classes of the CMU MoCap dataset.

explained in part by the presence of sudden, high amplitude and hard to predict motion in action classes like Direction or Greeting, it still indicate that using losses that solely constraint the poses during training lead to generating sequences with slower motion since fast motions are more prone to error. This might be hint that using losses on the speed of the motion will help produce even better predictions.

We present in Table 4 the average MPJS for each class over all time steps on Human3.6M. In consistency with Figure 2 this table shows a significant difference between the ground truth and generated motions with all methods. However, this difference changes significantly between classes; some classes like Walking Dog or Greeting present a high difference (6.19 and 4.89 respectively when compared with our method) while other have a lower difference like Eating or Smoking (1.93 and 1.70 respectively when compared with our method). Furthermore, our method is still able to outperform ConvSeq2Seq and FC-GCN on all classes except Walking and Walking together where FC-GCN performs better indicating a capability to better model periodical motion. On the other hand for non-periodic motion our method outperform FC-GCN by a large margin (Greeting, Sitting Down, etc.).

### 5.5 Qualitative Comparison

In this part we present some examples that illustrate the smoothness of the generated motion with our method compared to the ground truth and the baselines.

In Figure 4 we present the 3D pose sequences of a predicted motion using a model trained for long term prediction with our architecture. We also show the prediction of the same 3D pose sequence by the baseline methods ConvSeq2Seq [8] and FC-GCN [9] using their publicly available code. LDRGCN [29] is not included as the code for his method is not yet available. We observe that visually our method produce a realistic and smooth motion and that our pose sequence follow more closely the ground truth than the other methods event for long term prediction. The motion produced by our method do not show any discontinuity, this is the consequence of applying the predicted dynamic of the motion to a starting pose, it prevent the discontinuity than can appear when predicting directly the 3D poses as the other methods do.

| | Ground truth | ConvSeq2Seq | FC-GCN | Ours |
|---|---|---|---|---|
| Direction | 5.97 | 2.43 | 2.39 | 3.41 |
| Discussion | 8.42 | 3.03 | 3.28 | 4.7 |
| Eating | 6.24 | 3.35 | 3.77 | 4.31 |
| Greeting | 11.54 | 3.41 | 3.82 | 6.65 |
| Phoning | 7.77 | 3.29 | 3.74 | 4.66 |
| Photo | 7.77 | 2.42 | 2.99 | 3.81 |
| Posing | 10.56 | 3.34 | 3.85 | 4.84 |
| Purchases | 10.28 | 2.60 | 3.41 | 4.97 |
| Sitting | 7.37 | 1.85 | 2.04 | 3.34 |
| Sitting Down | 9.58 | 2.50 | 2.37 | 4.53 |
| Smoking | 6.33 | 2.90 | 4.01 | 3.95 |
| Waiting | 7.98 | 3.37 | 3.56 | 4.63 |
| Walking Dog | 13.29 | 4.59 | 5.33 | 7.1 |
| Walking | 12.76 | 8.11 | 9.9 | 8.78 |
| Walking together | 9.95 | 4.95 | 6.87 | 6.59 |
| Average | 9.05 | 3.48 | 4.09 | 5.08 |

TABLE 4: Averaged MPJS over 1000 *ms* for all classes of Human3.6M dataset. Closer to the ground truth is better.

### 5.6 Motion Smoothness

In Figure 5 we show the evolution of the y coordinate from the skeleton's left foot over time and in Figure 6 the evolution of the x axis of the right hand. The 25 frames samples were selected randomly from the walking and walking together action classes respectively from the Human3.6M dataset. We see clearly in the figure that our method is able to generate a smooth motion in both cases and that we are able to follow the real motion from the ground truth, closely for the walking sample and with a small temporal delay for walking together while for this later, the other methods show a completely different movement.

