Human Motion Prediction Using Manifold-Aware Wasserstein GAN

Baptiste Chopin, Naima Otberdout, Mohamed Daoudi, Angela Bartolo

Human Motion Prediction Using Manifold-Aware Wasserstein GAN

Baptiste Chopin[1], Naima Otberdout[1], Mohamed Daoudi[2,3], Angela Bartolo[4]
[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[2] IMT Lille Douai, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France
[3] Univ. Lille, CNRS, Centrale Lille, Institut Mines-Télécom, UMR 9189 CRIStAL, F-59000 Lille, France
[4] Univ. Lille, CNRS, UMR 9193 SCALab, F-59000 Lille, France

***Abstract*— Human motion prediction aims to forecast future human poses given a prior pose sequence. The discontinuity of the predicted motion and the performance deterioration in long-term horizons are still the main challenges encountered in current literature. In this work, we tackle these issues by using a compact manifold-valued representation of human motion. Specifically, we model the temporal evolution of the 3D human poses as trajectory, what allows us to map human motions to single points on a sphere manifold. 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. Extensive experiments show that our approach outperforms the state-of-the-art on CMU MoCap and Human 3.6M datasets. Our qualitative results show the smoothness of the predicted motions. The pretrained models and the code are provided at the following link.**

I. INTRODUCTION

The problem of predicting future human motion is at the core of many applications in computer vision and robotics, such as human-robot interaction [19], autonomous driving [25] and computer graphics [20]. In this paper, we are interested in building predictive models for short-term and long-term future 3D poses of a skeleton based on an initial history. Addressing this task gives rise to two major challenges: How to model the temporal evolution of the motion to ensure the smoothness of the predicted sequences? and how to take into consideration the spatial correlations between human joints to avoid implausible poses? Given the temporal aspect of the problem, human motion prediction was widely addressed with Recurrent Neural Networks (RNN) [8], [15], [9], [23]. However, while RNN based methods achieved good advance in term of accuracy, it was observed that the predicted motions present significant discontinuities due to the frame-by-frame regression process that discourage the global smoothness of the motion. Besides, RNNs models accumulate errors across time, which results in large error and bad performance in long-term prediction. As a remedy, more recent works avoid these models and explore feed-forward networks instead. Including CNN [21], GNN [29] and fully-connected networks [4], the hierarchical structure of feed-forward networks can better handle the spatial dependencies of human joints than RNNs. Nevertheless, these models require an additional strategy to encode the temporal information. To meet this challenge, an interesting idea was to model the human motion as trajectory [22], [3].

In this paper, we follow the idea of considering motions as trajectories but in a different context from the previous work. Among the advantages of our representation, the possibility to map these trajectories to single compact points on a manifold, which helps with the smoothness and the continuity of the predicted motions. In addition, the compact representation avoids the accumulation of errors through time and makes our method powerful for long-term prediction as illustrated in Figure 2. However, the resulting representations are manifold-valued data that cannot be handled with traditional generative models in a straightforward manner. To meet this challenge, we propose a manifold-aware Wasserstein Generative Adversarial Networks (WGAN) that anticipate future poses based on the input manifold-valued data that encodes the prior motion sequence. Our model incorporates the spatial dependencies between human joints through different loss functions that insure the plausibility of the predicted poses. A brief overview of our prediction process is illustrated in Figure 1.

Main contributions. The paper gives rise to the following contributions: (1) To the best of our knowledge, this is the first 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 manifold-aware WGAN for motion prediction. (3) Experimental results on Human 3.6M and the CMU MoCap datasets show quantitatively and visually the effectiveness of our method for short-term and long-term prediction.

