Prediction of Human Whole-Body Movements with AE-ProMPs

Oriane Dermy, Maxime Chaveroche, Francis Colas, François Charpillet, Serena Ivaldi

Prediction of Human Whole-Body Movements with AE-ProMPs

Oriane Dermy[1], Maxime Chaveroche[1,2], Francis Colas[1], François Charpillet[1], Serena Ivaldi[1]

***Abstract*— The ability to predict the future intended movement is crucial for collaborative robots to anticipate the human actions and for assistive technologies to alert if a particular movement is non-ergonomic and potentially dangerous for the human health. In this paper, we address the problem of predicting the future human whole-body movements given early observations. We propose to predict the continuation of the high-dimensional trajectories mapped into a reduced latent space, using autoencoders (AE). The prediction is based on a probabilistic description of the movement primitives (ProMPs) in the latent space, which notably reduces the computational time for the prediction to occur, and hence enables to use the method in real-time applications. We evaluate our method, named AE-ProMPs, for predicting future movements belonging to a dataset of 7 different actions performed by a human, recorded by a wearable motion tracking suit.**

I. Introduction

An important skill that allows humans to collaborate efficiently is their ability to predict the future movement of their partners [1]. This ability not only entails the "prediction of intention", often formalized as predicting the goal of an action, but the "prediction of the future intended movement", that we recently formalized as predicting the future trajectory given early observations of it [2].

The ability to predict the future intended movement is also crucial for collaborative robots to anticipate the human actions and for assistive technologies to alert if a particular movement is non-ergonomic and potentially dangerous for the human health [3]. To consequently act, this prediction must be very fast from the few available observations, despite the variability and high-dimensionality of the movements.

In our previous work, we used *Probabilistic Movement Primitives* (ProMPs) to learn trajectory distributions of robotic actions and to predict the future intended movement during human-robot interaction. We showed that a robot can use an initial portion of a trajectory, that we call "partial trajectory", to infer its continuation up to the goal [2]. The trajectories were demonstrated to the robot using physical interaction, visual cues or both [4]. These experiments concerned only the robot's arm movements, though combining kinematics and dynamics signals.

In this paper, we are interested in predicting the future outcome of human whole-body movements, given early observations or partial trajectories, and fast enough for a robot to plan a suitable assistive action if needed. Since we want to predict the future trajectories for all the human segments performing the action, our prediction is performed in a high dimensional space and our previous method [2] is computationally inefficient (as we will show in Section V-A), hence not suitable for our real-time application. To solve this issue, we propose here to reduce the dimensionality of the data space. The high-dimensional trajectories are mapped into a low-dimensional latent space (LS). Then, the ProMPs are learned directly in this LS, in which we also compute the predicted future trajectories. The compression is done using *autoencoders* (AE), which enable *encoding* the original trajectories into the LS and *decoding* the predicted trajectories from the LS to the original high-dimensional space. We call this method *AE-ProMPs*.

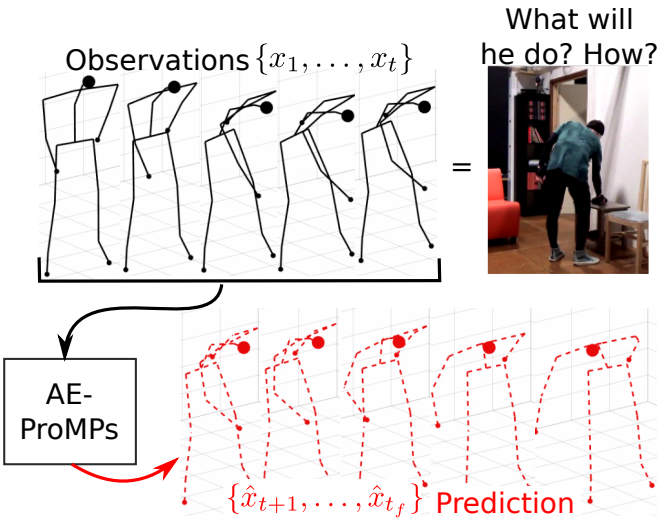


Fig. 1: Concept of our work: the goal is to predict at time t the future human whole-body movement $(t+1, \ldots, t_f)$ given t partial observations of this movement. AE-ProMPs is used to make this prediction.

We evaluate AE-ProMPs for predicting the future movements of a human performing 7 different whole-body movements, included in the dataset of [5], [6]. This scenario is represented in Figure 1. The movements were recorded by a wearable motion tracking suit (XSens MVN) [7], which provides a kinematic reconstruction of the human posture.

AE-ProMPs is computationally efficient and suitable for our application. We compare it with similar methods proposed to encode whole-body movements in a latent space, namely VAE-DMP [8] and *VTSFE* [5]. The first exploits variational autoencoders (VAE) to compress the movement in a reduced latent space, then forces the continuity of the latent space

*This work was supported by the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement No. 731540 (project AnDy).

[1] Inria, CNRS, University of Lorraine, Loria, UMR 7503 name.surname@inria.fr

[2] Heudiasyc, UTC

trajectories using *Dynamic Motion Primitives* (*DMPs*). The second is an improvement of VAE-DMP, notably by removing the dependence of the attractor in the *DMP* and by providing a lower bound for the variational inference. While both these methods are very interesting for their capacity to produce a coherent latent space that preserves the continuity of the trajectories, they are computationally expensive. Moreover, the complexity induced by the dynamic forcing function can be skipped in our case since we do not consider individual latent space trajectories, as in [8], but learn probability distributions over the latent space trajectories.