### 5.7 Computation Time

We show a comparison of the computing time in Table 5 of our method with ConvSeq2Seq and FC-GCN. This time comparison is done for long term prediction (*i.e*, predicting 25 frames) with 8 sequences for each of the 15 action classes from the Human3.6M dataset using the code provided by the author for ConvSeq2Seq and FC-GCN. The results from Table 5 show that despite the additional computations required to map the motion back and forth to the tangent space compared to standard GAN architecture, we can predict motion with a speed similar to the other two methods and faster than ConvSeq2Seq.

| | total time | time per sample (25 frames) |
|---|---|---|
| ConvSeq2Seq | $3.04s$ | $\approx 25ms$ |
| FC-GCN | $1.67s$ | $\approx 14ms$ |
| Ours | $2.42s$ | $\approx 20ms$ |

TABLE 5: Prediction time comparison for 8 predicted samples per action on Human3.6M.

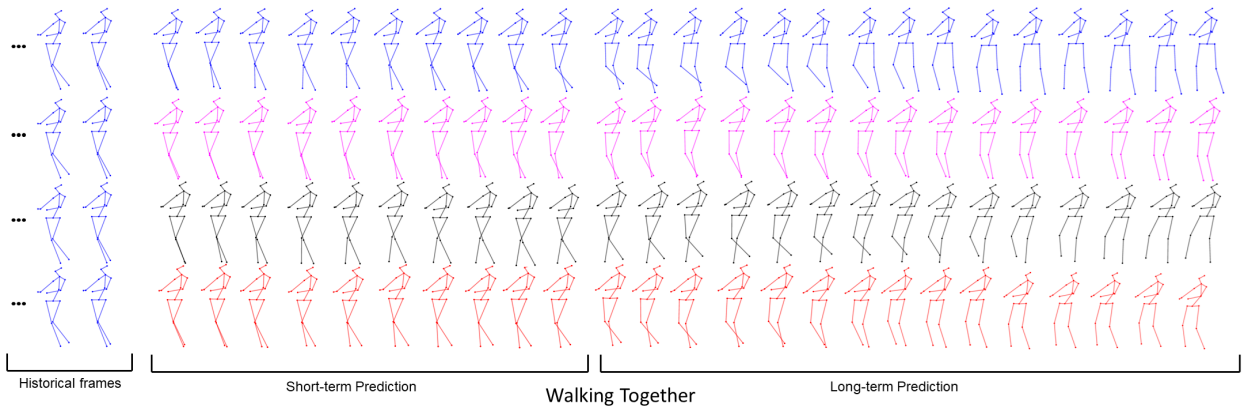



Fig. 4: The left frames correspond to the sequence used as a prior. From top to bottom : ground truth, the results of ConvSeq2Seq [8], FC-GCN [9] and our method. The illustrated action corresponds to 'Walking Together' from Human3.6M dataset. Short-term frames shown correspond to predicted frames 1, 9 and 10 and long-term frames to frames 11, 12, 22, 23, 24 and 25.

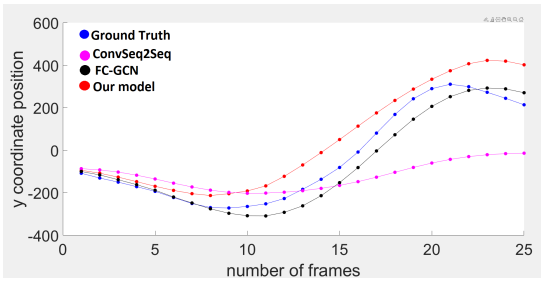



Fig. 5: Walking action from Human3.6M. X-axis and y-axis corresponds respectively to frame numbers and joint position on the y axis.