II. RELATED WORK

Human Motion Prediction with Deep Learning. Since the problem of human motion prediction is a temporal dependent task, recurrent models were the first potential solution to be investigated, thus many works applied RNN and their variants to address this task. In [8], the authors proposed a model that incorporates a nonlinear encoder and decoder before and after recurrent layers. Their approach was limited by the problem of error accumulation. Besides, they only capture the temporal dependencies while ignoring

This project has received financial support from the CNRS through the 80—Prime program. This work was also partially supported by the French State, managed by National Agency for Research (ANR) National Agency for Research (ANR) under the Investments for the future program with reference ANR-16-IDEX-0004 ULNE.

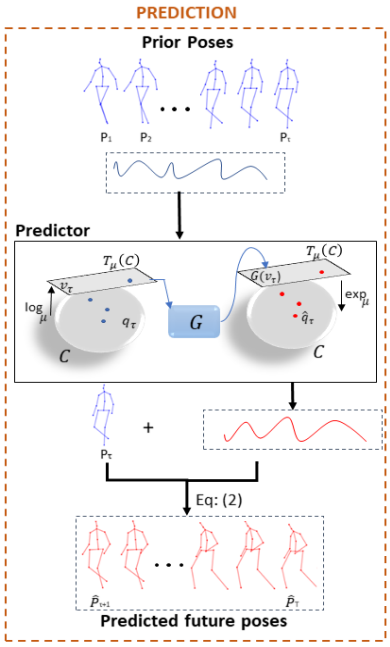


Fig. 1: Overview of the human motion prediction process. Given a pose sequence history represented as a curve, then mapped to a single point in a hypersphere. The predictor maps the input point to a tangent space, then feeds it to the network $\mathcal{G}$ that predicts the future motion as a vector in $T_\mu(\mathcal{C})$. 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.

the spatial correlations between joints. To solve this issue, [15] proposed a Structural-RNN model relying on high-level spatio-temporal graphs. In an other direction, to reduce the effect of the error accumulation in recurrent models, [9] used a feed forward network for pose filtering and a RNN for temporal filtering. However, this strategy only reduces the accumulation of the error which still exists and affects the performance of recurrent models. Taking a different direction, more recent works use feed-forward networks as an alternative model. To represent the temporal evolution with these models, different strategies were proposed. In [21], [4], convolution across time was adopted to model the temporal dependencies with convolution networks, while [22] exploit Discrete Cosine Transform to encode the motion as trajectory.

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): Human motion prediction was also addressed with GANs in [11] and [1], however, their generator is based on RNN structures to deal with the temporal aspect of this task. By doing so, their models keep the problem of error accumulation which may affect their 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, [13] 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 [13] 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. Besides, 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 [27], [2], [16]. More related to our work, in [6], 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.

III. 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, [29], 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.

A. 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 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 τ 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 $\{P_1, P_2, \ldots, P_T\}$.

B. 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 [26]. It was successfully exploited for human action recognition [6], 3D face recognition [7] and facial expression generation [24]. 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

$$q(t) = \frac{\dot{\alpha}(t)}{\sqrt{\|\dot{\alpha}(t)\|}} , \tag{1}$$

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 [26]. 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:

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

IV. 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 τ 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 τ 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. [13], 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.

A. 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_τ, 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_τ 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 3×3 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 3×3 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.

B. 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_τ as input rather than a random vector as done in [13], 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 $\log_\mu(.)$ and $\exp_\mu(.)$ are the logarithm and exponential maps on the sphere, used to iteratively map the SRVF data back and forth to the tangent space $T_\mu(C)$ at a reference point μ. They are given by:

$$\begin{aligned}\log_\mu(q) &= \frac{d_\mathcal{C}(q,\mu)}{sin(d_\mathcal{C}(q,\mu))}(q-cos(d_\mathcal{C}(q,\mu))\mu)\,,\\ \exp_\mu(s) &= cos(\|s\|)\mu+sin(\|s\|)\frac{s}{\|s\|},\end{aligned} \tag{5}$$

where $d_\mathcal{C}(.,.)$ is the geodesic distance defined by (3). The last term of $\mathcal{L}_a$ represents the gradient penalty proposed in [12]. $\widetilde{q}$ 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 μ of the tangent space used in our training is set to the mean of the training data. It is given by the Karcher mean [17] in $\mathcal{C}$, $\mu = \underset{q_i\in\mathcal{C}}{\operatorname{argmin}}\sum_{i=1}^m d_\mathcal{C}^2(\mu,q_i)$, where $\{q_i\}_{i=1}^m$ is m training data.