The paper is organized as follows. In Section II we report on previous works using motion primitives and dimensionality reduction techniques. Section III describes the elements of our proposed method, *ProMPs* and *AE*, as well as *VTSFE* that is combined with *ProMP* and compared to *AE-ProMPs* in the experiments of Section IV. In Section V are discussed the experimental results, where we show that *AE-ProMP* is computationally more efficient than *VTSFE-ProMP* and has a better reconstruction of the inferred trajectories from the latent space to the original space. Section VI draws the conclusions and outlines the future works towards the use of *AE-ProMP* in an assistive robotics scenario.

II. Related Works

The key elements in our framework are probabilistic movement primitives, which capture the information about the current trajectory and predict its future, and autoencoders, which reduce the dimensionality of our whole-body trajectories into a latent space. In the following, we outline the related works in these two domains.

A. Learning Movement Primitives

Complex trajectories and activities can be modeled and recognized with different approaches, such as recurrent neural networks [9] or Hidden Markov Models (HMM) [10]. Here, we are more interested in parametric techniques that represent trajectories as movement primitives. A classic method, called *Dynamic Movement Primitives* (DMPs) [11][12], models trajectories using an attractor point at the end (i.e., the goal in a reaching primitive), and a forcing function to capture the shape of the trajectory, i.e., its evolution in time. An improvement of DMPs, called Probabilistic Dynamic Movement Primitives [13] allows learning movement distributions rather than individual trajectories. This is a desirable feature to encapsulate the human movements variability and improve the inference of the future trajectories. A limitation of these methods, for our application, is their dependency on the attractor point. While it can be available for goal-directed movements, it is not necessarily the case for generic human movements such as walking or carrying an object.

In [14], Paraschos *et al.* proposed the *Probabilistic Movement Primitives* method (ProMPs - detailed in Section III-A), which captures the probability distribution of demonstrated trajectories over time, with several features such as co-activation, coupling and temporal scaling. In [15], Maeda *et al.* showed that ProMPs are more efficient than DMPs for prediction, while [16][17] showed that ProMPs are better for generalizing trajectories into primitives. In our previous work [2], [4], we used ProMPs to infer the future intended trajectories during human-robot interaction, using haptic signals and gaze cues: we were able to predict the future trajectories despite variations in the demonstrated trajectories and their duration, and considerable noise. However, we addressed simple movements with a small dimension (e.g., 6), whereas here we need to infer the future of whole-body movements with a bigger dimension (e.g., 69 and more)[1]. For such bigger dimensions, classical ProMPs are computationally inefficient (c.f. Section V-A). Scaling up to higher dimension while being computationally efficient is possible with ProMPs by optimizing the matrix computation and setting a fixed number of observations for prediction, which permit to pre-compute the gain matrix that does not depend on the observations. However, in our application the number of observations shall remain variable, and whatever optimization we do, the computation time will still increase with the input dimension. For this reason, dimensionality reduction techniques are an appealing alternative.

B. Dimensionality reduction in a latent space

Dimensionality reduction is a critical process in machine learning and data processing, as it requires extracting from the data a reduced set of principal features that describe a process. The most common method for dimensionality reduction is the Principal Component Analysis [18]. *Autoencoders* (AEs) are another classical method for reducing the dimensionality of data. They are neural networks that learn to encode data in a latent space of a lower dimension than the original input space, through the minimization of a loss function that measures the distance between the original data and the data reconstructed from this latent space. A recent variant of AEs is *Variational autoencoders* (VAEs) [19], which is a combination of an autoencoder with variational inference [20]. Variational inference is an approach that approximates a probability density function through parameter optimization of a known probability density function (*e.g.*, Gaussian distribution). Both methods are excellent functional approximators and have been used for dimensionality reduction of complex functions in unsupervised way [21], [22].

In [23], Colomé *et al.* presented a method that reduces the dimensionality of the ProMPs, called DR-ProMP, to find low-dimensional walking policies for the Nao robot. They used probabilistic dimensionality reduction techniques on a set of trajectory demonstrations to extract the unknown synergies between the dimensions, producing a new ProMP expression in which a coordination matrix maps the lower-dimensional latent virtual joint space into the real-dimensional robot joint space. The latent space dimension was manually tuned (usually 4-5); the maximum size of the original space was 15, which makes the compression ratio not interesting for our application.

[1]In this paper our data size is 69, since we only consider the kinematics of the human skeleton model, but the size could grow considerably if we were considering also joint torques and wrenches.