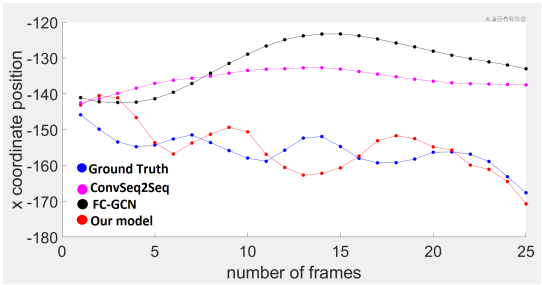
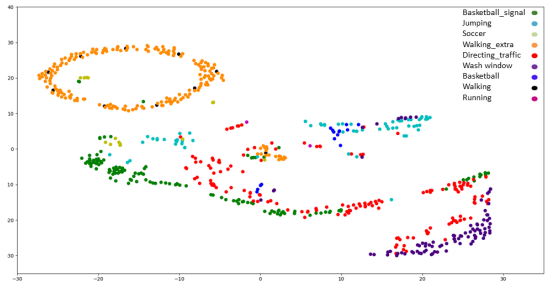
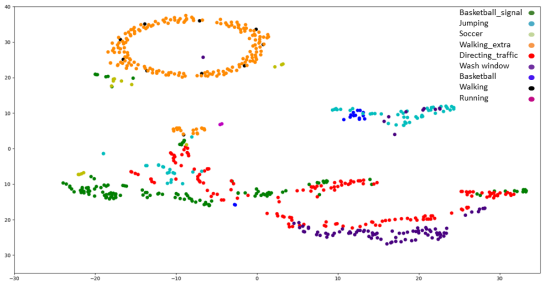



Fig. 6: Walking Together action from Human3.6M. X-axis and y-axis corresponds respectively to frame numbers and joint position on the x axis for the right hand joint.




(a) Predicted motions




(b) Ground truth motions

Fig. 7: 2D visualization of the predicted motions by our method and their associated ground truth using t-SNE algorithm based on Gram distance eq.7. Each color represents an action.

### 5.8 Distribution Visualization

With Figure 7 we further assess the quality of the predicted samples using the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [31]. We present a 2D visualization of 677 samples of long term prediction from the CMU MoCap dataset. The resulting representation clearly indicates that the motion from the ground truth and the predicted motion are from very close distributions. Moreover we can see that the different generated 3D sequences from the same action are relatively distant from each other, meaning for the same action class our model can predict several motions while respecting the prior motion used for the prediction.

### 5.9 Recursive Generation

One of the main limitation of our method is its inability to generate sequence of lengths it has not been trained on. We can however still generate longer sequences trough recursive generation by predicting subsequent motion based on previous prediction. This recursive generation can be done without specific training simply by modifying the input during testing. However, by feeding our prediction to the network to get further prediction we cause the network to accumulate error over each recursive iteration. In fact we can not reliably extend the duration of the prediction more than 2 or 3 times. For all types of motion the first and second prediction using predicted data as input are good, the third one is usually still good for periodic motion (*e.g* walking) but not for non-periodic motion (*e.g* greeting). From the fourth prediction onward even the periodic motion will start to deteriorate significantly. Non-

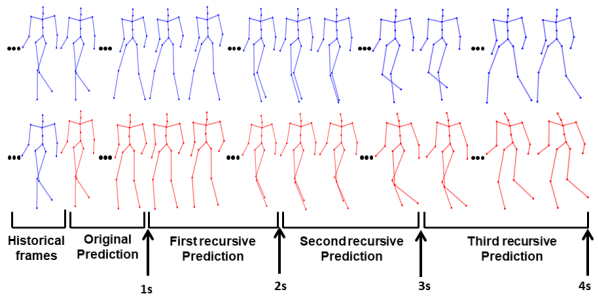



Fig. 8: Exemple of recursive prediction on sample of action class walking from the CMU MoCap Dataset for a total of 4 seconds of prediction. On top the ground truth, on the bottom our prediction.

periodic motion will usually freeze into a static pose which is to be expected as our prediction can only predict the end of the motion, not infer what other motion might follow. For periodic motion the deterioration comes from the accumulating error which will cause the skeleton to deform. Still we are able to generate motion 3 to 4 time longer than what the network was trained on, which allows us to tackle one of the limitation of the method in some ways. We present in Figure 8 an example of recursive prediction on the action class walking from the CMU MoCap dataset. The figure shows the original prediction and the three subsequent predictions. This shows that our model can predict motion for longer sequences than it has been trained on as we observe significant differences only during the third recursive prediction, the motion however still follow walking action.