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 [10, p. 328] as:

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

where $\operatorname{tr}(.)$ denotes the trace operator, and $\{\sigma_i\}_{i=1}^3$ are the singular values of $P_j^TP_i$. 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, τ 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 β_i are the coefficients associated to different losses, they are set empirically in our experiments.

The algorithm 1 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.

V. Experiments

In order to evaluate the proposed approach, we performed extensive experiments on two commonly used datasets. In what follows, we present and discuss our results.

A. Datasets and Pre-processing

Human 3.6M. Human 3.6M [14] has 11 subjects in 15 various actions (Eating, Walking, Taking photos…). It is the largest dataset and the most commonly used for human motion prediction with 3D skeletons in literature. As previous works [23], [5], our models are trained on 6 subjects and tested on the specific clips of the 5th subject. Following [5] we use only 17 joints out of 32; the removed joints correspond to duplicate joints, hands and feet.

CMU Motion Capture (CMU MoCap). CMU Mocap dataset [1] consists of 5 categories, each containing several actions. To be coherent with [21], we choose 8 actions: 'basketball', 'basketball signal', 'directing traffic','jumping', 'running', 'soccer', 'walking' and 'washing window'. We use the same joint configuration and pre-processing as for Human3.6M.

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

Algorithm 1: PredictiveMAWGAN algorithm

// Training

Data: $\{q_\tau^i\}_{i=1}^m$: SRVFs of training prior poses, $\{q_T^i\}_{i=1}^m$: real future poses, θ_0 : initial parameters of $\mathcal{G}$, η_0 : initial parameters of $\mathcal{D}$, ϵ: learning rate, K: batch size, λ: balance parameter of gradient penalty, ζ: iterations number.

Result: θ: 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: θ: 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_τ 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 θ;

Transform $\hat{q}_T$ into pose sequence $\{\hat{P}_i\}_{i=\tau+1}^T$ using Eq. 2, with $\alpha(0) = P_\tau$

B. Implementation Details

Our method is implemented using Tensorflow 2.2 on a PC with two 2.3Ghz processors, a Nvidia Quadro RTX 6000 GPU and 64Go of RAM. The models are trained using the Adam optimizer [18]. The batch size is set to 64 and the number of epochs is fixed to 500. The learning rate is fixed to 10^{-4}. The loss coefficients β_1, β_2, β_3 and β_4 are respectively set to 1, 1, 10 and 10.

C. Evaluation Metrics and Baselines

We compare our results with state-of-the-art motion prediction methods that were based on 3D coordinate representation, including RNN based method (Residual sup). [23], CNN based method (ConvSeq2Seq) [21] and graph models; (FC-GCN) [29] and (LDRGCN) [5]. We also compare with a simple baseline, Zero velocity introduced by [23], which sets all predictions to be the last observed pose at $t = \tau$. For LDRGCN we present the results reported by the authors for the method trained with data in 3D coordinate space. For FC-GCN, ConvSeq2Seq and Residual sup., we present the results reported by [29] with the methods that use 3D coordinate data for training. For the long-term (1000ms) on Human 3.6M, we use the results presented by [5] since they are not provided in [29]. We do not present the long-term results for Residual sup. on Human 3.6M as they are not available.

Following the state-of-the-art [5], our quantitative evaluation is based on the Mean Per Joint Position Error (MPJPE) [14] in millimeter. This metric compares the predicted motions and their corresponding ground-truths in the 3D coordinate space. It is given by,

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

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, τ the number of frames in prior sequence and Δt the number of predicted frames at which the sequence is evaluated.