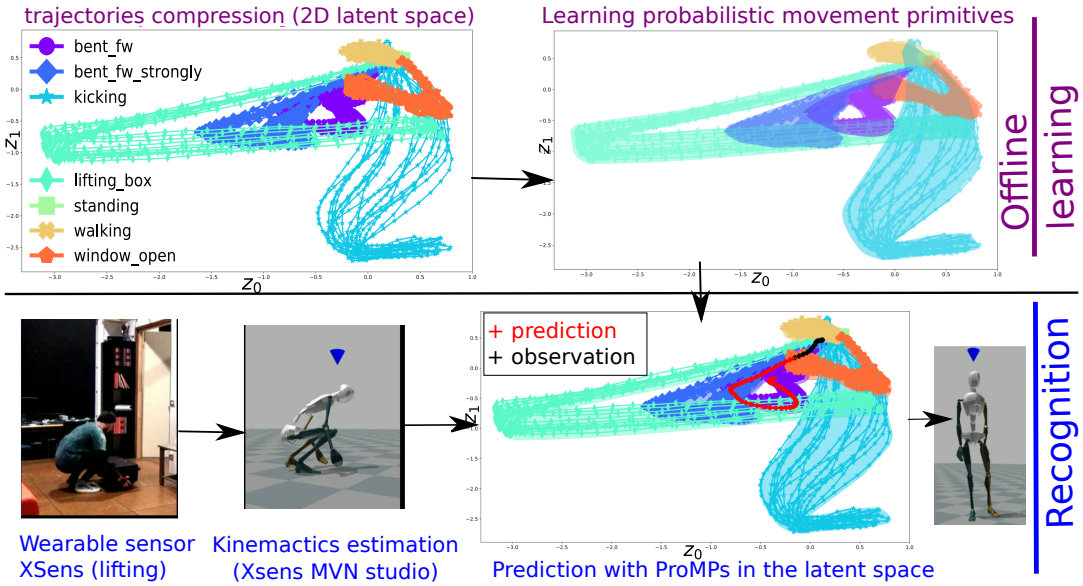


Fig. 2: Concept of our proposed method to predict future trajectories. We encode high-dimensional postures (with AEs) or postural trajectories (with VTSFE) into a low-dimensional latent space (in the figure, 2-dimensional with z_0 and z_1). From several low-dimensional encoded trajectories the ProMPs learn a trajectory distribution for each action. This prior information is used to predict the future trajectory (in red) given some initial observations (in black).

While AEs and VAEs can be used to learn the whole-body human posture, they cannot be used to represent whole-body trajectories over time in a smooth and coherent manner (*i.e.*, without jolts) since there is no postural time-dependence. This issue was well explained by Chen *et al.* in [8], which proposed to force the temporal dependency by learning DMPs in the latent space. Their method, called VAE-DMP, uses Deep Variational Bayes Filters (DVBF) [24], where Bayesian filtering is applied on latent variables with temporal dependencies, with a recurrent deep neural network composed of chained VAEs An improvement of VAE-DMP, called *Variational Time Series Feature Extractor* (VTSFE), was proposed by Chaveroche *et al.* in [5] to encode features of the time series for both classification and prediction purposes. These methods are interesting for mapping trajectories from high to low dimensional spaces, however they are computationally expensive.

In this paper, we propose two methods that combine the prediction capabilities of ProMPs with the dimensionality reduction of AEs and VTSFE: we call them respectively AE-ProMPs and VTSFE-ProMPs. These two methods follow two different ideas: in AE-ProMPs the AE compresses postures while in VTSFE-ProMPs the VTSFE compresses the whole trajectory directly; in both cases ProMPs infer the future trajectory in the latent space. We will show that to predict the future trajectories given early observations AE-ProMPs is faster and more performing. The next section details the methods.

III. Methods

In the following, we present in detail AE-ProMPs and VTSFE-ProMPs. We first present ProMPs and how they are used to do inference, then introduce AEs and VTSFE. For the sake of clarity, we sketch in Figure 3 the differences between the two methods applied to our problem of encoding human postures/trajectories in a latent space.

A. Probabilistic Movement Primitives (ProMPs)

A *ProMP* is a Bayesian parametric model:

$$\xi(t) = \Phi_t \boldsymbol{\omega} + \epsilon_\xi \tag{1}$$

where:

- $\xi(t) \in \mathbb{R}^N$ is the vector containing all the variables to be learned at time t.
- the matrix Φ_t corresponds to the M Radial Basis Functions (RBFs) evaluated at time t, with $\Phi_t = [\psi_1(t), \psi_2(t), \ldots, \psi_M(t)]$;
- $\boldsymbol{\omega} \in \mathbb{R}^M$ is a time-independent parameter vector that weighs the Φ matrix;
- $\epsilon_\xi \sim \mathcal{N}(0, \beta)$ is the trajectory noise variable.

During the learning phase, the weights $\boldsymbol{\omega}$ are learned from a set of trajectory demonstrations $\{\Xi_1, \ldots, \Xi_{n_1}\}$, where the i-th trajectory is $\Xi_i = \{\xi(1), \ldots, \xi(t_{f_i})\}$. The weights encode the probability distribution over the trajectories.

Given a set of different demonstrations for N_A different actions, N_A ProMPs are learned. They are used as prior knowledge to estimate from partial observations $\Xi^o = [\Xi_1 \ldots \Xi_{n_o}]^\top$ what is the current action $\hat{k} \in [1 : N_A]$ (*i.e.*, the most likely ProMP from the ones learned) and to predict its future trajectory, as done in [2][4].

Once the current $\hat{k}$-th ProMP is identified, the recognized

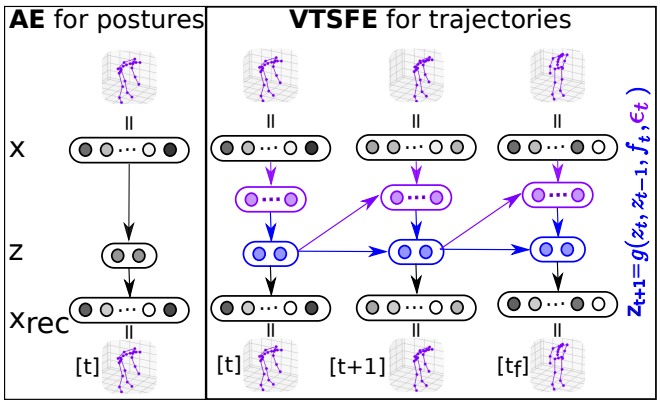


Fig. 3: Relation between AEs and VTSFE for encoding trajectories in a low-dimensional latent-space.