### 5.10 Cross-dataset capabilities

Due to differences in the skeleton formats used by Human3.6M and CMU MoCap it is not possible to perform cross-dataset evaluation on these datasets. We have however trained our model on the NTU RGB+D 120 dataset [32] and then predicted motion from data captured with a Kinect camera in real time. Qualitative results are presented in figure 9 (in blue the ground truth and in red the prediction). We show the prediction a person rubbing its hands. We can see that our method is able to predict on data that does not come from the dataset used for training. While there is a small difference between the ground truth and our prediction we see that we do not reproduce the discontinuities from the Kinect and our prediction has not been influenced by the discontinuities in the prior

### 5.11 Ablation Study

To show the efficiency of the different losses used by our network especially the effect of the combination of the skeleton integrity loss $\mathcal{L}_s$ and the bone length loss $\mathcal{L}_b$, we perform our ablation study using model that were trained using only the mentioned losses. The ablation is performed on the Human3.6M dataset due to the huge quantity of data from the dataset. The ablation results are reported in Table 6 for short term and long term prediction using the the average error of all actions classes at different time steps. The results show a clear improvement when adding one of either the skeleton integrity loss or the bone length loss compared to using only $\mathcal{L}_a$ and $\mathcal{L}_r$. Furthermore using both $\mathcal{L}_s$ and $\mathcal{L}_b$ improve significantly the results for long term prediction while keeping a similar accuracy for short term prediction with regard to using only $\mathcal{L}_s$ or only $\mathcal{L}_b$. This evidences the importance of using both losses when doing long term prediction, it allows the model to capture the spatial dependencies between joints and to be able to predict plausible poses even for longer term horizons.

We show in Figure 10 the effect of the losses on the visual quality of the prediction. We notice that excluding $\mathcal{L}_s$ and $\mathcal{L}_b$ leads to important deformations in the upper body but the produced legs motion is rather coherent. Adding $\mathcal{L}_s$ helps produce a motion closer to the ground truth, we however still see noticeable bones deformations (better seen as animations in the supplementary material ) even if we are able to keep a coherent skeleton. Using only $\mathcal{L}_b$ leads to a skeleton without any deformation even during long time prediction but also to very little motion being produced. Using both losses allows us to keep the best skeleton coherency while producing a motion that is close to that of the ground truth.

We show in table 7 the MPJPE values for different input sequence length for long term prediction on Human3.6M. We report the values for sequences of 25 frames (default value used for comparison with the state of the art), 15 frames, 10 frames and 5 frames. We see that a shorter prior lead to a decrease in performance for both short and long term prediction but this decrease is less important for long term prediction (except for the 5 frames prior) highlighting the ability of our network to generate accurate prediction for long term motions. We observe that we can use priors of 10 or 15 frame with a moderate drop in performance but with only 5 frames the drop increase significantly especially for long term prediction.

| loss functions | 80 | 160 | 320 | 400 | 1000 |
|---|---|---|---|---|---|
| $\mathcal{L}_a + \mathcal{L}_r$ | 20.2 | 34.9 | 62.4 | 74.9 | 133.3 |
| $\mathcal{L}_a + \mathcal{L}_r + \mathcal{L}_s$ | 13.6 | 23.4 | 42.6 | 51.6 | 103.8 |
| $\mathcal{L}_a + \mathcal{L}_r + \mathcal{L}_b$ | 12.6 | 22.4 | **41.3** | **49.9** | 105.6 |
| $\mathcal{L}_a + \mathcal{L}_r + \mathcal{L}_s + \mathcal{L}_b$ | **12.3** | **22.2** | **41.3** | 50.1 | **96.2** |

