# Multi-frame sequence generator of 4D human body motion

Mathieu Marsot, Stefanie Wuhrer, Jean-Sébastien Franco, Stephane Durocher

# Multi-frame sequence generator of 4D human body motion

Mathieu Marsot[a], Stefanie Wuhrer[a], Jean-Sébastien Franco[a], Stephane Durocher[a,b]

[a] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France*
[*] *Institute of Engineering Univ. Grenoble Alpes,*
[b] *University of Manitoba, 66 Chancellors Cir, Winnipeg, MB R3T 2N2, Canada*

**Abstract**

We examine the problem of generating temporally and spatially dense 4D human body motion. On the one hand generative modeling has been extensively studied as a per time-frame static fitting problem for dense 3D models such as mesh representations, where the temporal aspect is left out of the generative model. On the other hand, temporal generative models exist for sparse human models such as marker-based capture representations, but have not to our knowledge been extended to dense 3D shapes. We propose to bridge this gap with a generative auto-encoder-based framework, which encodes morphology, global locomotion including translation and rotation, and multi-frame temporal motion as a single latent space vector. To assess its generalization and factorization abilities, we train our model on a cyclic locomotion subset of AMASS, leveraging the dense surface models it provides for an extensive set of motion captures. Our results validate the ability of the model to reconstruct 4D sequences of human locomotions within a low error bound, and the meaningfulness of latent space interpolation between latent vectors representing different multi-frame sequences and locomotion types. We also illustrate the benefits of the approach for 4D human motion prediction of future frames from initial human locomotion frames, showing promising abilities of our model to learn realistic spatio-temporal features of human motion. We show that our model allows for data completion of both spatially and temporally sparse data.

## 1. Introduction

This work investigates learning models for generating temporally and spatially dense 3D human body motion. That is, the proposed model simultaneously generates a multi-frame sequence of dense 3D meshes based on a single low-dimensional latent space.

Such a model is of significant interest for several key reasons. Static generative models are widely used for 3D morphable shapes such as human faces or bodies [1], because they allow for shapes to be encoded with latent parameterization spaces that possess a semantically meaningful structure, with orthogonal parameters for morphology and pose that can be individually controlled and interpolated. Allowing a generative model to additionally encode multi-frame evolution of full shapes in a single latent vector as proposed, should yield a widely useful generalization of these structural latent space properties to the temporal domain, where not only single shape poses but full shape-in-motion sequences can be parameterized, interpolated and controlled in a semantically meaningful and concise fashion. This is of broad interest for a wide set of applications where the lightweight 4D representation translates to gains in information processing and transfer, *e.g.* virtual reality, computer graphics, or 3D telepresence. But it also opens new possibilities for a wide set of spatio-temporal completion tasks from temporally or spatially sparse or incomplete inputs, for shape sequence reconstruction, motion transfer and retargeting.

Learning a generative model for spatially and temporally dense 3D human motion data presents two major challenges. First, the amount of data that needs to be processed for training is large, as typical acquisition systems for dense human body motions produce $30 - 50$ frames per second, with each frame containing thousands of geometric primitives. Second, the model needs to capture different factors of variation, including global motion, body pose, temporal evolution of the motion, and morphology. We demonstrate that the latent space learned by our model not only captures these variations, but allows to explore them in a semantically meaningful way.

To address these challenges, we combine the advantages of two existing lines of work. The first studies temporally dense skeletal data, with the goal of generating skeletal human body motion sequences [2, 3], and good results can be achieved for this problem using deep learning based methods (*e.g.* [4]). Extending this work to allow for spatially dense data and varying morphology directly is not straightforward, as the employed algorithms do not scale up to per-frame representations containing thousands of vertices. The second line of work considers