distribution (called "prior") can be updated by:

$$\begin{cases} \hat{\mu}_{\boldsymbol{\omega}_{\hat{k}}} &= \mu_{\boldsymbol{\omega}_{\hat{k}}} + K(\Xi^o - \Phi_{[1:n_o]}\mu_{\boldsymbol{\omega}_{\hat{k}}}) \\ \hat{\Sigma}_{\boldsymbol{\omega}_{\hat{k}}} &= \Sigma_{\boldsymbol{\omega}_{\hat{k}}} - K(\Phi_{[1:n_o]}\Sigma_{\boldsymbol{\omega}_{\hat{k}}}) \\ K &= \Sigma_{\boldsymbol{\omega}_{\hat{k}}}\Phi_{[1:n_o]}^\top(\Sigma_{\xi^o} + \Phi_{[1:n_o]}\Sigma_{\boldsymbol{\omega}_{\hat{k}}}\Phi_{[1:n_o]}^\top)^{-1} \end{cases} \quad (2)$$

Finally, the inferred trajectory is given by:

$$\forall t \in [1:t_f], \hat{\xi}(t) = \Phi_t\,\hat{\mu}_{\boldsymbol{\omega}_{\hat{k}}}$$

with the duration of the trajectory t_f, which corresponds to the number of frames used to represent the trajectories. The movement continuation can be predicted by identifying the most-likely "future" trajectory $\hat{X} = [\hat{X}_{n_o+1} \dots \hat{X}_{t_f}]^\top$, as explained in [2], [14].

B. Dimension compression using an autoencoder

In their simplest form, AEs are multilayer perceptrons in which the output layer, of the same dimension as the input layer, is trained to reconstruct the input [25]. Their structure is often symmetrical and split between the encoder and decoder parts. On the one hand, the encoder transforms the input x_t of dimension N, in our case the whole-body kinematic information, into a value z_t in the latent space of dimension R. On the other hand, the decoder transforms the latent space back into the reconstructed kinematic space. Each of the encoder and decoder usually includes a number of hidden layers and non-linear activation functions in order to build a non-linear compressed latent space.

C. Dimension compression using Variational Time Series Feature extractors (VTSFE)

As introduced in Section II, VTSFE [5] is an improvement of VAE-DMP [8]. Both methods construct a function to simply project the input vector x in the latent space independently from time, like AEs or VAEs do. However, they differ in the way that function is learned and therefore construct different latent spaces.

First, VTSFE has a more accurate model to represent the noise of the trajectory inference (*e.g.* the distribution representing inference errors), since inference errors are the difference between what the transition model predicts (through all its variables) and the real trajectory. It has a simpler transition model, representing better the trajectories since their acceleration is only constrained in latent space, i.e. the space of z, and not in the space of its arbitrarily defined derivative $\dot{z}$. It also does not require to know the final trajectory position/goal, which is better since the constraint does not rely on a changing and arbitrarily defined position in latent space. Moreover, the inferred trajectories are further improved through the design of a loss term on the "inferred dynamics f", which makes the final optimization closer to the actual theoretical optimization pointed by *Variational Inference*. The "Inferred dynamics f" is the sequence of forcing terms f_t in the latent space (i.e. a force applied at time t in latent space that influences the encoded trajectories). These forcing terms are inferred from multiple trajectory demonstrations for each movement type and only used during the latent space learning process. Thus, they force the encoded trajectories to follow the same dynamics than the demonstrations.

More details about this method can be found in [5]. The most visible difference between VTSFE and VAE-DMP is the noise inference: with VTSFE, the encoder tries to optimize its parameters ϕ to infer ϵ_t not only with x_{t+1} and z_t, but also with all other variables used in the transition model. Therefore, it sets the known distribution as a Gaussian one for the inference of ϵ_t as follows: $q_\phi(\epsilon_t|f_t, x_{t+1}, z_t, z_{t-1}) = \mathcal{N}(\mu_{\epsilon,t}, \sigma^2_{\epsilon,t}I)$.

The other difference is that the transition model of VTSFE does not use the classical *DMP* model but a simple discretized acceleration model with an extra forcing term $f_t + \epsilon_t$ instead:

$$z_{t+1} = (f_t + \epsilon_t)dt^2 + 2z_t - z_{t-1} = g(z_t, z_{t-1}, f_t, \epsilon_t)$$

This transition model does not need z_T nor $\dot{z}_t$ nor the *DMP* parameters α and β that needs to be optimized, for exemple with grid search.

D. Inference on a compressed representation of trajectories using AE-ProMPs or VTSFE-ProMPs

We propose to use ProMPs to represent the movement primitives encoded in a low-dimensional latent space, encoded by either AE or VTSFE: the two methods are then called AE-ProMPs and VTSFE-ProMPs. The concept is shown in Figure 2 and 4. For both cases, there is a preliminary learning phase, offline, before the online prediction phase.

First, we train the AE or the VTSFE to compress the original data into the latent space. For the AE, we encode the individual postures (dimension N) into a R-dimensional latent space, with $R \ll N$. For the VTSFE, we encode the entire posture trajectories (dimension N, for each frame $t = 1, \dots, t_f$) into a cascade of t_f VAE, with $R \ll N$ and t_f the number of frames used to represent a movement. The decoded trajectories are $x_{rec}(t) = dec(enc(x(t)) = x(t) + \epsilon_v(t), \forall t \in [1:t_f]$, with $\epsilon_v(t)$ the reconstruction error. Then, we learn the N_A ProMPs associated to the N_A actions, using the sets of demonstrated trajectories compressed in the latent space: $\xi(t) = [z_1(t), \dots, z_R(t)]^\top = \Phi_t\boldsymbol{\omega} + \epsilon_\xi$

Once the ProMPs are learned, we can use them to predict the future trajectories given partial observations. This phase

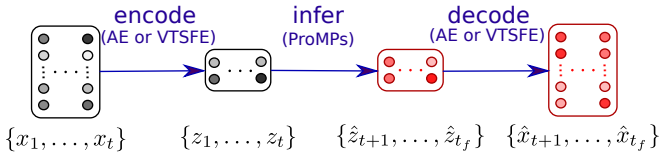


Fig. 4: The encoding-decoding actions of AE and VTSFE combined with the prediction performed by the ProMPs in the latent space.