TABLE 6: Impact of the bone length loss and the skeleton integrity loss on the prediction performance for short-term and long-term.

| loss functions | 80 | 160 | 320 | 400 | 1000 |
|---|---|---|---|---|---|
| 5 frames prior | 14.2 | 25.3 | 47.0 | 56.5 | 104.4 |
| 10 frames prior | 13.6 | 24.2 | 44.7 | 53.5 | 98.6 |
| 15 frames prior | 13.3 | 23.6 | 43.7 | 52.4 | 96.8 |
| 25 frames prior | **12.3** | **22.2** | **41.3** | **50.1** | **96.2** |

TABLE 7: Impact of the prior length on long term prediction

## 6 Conclusion and Limitations

In this paper we presented a new and robust method to deal with human motion prediction. In our method we represent the temporal evolution of 3D human poses as trajectories, these trajectories can be mapped to points on a hypersphere. To be able to learn learn this manifold-valued representation we use a manifold-aware Wasserstein GAN that can capture both the spatial and temporal dependencies involved in human motion. Through extensive experiments we prove the robustness of our method for long term motion prediction when compared to recent literature. With our qualitative results we confirm that we are able to predict plausible poses and smooth motions in long term horizons.

Predicting human gait is a nice application of our predictive networks. Our results on these classes are promising and we believe we would obtain good results on 3D gait databases. The two main limitations of the proposed method are the following: the fixed length of sequence and the inability to deal with sudden

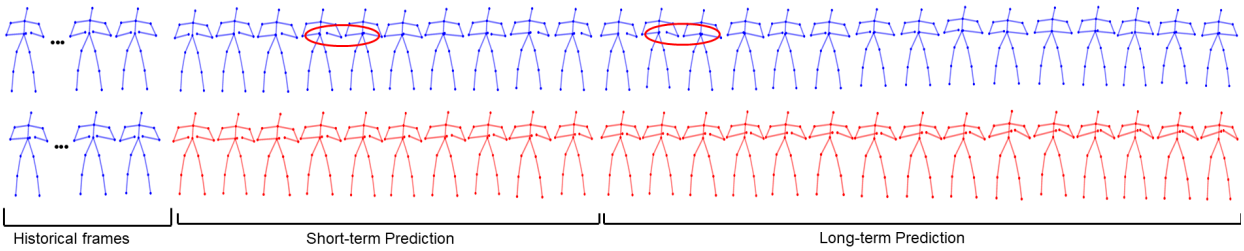



Fig. 9: Cross database results from NTU RGB+D to Kinect real time capture. In blue the ground truth and in red the prediction. The discontinuities from the Kinect camera are highlighted in red

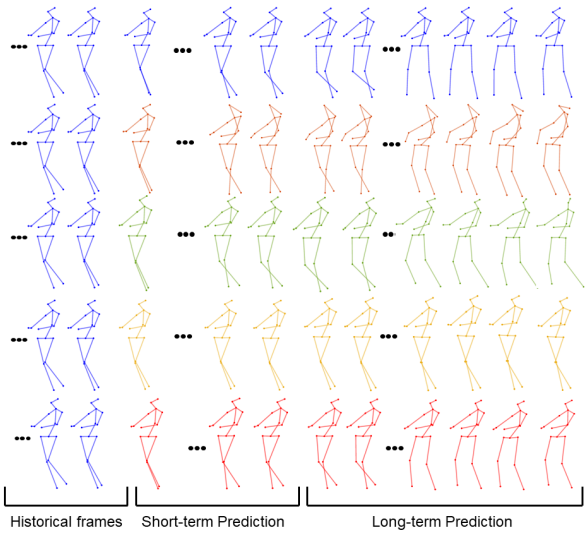



Fig. 10: Impact of the bone length loss and the skeleton integrity loss on prediction quality on a sample from action class Walking together from Human3.6M. From top to bottom: the ground truth, neither $\mathcal{L}_s$ nor $\mathcal{L}_b$, only $\mathcal{L}_s$, only $\mathcal{L}_b$ and both $\mathcal{L}_s$ and $\mathcal{L}_b$