D. Quantitative Comparison

In consistency with recent work, we report our results in short-term and long-term prediction. Given 10 prior poses, 10 future frames are predicted within 400ms in short-term, while 25 frames are predicted in 1s for long-term prediction based on the previous 25 frames. In Table I, we compare our results with recent methods based on 3D joint coordinates representation. This latter, has been proven in [29] to provide a reliable comparison in contrast to the angle based representation. The table shows a clear superiority of our approach over the state-of-the-art for both Human3.6M and CMU-MoCap datasets. We highlight that our approach in 80ms and 160ms is very competitive with LDRGCN approach, while in longer horizons we outperform this method in 320ms, 400ms and 1s, which demonstrates the robustness of our method in predicting motions that are closer to the ground-truth in long-term.

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

We further report in Table II and III, our results and those of the literature on all actions of Human3.6M and CMU

MoCap datasets, respectively. The protocol adopted by the baseline methods is to report the average error on eight randomly sampled test sequences. However, we found that the error is significantly affected by this random sampling, which makes it difficult to present a fair comparison. To alleviate this issue, we report the mean error obtained over 100 runs; in each run, we randomly sample 8 test sequences. Hence we report the average error as well as the standard deviation obtained with our model. Indeed, the standard deviation allows us to better measure the general performance of our model on different samples. The large variance obtained for some classes (*e.g,* jumping). is due to the high diversity of samples corresponding to these classes in the training data, while the other classes (*e.g,* walking) are present with less variability and then show less variance. According to Tables II and III, our approach outperforms the state-of-the-art especially for long-term prediction, which is consistent with the average error over all actions. Our results show also that the simple zero-velocity baseline outperforms the state of art in long-term for some actions (*e.g,* Photo, Sitting and Walking dog for Human3.6H, Soccer and Jumping for CMU MoCap), while in short-term, zero-velocity baseline error is generally higher. This evidences that the performance of the compared approaches decrease over time, while ours is more robust in long-term horizons, performing better than both the literature and the zero velocity baseline overall.

E. Qualitative Comparison

We show in Figure 2, 3D pose sequences of a predicted motion using our trained model for long-term prediction. We show also the predicted 3D poses of the same sequence obtained with the baseline methods ConvSeq2Seq [21] and FC-GCN [29], based on their publicly available codes. We did not include LDRGCN [5] in this comparison since their code is not yet available. Visually, we observe that our method produces a realistic pose sequences with a smooth motion that follows the ground truth more closely than the other methods even for long-term prediction. Our method does not show any discontinuity as a consequence of predicting the dynamic of the motion then applying it to a starting pose rather that directly predicting the pose sequence as the other methods do. We provide more qualitative comparison as well as video comparisons with our supplementary materials.

F. Smoothness of the motion

In order to quantitatively assess the smoothness of our predicted motions, we report in table IV, the average euclidean distance between consecutive frames for our method against the ground truth data for some actions of the CMU MoCap dataset over all frames (25), all joints (17) and all samples from the given action (variable). The results demonstrate that the generated movements are characterized by changes in time that are close to those shown in real videos. The Fig 3 shows the evolution over time of the y coordinate from the skeleton's left foot on a random sample of 25 frames from the walking action from the Human3.6M dataset. The motion

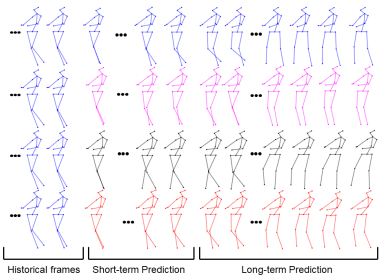


Fig. 2: The left frames correspond to the sequence used as a prior. From top to bottom : ground truth, the results of ConvSeq2Seq [21], FC-GCN [29] and our method. The illustrated action corresponds to 'Walking Together' from Human3.6M dataset. Short-term frames shown correspond to predicted frames 1, 9 and 10 and long-term frames to frames 11, 12, 22, 23, 24 and 25.