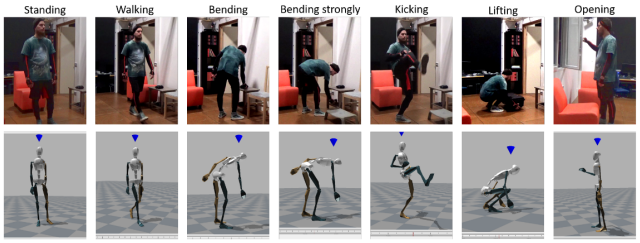


Fig. 5: The 7 actions of the dataset of [6], with the kinematic estimation of the human posture by Xsens MVN Studio.

can be performed online. During the "recognition step", observations of a movement initiated by the user are retrieved (first image of the "recognition part"). In the prediction phase, we record early observations of the human movement, and encode them in the pre-trained latent space: $\xi^o = [z_1^o(t), \ldots, z_R^o(t)]^\top, \forall t \in [1 : n_o]$. Then, using the *ProMPs* inference [2], we compute the continuation of this compressed-partial trajectory: $\hat{\Xi} = [\hat{\xi}_{n_o+1} \ldots \hat{\xi}_{t_f}]^\top$. Finally, the *AE* or *VTSFE* decoding is performed to obtain the prediction of the future whole-body movement: $dec(\hat{\Xi}) = [\hat{x}_{n_o+1} \ldots \hat{x}_{t_f}]^\top$.

IV. Experiments

The goal of the experiments is to compare three different methods (ProMPs, AE-ProMPs and VTSFE-ProMPs) when applied to predict the future human movement given early observations. We use the actions dataset from [6], consisting of ten movement demonstrations for seven different actions (see Figure 5): bending forward, bending strongly forward, lifting a box, kicking, opening a window, walking and standing. The trajectories were recorded with the XSens MVN suit, which tracks the human motion with a 23DOF skeleton model. From these recordings, using XSens MVN Studio, we retrieve the 3D Cartesian positions of the human segments. Thus, the posture of the human operator is represented by $3 \times 23 = 69$ Cartesian position variables. Each trajectory demonstration has been re-sampled to last seventy frames ($t_f = 70$), to enable the comparison with VTSFE that needs a fixed duration trajectory, as explained in [5].

A. ProMPs-only

We compute the ProMPs of the 7 different actions without dimensionality reduction. The objective of this experiment is to show the accuracy of the prediction when using ProMPs, but also the main limitation of the computation time when the input dimension grows.

In this experiment, the vector $\xi(t) \in \mathbb{R}^{69}$ of Eq.1 contains all the segment positions that represent the whole-body movement of the human: $\xi(t) = [a_{1,1(t)}, ..., a_{23,3}(t)]^\top$, with $a_{i,j}$ the i^{th} segment position with coordinate $j \in \{x, y, z\}$. The ProMP model contains 10 basis functions to represent trajectories. We compute the full covariance matrix, coupling all the joints to record redundancy of information between the joints.

B. AE-ProMPs

We use AEs to compress the 69-dimensional original posture data into a low-dimensional latent space, where we learn the ProMPs to make our predictions. In this experiment, ProMPs are learned from the encoded postures (*i.e.,* from the latent space of the AEs), with for example $\xi(t) = [z_1(t), z_2(t)]^\top$ when $R = 2$.

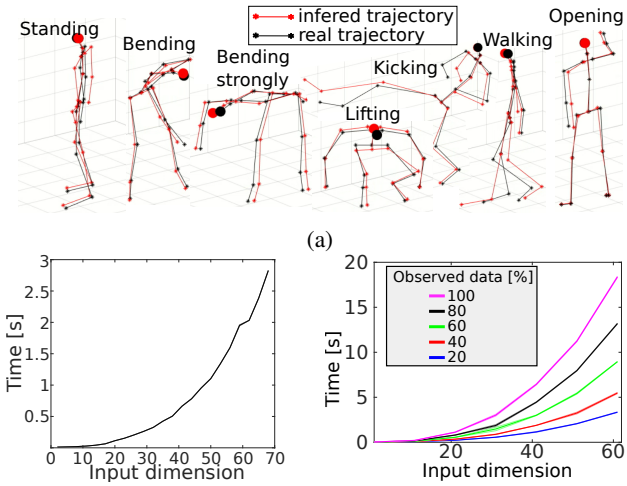
To compress the original data in the latent space, we use a simple AE composed of different layers. An input layer with N units $x = \{x_1, \ldots, x_{69}\}$ for the entire posture values (*i.e.*, 69 units). A compressed-layer (latent space) with a variable number R (*e.g*, 10 units) of units $z = \{z_1, \ldots, z_R\}$. An output layer with the decoded posture that has the same dimension N of the input layer (*i.e.*, 69 units). We call these units $x_{rec} = \{x_{1,rec}, \ldots, x_{69,rec}\}$. Finally, two hidden layers, one between the input and the compressed layers and the other between the compressed and the output layers (*e.g.*, 500 units). We call these layers h_j with $j \in [1, \ldots, R]$ and its $i-th$ unit: h_{jx_i}. The weights of this neural networks are initialized using the Xavier initialization [26], where weights are scaled by a uniform distribution. For the activation function of all units, we choose the "leaky ReLU" [27]: it is similar to "ReLU" (rectified linear unit), but the function is not zero for negative values, it has a small negative slope (*i.e.,* $f(x) = \mathbb{1}_{x<0}\alpha x + \mathbb{1}_{x>=0}x$, with $\alpha = 0.5$ in our case). We choose this function after having compared its performance with the sigmoid and ReLu activation functions. Finally, the neural networks are learned by using the least mean square error between x and x_{rec} as loss function, and ADAM as gradient descent optimizer [28].