changes in motion. The fixed length in motion is a consequence of the GAN architecture where the input and output sizes a fixed. We demonstrate in our experiment that we can deal with this problem by using recursive generation that show the ability of the model to generate up to 4$s$ motion for some classes when trained on 1$s$ sequences. The inability to deal with a sudden change of motion is inherent to the way motion prediction is usually approached. Indeed, we only consider the historical motion as a condition to predict the motion but it is not always enough to get an accurate prediction. Things like the environment, the goal of the motion and the motion of other persons can influence the future. Taking some of these modalities into account would surely allows for longer and more accurate predictions.

## ACKNOWLEDGMENTS

This project has received financial support from the CNRS through the 80—Prime program and from the French State, managed by the National Agency for Research (ANR) under the Investments for the future program with reference ANR- 16-IDEX-0004 ULNE. This work has been supported by "La Fédération de Recherche Sciences et Cultures du Visuel" (FR CNRS 2052), and by a State grant managed by the National Research Agency under the Programme d'Investissements d'Avenir with the reference ANR-21-ESRE-0030 - Equipex+ Continuum. Most of this work was done when Naima Otberdout was at University of Lille.

## REFERENCES

[1] H. S. Koppula and A. Saxena, "Anticipating human activities for reactive robotic response," in *IROS*, 2013, pp. 2071–2071.
[2] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *T-IV*, vol. 1, no. 1, pp. 33–55, 2016.
[3] L. Kovar, M. Gleicher, and F. H. Pighin, "Motion graphs," in *ACM SIGGRAPH Classes*, 2008, pp. 51:1–51:10.
[4] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *ICCV*, 2015, pp. 4346–4354.
[5] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *CVPR*, 2016, pp. 5308–5317.
[6] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," in *3DV*, 2017, pp. 458–466.
[7] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *CVPR*, 2017, pp. 4674–4683.
[8] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional Sequence to Sequence Model for Human Dynamics," in *CVPR*, 2018, pp. 5226–5234.
[9] M. Wei, L. Miaomiao, S. Mathieu, and L. Hongdong, "Learning trajectory dependencies for human motion prediction," in *ICCV*, 2019, pp. 9488–9496.
[10] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *CVPR*, 2017, pp. 6158–6166.
[11] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *ICCV*, 2019, pp. 9488–9496.
[12] S. Berretti, M. Daoudi, P. K. Turaga, and A. Basu, "Representation, analysis, and recognition of 3d humans: A survey," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 14, no. 1s, pp. 16:1–16:36, 2018. [Online]. Available: https://doi.org/10.1145/3182179
[13] B. Chopin, N. Otberdout, M. Daoudi, and A. Bartolo, "Human motion prediction using manifold-aware wasserstein gan," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 2021, pp. 1–8.
[14] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. F. Moura, "Adversarial Geometry-Aware Human Motion Prediction," in *ECCV*, 2018, pp. 823–842.
[15] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3D human motion prediction via gan," in *CVPR Workshops*, 2018, pp. 1418–1427.
[16] Z. Huang, J. Wu, and L. Van Gool, "Manifold-valued image generation with wasserstein generative adversarial nets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3886–3893.
[17] P. K. Turaga and R. Chellappa, "Locally time-invariant models of human activities using trajectories on the Grassmannian," in *CVPR*, 2009, pp. 2435–2441.
[18] B. Ben Amor, J. Su, and A. Srivastava, "Action recognition using rate-invariant analysis of skeletal shape trajectories," *PAMI*, vol. 38, no. 1, pp. 1–13, 2016.
[19] A. Kacem, M. Daoudi, B. Ben Amor, S. Berretti, and J. C. Álvarez Paiva, "A novel geometric framework on Gram matrix trajectories for human behavior understanding," *PAMI*, vol. 42, no. 1, pp. 1–14, 2020. [Online]. Available: https://doi.org/10.1109/TPAMI.2018.2872564
[20] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE TC*, vol. 45, no. 7, pp. 1340–1352, 2014.