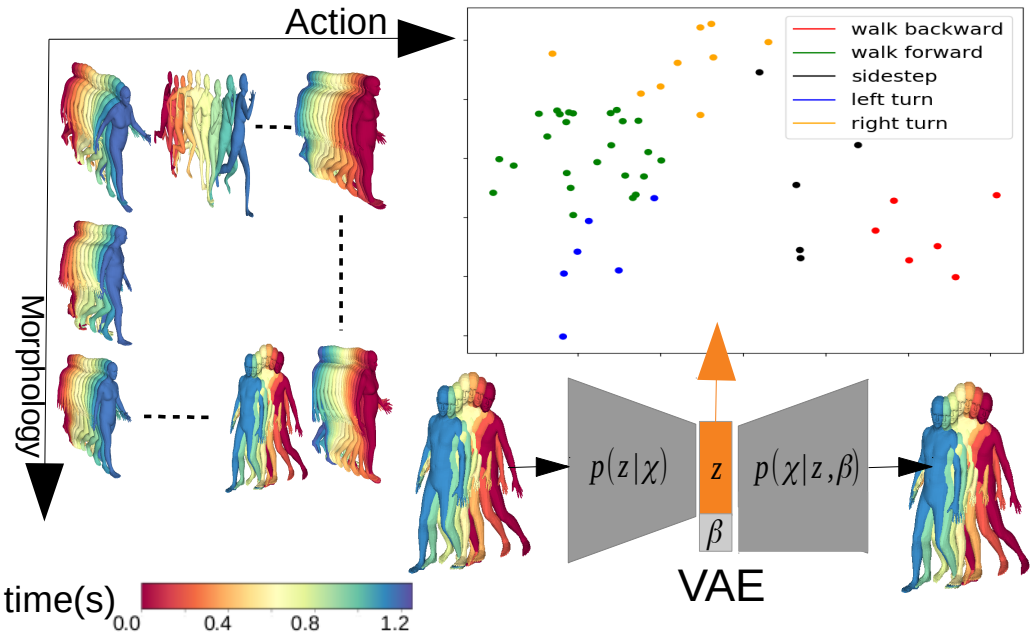


Figure 1: We learn a latent *motion space* from multi-frame 4D sequences of different types of cyclic motion performed by different subjects (left). A simple architecture (bottom right) leads to a single latent space encoding 4D motion that holds semantic structure (top right).

spatially dense static data, with the goal of representing realistic 3D human body surfaces in a low-dimensional shape space [5, 6]. While this problem has been studied in the context of shape and pose variations, and even including soft tissue components learned from dynamic data (*e.g.* [7]), existing works model motion as a sequence of per-frame fittings. That is, variation in the temporal dimension, most notably global displacements and the temporal evolution of the motion, is not learned by these models, but needs to be provided as per-frame control signal to generate motion.

We combine the advantages of these two lines of work in an encoder-decoder architecture that operates directly on multi-frame sequences as input and output of our architecture, a thorough review of which is provided in §2. In order to overcome the problem of the increased complexity caused by spatially dense data, we opt for a low-dimensional shape space parameterization of static human bodies, *e.g.* [8, 1]. The key difference of our work with the latter methods (details in §3) is that we present a latent space which encodes a full sequence of these low-parameter shapes in a single latent space vector, which serves as *motion space* and

captures variations in human posture, global locomotion and the temporal evolution of the motion. To capture the interaction between differences in morphology and motion, the decoder is conditioned on a low-dimensional representation of morphology.

To validate our model, we need to characterize on a representative example set its ability to factor information underlying to the data and generalize it for unobserved data. For this purpose, we focus our study on cyclic locomotions automatically selected as a subset of the AMASS dataset [9] of shapes in motion (§4.1). Importantly, the motions in our data can be of different durations and we include time stamps in our multi-frame vector, such that the temporal unfolding of actions can be analyzed and automatically refactored by our model. This allows us to produce a set of experiments meant to validate the accuracy of the generative model and the meaningfulness of interpolations between different types of cyclic locomotion, such as running, walking forward or backward, or sidestepping. We demonstrate that semantically similar locomotions (*e.g.* all walking motions or all running motions) tend to form clusters in the latent space, quantify the generalization and specificity of the model, and verify the interpolative property between motions that differ in temporal unfolding and duration, global translations, and pose (§4).