After having modeled this neural network, the AE was trained using $\frac{2}{3}$ of all postures of the 70 trajectory demonstrations, that is 30916 postures. Then, the AE was tested using the last $\frac{1}{3}$ postures.

Then, the ProMPs are learned, from 69 encoded trajectory demonstrations. The learning steps are done 35 times since we tested 5 samples from each of the 7 actions using the *leave-one-out cross-validation*.

C. VTSFE-ProMPs

We use VTSFE to encode the entire postural trajectory (69-dimensions $\times$ 70 frames $=$ 4830) in a dynamically consistent latent space, where we learn the ProMPs to make our predictions. In this experiment, different latent space dimensions R are tested, for example $\xi(t) = [z_1(t), z_2(t)]^\top$ when $R = 2$. The objective of this experiment is to verify whether encoding the entire trajectory instead of instant



(b) Time to compute movement distributions from 10 demonstrations. (c) Time to infer the movement trajectory continuation.

Fig. 6: *ProMPs-only* - (a) some frames representing the real trajectories (black) and their inference (red) after observing 5% of the whole trajectory (3 frames). Bottom: Computation time according to the latent space dimension used to represent trajectory. The trajectories are represented by 70 frames.

postures improves not only the latent space (we already know this from [8] and [5]) but the inference.

V. Results and Discussion

We are interested into evaluating the performance of the three methods from the point of views of computation time, complexity, inference and prediction capabilities. All results and their representations with boxplots are done using the leave-one-out cross-validation and the 70-frames trajectories in off-line trials. To test the performance of the proposed methods, we use three different distance metrics:

1) Err_{AE} or *autoencoding error*, evaluates the autoencoding performance. It is the average distance between the reconstructed trajectory x_{rec} (*i.e.*, the autoencoded trajectory) and the real one x: $Err_{AE} = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^{N} |x_{j,rec}(i) - x_j(i)|$

2) Err_I or *Inference error*, evaluates the inference performance. It is the average distance between the reconstructed trajectory x_{rec} and the reconstruction of the infered one $\hat{x}$: $Err_I = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^{N} |x_{j,rec}(i) - \hat{x}_j(i)|$

3) Err_{AE+I} or *global error*, evaluates the overall method, so the global performance in encoding and inference. It is the average distance between the real trajectory x and the infered one $\hat{x}$: $Err_{AE+I} = \frac{1}{t_f N} \sum_{i=1}^{t_f} \sum_{j=1}^{N} |\hat{x}_j(i) - x_j(i)|$

A. Future movement prediction using ProMPs only

Figure 6a represents the inference of the ProMPs in the original data space ($N = 69$) after observing 5% of a trajectory (3 frames). The frames were taken from the video attachment at a representative moment of the movements. Even if this method is performing in representing trajectories [17], [15], its computation time for prediction increases quadratically with the number of data represented by the ProMPs [23]. In this case, prediction is done on a 69-dimensional vector: Figure 6b represents the average time to compute movement distributions on all the demonstrations, and Figure 6c the average time to infer the movement continuation. The computation time is simply too long for our targeted application, where we need to predict the future human trajectory in few ms. This issue motivates our approach to reduce the dimensionality of the problem.

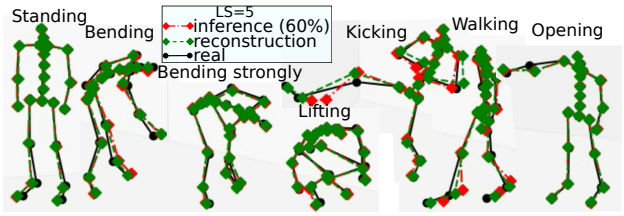
B. Future movement prediction using AE-ProMPs

Figure 7a shows the prediction of human trajectories encoded in a 5-dimensional latent space. Again, the frames were taken at representative moments of the movement, from the video attachment. Figure 7b shows an extract from the latent space trajectories for z_1: the few irregularities in the trajectories do not negatively affect the ProMPs, therefore not causing problems in the prediction phase.

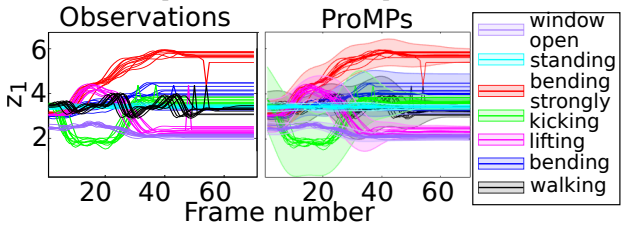
Figure 8 shows the accuracy of AE-ProMPs with respect to the latent space dimension and the percentage of partial observations used for the inference. In the top row, the boxplots compare the distance errors from the entire original trajectory after reconstruction (label "*AE only*") and the infered trajectory after reconstruction, when the prediction is done on 60% of partial observations. In Figure 8a, the distance error decreases with the latent space dimension, and the plots suggest that a latent space dimension of 5 is a good compromise to get a good inference without a long computation time. In Figure 8b, the distance error is computed within the latent space, between the compressed real trajectory and the inferred one. In contrast to the previous graph, the distance error increases with the latent space dimension. Indeed, the smaller the latent space dimension is, the less the compressed-trajectory contains information and thus, the smaller the error is. But when the latent space is smaller, the compressed-trajectory contains less information about the real trajectory: so when the trajectory is decompressed, the inference gives poorer results (Figure 8a). The bottom row shows the accuracy of the method for this latent space dimension ($R = 5$). The error induced by the encoding Err_{AE} is relatively small, the global error at the end is affected by the inference error Err_I, which decreases, as expected, with the more observations available for the inference. One can remark that after 60% of observations, the method can infer the future whole-body trajectory with a distance error around $1cm$, which is a very good performance for our targeted application.