[21] N. Otberdout, M. Daoudi, A. Kacem, L. Ballihi, and S. Berretti, "Dynamic facial expression generation on hilbert hypersphere with conditional wasserstein generative adversarial nets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 848–863, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2020.3002500
[22] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in euclidean spaces," *PAMI*, vol. 33, no. 7, pp. 1415–1428, 2011. [Online]. Available: https://doi.org/10.1109/TPAMI.2010.184
[23] H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama, "3D face recognition under expressions, occlusions, and pose variations," *PAMI*, vol. 35, no. 9, pp. 2270–2283, 2013.
[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *NIPS*, 2017, pp. 5767–5777.
[25] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on pure and applied mathematics*, vol. 30, no. 5, pp. 509–541, 1977.
[26] A. Srivastava and E. P. Klassen, *Functional and Shape Data Analysis*. Springer, New York, NY, 2016.
[27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. The Johns Hopkins University Press, 1996.
[28] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *PAMI*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6682899/
[29] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3D human motion prediction," in *CVPR*, 2020.
[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[31] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *JMLR*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html
[32] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020.

**Baptiste Chopin** received the Engineering degree in computer science from IMT Nord Europe (France). He is currently pursuing the Ph.D. degree with the university of Lille (France). His research concern computer vision and the generation of human motion with application to cognitive sciences.

**Naima Otberdout** received the master's degree in computer sciences and telecommunication from Mohammed V University, Rabat, Morocco in 2016. She received the Ph.D. degree in computer science from the same university in 2021. After a post-doctoral position in the University of Lille in France, she is currently a research and education fellow in Ai movement - University Mohammed IV Polytechnic in Morocco.
Her current research interests include computer vision and pattern recognition with applications to human behavior understanding.

**Mohamed Daoudi** is Full Professor of Computer Science at IMT Nord Europe and the lead of Image group at CRIStAL Laboratory (UMR CNRS 9189). He received his Ph.D. degree in Computer Engineering from the University of Lille (France) in 1993. His research interests include pattern recognition, shape analysis and computer vision. He has published over 150 papers in some of the most distinguished scientific journals and international conferences. He is Associate Editor of Image and Vision Computing Journal, IEEE Trans. on Multimedia, Computer Vision and Image Understanding, IEEE Trans. on Affective Computing and Journal of Imaging. He has served as General Chair of IEEE International Conference on Automatic Face and Gesture Recognition, 2019. He is Fellow of IAPR and IEEE Senior member.

**Angela BARTOLO** is Full Professor of Neuropsychology of Motor Cognition at the University of Lille and Dean of the Faculty of Psychology, Educational and Formation Sciences of the same university. She is co-responsible of the European Master in Psychology of Neurocognitive Processes and Affective Sciences. She received her Ph.D. degree in Science from the University of Aberdeen (UK) in 2002 and was subsequently recruited as post doc researcher at the University of Toronto (Canada), at the University of Modena and Reggio Emilia (Italy) and at the University of Lille (France). She is member of the Laboratory SCALab (UMR CNRS 9193) and scientific coordinator of the ESTRA International Association Laboratory. Her research interests focus on the processing of manual actions, action semantic and on the relation between action and social cognition in normal population and in brain damage patients, these issues are investigated by means of behavioural and neuroimgaing techniques. She has published over 70 scientific outputs in some of the most distinguished scientific journals. She is a former Junior Member of the Institut Universitaire de France (IUF).