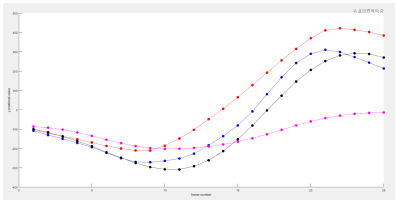


Fig. 3: Walking action from Human3.6M. In blue the ground truth, in red the sequence generated by our model, in magenta ConvSeq2Seq [21] and in black FC-GCN [29] , x-axis and y-axis corresponds respectively to frame numbers and joint position on the y axis.

of the selected action can be observed in the video in the supplementary materials. We represent the ground truth in blue, the sample generated by our model in red and the sample generated by ConvSeq2Seq [21] and FC-GCN [29] in magenta and red respectively. We can see that our method produces a smooth motion that follows the motion of the ground truth.

G. Computation Time

In Table V, we compare the computation time required by our method for long-term prediction with that of ConvSeq2Seq and FC-GCN. The time was obtained by predicting the long-term motion (*i.e*, 25 frames) of 8 sequences for each of the 15 actions from Human3.6M dataset. It is worthy to note that the codes used for ConvSeq2Seq and FC-GCN are provided by their authors. The results of the table show that regardless of the additional computations required to map the motion back and forth to the tangent space w.r.t standard GAN models, our prediction time is similar to those of the two other methods and even faster than ConvSeq2Seq.

H. Visualization

To further assess the quality of the predicted samples, we present, in Figure 4 a 2D visualization of 677 long-term prediction samples from the CMU MoCap dataset predicted with our model using the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm [28]. This figure clearly evidences that the predicted motions and their ground truth belong to very close distributions. Furthermore, the

	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 II: Motion prediction results measured with eq.11 for all actions in the Human 3.6M dataset for short-term 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 III: Motion prediction results measured with eq.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.

	Ground truth	Generated samples
Basketball signal	2.33	1.94
Running	12.71	10.96
Walking	6.86	6.62
Wash Window	5.08	4.24

TABLE IV: Averaged Euclidean distance between consecutive frames for all joints and time-steps.

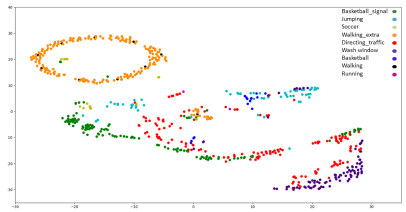
	total time	time per sample (25 frames)
ConvSeq2Seq	$3.04s$	$\approx 25ms$
FC-GCN	$1.67s$	$\approx 14ms$
Ours	$2.42s$	$\approx 20ms$

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

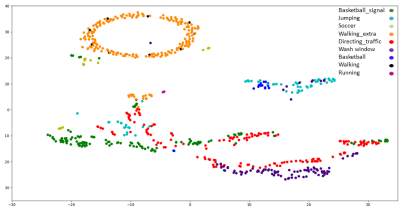
predicted 3D sequences of the same action are relatively distant from each other, what shows that our model can predict different motions for the same action while respecting the action of the given prior poses.

I. Ablation Study

In order to show the efficiency of the skeleton integrity loss $\mathcal{L}_s$ and the bone length loss $\mathcal{L}_b$ on the prediction results, we perform an ablation analysis on different models trained with different losses. We chose Human 3.6M to conduct this



(a) Predicted motions



(b) Ground truth motions

Fig. 4: 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.

study motivated by the huge data it provides. In Table VI, we report our results for short-term and long-term using the average error over all actions at different time steps. These results show a clear improvement when adding one of the losses, either $\mathcal{L}_s$ or $\mathcal{L}_b$, to the model that use only $\mathcal{L}_a$ and $\mathcal{L}_r$. Furthermore, while we obtain similar results for short-term prediction when using both losses (*i.e.*, $\mathcal{L}_s$ and $\mathcal{L}_b$) or only $\mathcal{L}_b$, we notice a remarkable enhancement for long-term prediction when adding both $\mathcal{L}_b$ and $\mathcal{L}_s$ to the objective function over the model that add only one of them. This evidences the importance of integrating both losses $\mathcal{L}_s$ and $\mathcal{L}_b$ to capture the spatial correlations between joints and keep predicting plausible poses in the long-term horizons.