C. Future movement prediction using VTSFE-ProMPs

Figure 9a shows the prediction of human trajectories encoded in a 5-dimensional latent space. The bottom of Figure 9 shows the accuracy of VTSFE-ProMPs according to the number of observations, for different latent spaces dimensions. Figure 9c shows that the reconstruction is not performing: whatever the latent space dimension, Err_{AE} is almost constant, the global error Err_{AE+I} as well, which suggests that the inference error Err_I is significantly smaller than the encoding error Err_{AE} and does not influence the

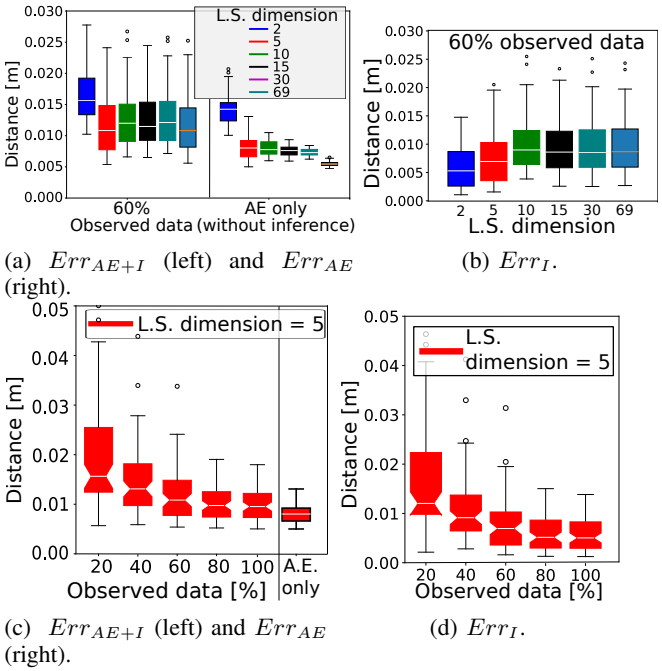


(a) Inferred postures after 60% of partial observations.



(b) An example of the latent space: the latent trajectories in z_1 (left) and their corresponding ProMPs (right).

Fig. 7: *AE-ProMPs*: experiment with a 5-D latent space.



(a) Err_{AE+I} (left) and Err_{AE} (right).

(b) Err_I.

(c) Err_{AE+I} (left) and Err_{AE} (right).

(d) Err_I.

Fig. 8: *AE-ProMPs* - Accuracy according to the latent space dimension (top) and the percentage of observations (bottom).

overall performance of the method. Figure 9b compares the inference error Err_I for two latent space dimensions (2 and 5): overall Err_I decreases with the more available observations in both dimensions, as expected. However, this error is 1 to 2 orders of magnitude smaller than the encoding error, so its contribution does not impact the overall performance of the method.

In [5], Chaveroche *et al.* explain that VTSFE gives better encoding than VAE-DMP but a poorer decoding due to the variational inference during its learning, which forces the decoded data not to vary much. This is consistent with our observations (see the video attachment) where VTSFE has problems in encoding movements with big variations of the segments positions (or the joints), such as the knees and feet in kicking and walking. The encoded trajectories are more conservative around the average posture. In AE-ProMPs we do not have this problem, because the AE learns to encode instantaneous postures without forcing a continuity over the postural trajectories: therefore, it is capable of encoding also more extreme postures, and reconstruct trajectories with high variations in the segments and joints positions.

D. Accuracy vs computation time

Table I provides a comparison between the three methods in terms of accuracy and computation time necessary for the prediction of the future whole-body movement.

Inference from 20% observation		Accuracy prediction [m]	Computation time [s]
ProMPs (69 dimensions)	mean	0.0145	2.5378
	var	1.0038e-04	0.0357
VTSFE-ProMPs (L.S.= 5)	mean	0.04219	0.0565
	var	0.002	0.0024
AE-ProMPs (L.S.= 5)	mean	**0.02793**	**0.0516**
	var	0.003	0.0028

TABLE I: Mean and variance of the distance error between the ground truth trajectories and the inferred ones, and of the inference computation times for both *ProMP* et *VTSFE-ProMPs* methods.

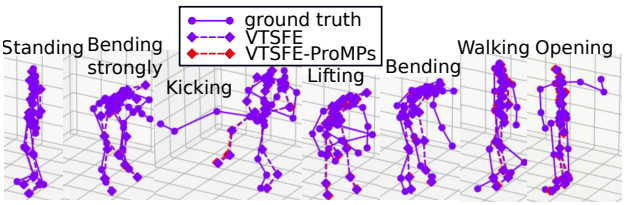
Even if this prediction is done only from 20% of partial observations of the whole trajectory, we can see that the computation time of ProMPs is a lot longer than the two other methods. VTSFE-ProMPs is the less accurate, for the reasons we already explained. Thus, for our targeted application, the best method is AE-ProMPs, which outperforms ProMPs for the computation time and VTSFE-ProMPs for the inference and reconstruction ability.

E. VIDEO

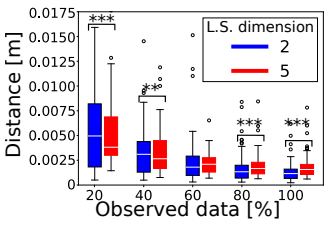
The video attachment shows the predicted future movements after $30-60\%$ of observations, for the three methods (ProMPs-only, AE-ProMPs, VTSFE-ProMPs).

VI. Conclusion

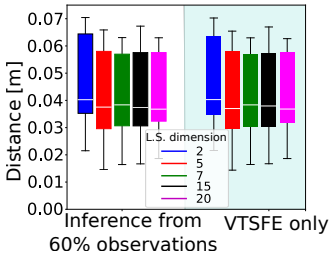
In this paper we propose a new method for high-dimensional movement prediction, called *AE-ProMP*. This method combines the dimensionality reduction of an AEs with the prediction capabilities of the *ProMP* method. The AEs is used to compress the postures into a low-dimensional latent space. The *ProMP* method is used to infer the continuation of movement given some early postures, by learning the trajectory distributions in the latent space. Our results show that AE-ProMPs allows to predict accurately whole-body movements encoded in a low-dimensional (e.g., 5D) latent space, with a reduced computation time. However, we can



(a) Inferred postures after 60% of observations from video V-E.



(b) Err_I.



(c) Err_{AE+I} (left) and Err_{AE} (right).

Fig. 9: *VTSFE-ProMPs* - Inference of future postures after 60% of observations, for 7 actions. Top: reconstructed trajectory inference (red); reconstructed observed trajectory (dashed purple); and the real trajectory (purple). Bottom: Accuracy of the method.

assume that the larger the input dimension, the less accurate the prediction is. Thus, to go further in our study, we should quantify the decrease in prediction accuracy, according to the input dimension for a same latent space dimension. We compared our proposed method with a simple prediction using ProMPs alone and with a combination of VTSFE and ProMPs, where the encoding is about the whole trajectory rather than the single posture. In the first case, ProMPs alone are too computationally slow for our targeted real-time applications. In the second case, despite the better dynamically consistent latent space, VTSFE requires more training resources to perform. In the future, we plan to improve AE-ProMPs by predicting trajectories of different durations, as we did in [2], and improving the accuracy of the encoding-decoding by automatically setting the latent space dimension and testing different variants of AEs.

Acknowledgments

The authors wish to thank Adrien Malaisé for his support with the XSens suit and the anonymous reviewers for their insightful comments.

References

[1] S. Ivaldi, "Intelligent human-robot collaboration with prediction and anticipation," ERCIM news, vol. 114, pp. 9–11, July 2018.
[2] O. Dermy, A. Paraschos, M. Ewerton, J. Peters, F. Charpillet, and S. Ivaldi, "Prediction of intention during interaction with iCub with probabilistic movement primitives," Front. Robotics and AI, 2017.
[3] S. Ivaldi, L. Fritzsche, J. Babič, F. Stulp, M. Damsgaard, B. Graimann, G. Bellusci, and F. Nori, "Anticipatory models of human movements and dynamics: the roadmap of the andy project," in DHM, 2017.
[4] O. Dermy, F. Charpillet, and S. Ivaldi, "Multi-modal intention prediction with probabilistic movement primitives," in Human-Friendly Robotics, 2017.
[5] M. Chaveroche, A. Malaisé, F. Colas, F. Charpillet, and S. Ivaldi, "A Variational Time Series Feature Extractor for Action Prediction," ArXiv e-prints, July 2018.
[6] A. Malaisé, P. Maurice, F. Colas, F. Charpillet, and S. Ivaldi, "Activity recognition with multiple wearable sensors for industrial applications," in Advances in Computer-Human Interactions, 2018.
[7] "Xsens the leading innovator in 3d motion tracking technology." 2017. [Online]. Available: http://www.xsens.com/products/xsens-mvn/
[8] N. Chen, M. Karl, and P. van der Smagt, "Dynamic movement primitives in latent space of time-dependent variational autoencoders," in 16th IEEE-RAS, Humanoids, Cancun, Mexico, 2016, pp. 629–636.
[9] R. W. Paine and J. Tani, "Motor primitive and sequence self-organization in a hierarchical recurrent neural network," Neural Networks, vol. 17, no. 8-9, pp. 1291–1309, 2004.
[10] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden markov model," in CVPR. Computer Society Conference on, vol. 2. IEEE, 2005, pp. 955–960.
[11] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," Neural computation, vol. 25, no. 2, pp. 328–373, 2013.
[12] S. Schaal, "Dynamic movement primitives: a framework for motor control in humans and humanoid robotics," in Adaptive motion of animals and machines. Springer, 2006, pp. 261–280.
[13] F. Meier and S. Schaal, "A probabilistic representation for dynamic movement primitives," arXiv preprint, 2016.
[14] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in NIPS, 2013, pp. 2616–2624.
[15] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in Humanoids, 2014 14th IEEE-RAS International Conference on. IEEE, 2014, pp. 527–534.
[16] M. Ewerton, "Learning motor skills from partially observed movements executed at different speeds," 2016.
[17] A. Paraschos, G. Neumann, and J. Peters, "A probabilistic approach to robot trajectory generation," in Humanoids, 13th IEEE-RAS International Conference on, 2013, pp. 477–483.
[18] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 61, no. 3, pp. 611–622, 1999.
[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," CoRR, 2013.
[20] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," CoRR, 2016.
[21] C. Doersch, "Tutorial on variational autoencoders," CoRR, 2016.
[22] A. Droniou, S. Ivaldi, and O. Sigaud, "Deep unsupervised network for multimodal perception, representation and classification," Robotics and Autonomous Systems, vol. 71, pp. 83–98, 2015.
[23] A. Colomé, G. Neumann, J. Peters, and C. Torras, "Dimensionality reduction for probabilistic movement primitives," in Humanoids, 14th IEEE-RAS International Conference on, 2014, pp. 794–800.
[24] M. Karl, M. Sölch, J. Bayer, and P. van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," CoRR, 2016.
[25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.
[27] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.