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

VI. Conclusions

In this paper, we have introduced a novel and robust method to deal with human motion prediction. We have represented the temporal evolution of 3D human poses as trajectories that can be mapped to points on a hypersphere. To learn this manifold-valued representation, a manifold-aware Wasserstein GAN that captures both the temporal and the spatial dependencies involved in human motion, has been proposed. We have demonstrated through extensive experiments the robustness of our method for long-term prediction compared to recent literature. This has been confirmed also by our qualitative results that show the ability of the method to produce smooth motions and plausible poses in long-term horizons. As a limitation, our model is restricted by the fixed number of frames that can predict at a time, which corresponds to the number of frames used during training. However, our model can be used in iterative prediction up to 2 seconds for non-periodic actions (e.g., wash windows) and for longer time for periodic ones (e.g., walking).

VII. 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.

References

[1] E. Barsoum, J. Kender, and Z. Liu. Hp-gan: Probabilistic 3D human motion prediction via gan. In *CVPR Workshops*, pages 1418–1427, 2018.

[2] B. Ben Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *PAMI*, 38(1):1–13, 2016.

[3] 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.*, 14(1s):16:1–16:36, 2018.

[4] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom. Deep representation learning for human motion prediction and classification. In *CVPR*, pages 6158–6166, 2017.

[5] Q. Cui, H. Sun, and F. Yang. Learning dynamic relationships for 3D human motion prediction. In *CVPR*, 2020.

[6] 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*, 45(7):1340–1352, 2014.

[7] H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama. 3D face recognition under expressions, occlusions, and pose variations. *PAMI*, 35(9):2270–2283, 2013.

[8] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354, 2015.

[9] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning human motion models for long-term predictions. In *3DV*, pages 458–466, 2017.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, fourth edition, 1996.

[11] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. F. Moura. Adversarial Geometry-Aware Human Motion Prediction. In *ECCV*, pages 823–842. 2018.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *NIPS*, pages 5767–5777, 2017.

[13] Z. Huang, J. Wu, and L. V. Gool. Manifold-valued image generation with Wasserstein generative adversarial nets. In *AAAI*, pages 3886–3893, 2019.

[14] 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*, 36(7):1325–1339, July 2014.

[15] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *CVPR*, pages 5308–5317, 2016.

[16] 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*, 42(1):1–14, 2020.

[17] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] H. S. Koppula and A. Saxena. Anticipating human activities for reactive robotic response. In *IROS*, pages 2071–2071, 2013.

[20] L. Kovar, M. Gleicher, and F. H. Pighin. Motion graphs. In *ACM SIGGRAPH Classes*, pages 51:1–51:10, 2008.

[21] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee. Convolutional Sequence to Sequence Model for Human Dynamics. In *CVPR*, pages 5226–5234, 2018.

[22] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9488–9496, 2019.
[23] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 4674–4683, 2017.
[24] N. Otberdout, M. Daoudi, A. Kacem, L. Ballihi, and S. Berretti. Dynamic facial expression generation on hilbert hypersphere with conditional Wasserstein generative adversarial nets. *PAMI*, pages 1–1, 2020.
[25] 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*, 1(1):33–55, 2016.
[26] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *PAMI*, 33(7):1415–1428, 2011.
[27] P. K. Turaga and R. Chellappa. Locally time-invariant models of human activities using trajectories on the Grassmannian. In *CVPR*, pages 2435–2441, 2009.
[28] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9(86):2579–2605, 2008.
[29] M. Wei, L. Miaomiao, S. Mathieu, and L. Hongdong. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9488–9496, 2019.