We apply our model to two application scenarios. First, we show that our model allows to successfully predict future frames of spatially dense 3D motion from a short input motion sequence, significantly outperforming a state of the art RNN-based method (§5). Second, we use our model to complete both spatially and temporally sparse sequences, with unmatched and temporally incoherent input point data, and show that accurate reconstructions are obtained using few sparsely sampled frames. For this application, we capture a new test dataset of cyclic human motion in a multi-camera system.

## 2. Related Work

There is a vast literature on the generation of human models and motions. We briefly review works that generate spatially dense 3D bodies, temporally dense 3D human motion, and first methods that model 4D human motion.

### *2.1. Generation of spatially dense 3D humans*

The analysis and synthesis of the 3D human body surface has been studied extensively. The first work in this area statistically analyzes static 3D body scans of

different subjects captured in similar postures using a linear model [10]. Follow-up works enhanced this model by including different postures. While initial works in this direction proposed models that are non-linear in the vertex coordinates (*e.g.* [5, 11, 12]), multiple linear models are now commonly used [13, 1, 14]. Most of these models have two separate sets of parameters for identity and pose information, and aim to decouple the two. Different methods focus on enhancing generative models of 3D human body shape to include additional variations such as hands and faces [15, 16]. Most related to this work, some methods learn soft-tissue deformations [6, 14, 17]. Recent works in this area leverage deep learning techniques to build dense 3D body surface models. These models can decouple variations due to different factors [18, 19] and include hands, faces and soft-tissue deformation [7].

These works allow to generate realistic 3D human models, and to learn dynamic aspects related to motion. However, the treatment of motion and dynamics is performed over single or few consecutive frames. Global body motions are therefore not analyzed. In contrast, our model learns a motion space that captures full 4D motion sequences in a single latent space. This allows in particular to generate motions with different global displacements and temporal evolutions without requiring per-frame input signals.

### *2.2. Generation of temporally dense 3D humans*

The analysis and synthesis of multi-frame 3D human motion sequences is a long-standing problem. Most existing works operate on spatially sparse data, such as motion capture (MoCap) markers or skeletal representations, and assume spatial alignment over time. Early works [20, 2] statistically analyze a dataset of motion-capture data recorded for different subjects performing a similar motion (*e.g.* walking). Motion sequences are spatially and temporally aligned for analysis, and the resulting model is subsequently used to synthesize gait patterns. Early work [21] also considers the problem of meaningfully interpolating between skeletal motions performed under different variations. More recently, deep learning has been used [22]. Sparse locomotion has also been analyzed toward predicting future motions based on a sequence captured either sparsely in 3D or as 2D video in various fields, *e.g.* robotics [23], computer vision [24] and computer graphics [25]. While traditional approaches include prior knowledge, recent methods use recurrent networks (*e.g.* [26, 4]). A recent work [27] completes motion sequences based on few input frames. While the works discussed so far learn the motion structure, they are spatially sparse.

Another line of work aims to generate dense 3D human motion from sparse MoCap [5, 28, 9] or 2D video data [29, 30]. For MoCap, the first approaches fit statistical shape models learned from static scans to MoCap data, allowing to synthesize realistic 3D human motion [5, 28]. More recently, this work has been extended using a statistical model that includes soft-tissue deformations. This model was fit to a large corpus of MoCap data [9], which provides the community access to a semi-synthetic training dataset for deep learning, which we leverage in this work. Unlike prior works, which require per-frame input signals to generate sequences, our work models full sequences in a single latent space, thereby allowing for motion generation based on sparse spatial and temporal input. For 2D video data, methods reconstruct dense 3D body motions from 2D input by fitting statistical models to individual frames of video data, and predicting short-term past and future motions from short clips and even static images [31, 29, 30].

### *2.3. 4D human motion models*

The work most closely related to ours has the same goal: generating dense 4D human motion for arbitrary identities [32]. To achieve this, two linear models are combined: one capturing static shape data and one capturing MoCap data. The two models are coupled based on semantic parameters, *i.e.* the method couples shape parameters from the first and skeletal motion parameters from the second linear model that match in terms of their semantic parameters. This allows generating 4D human motion sequences. In contrast to this method, our model learns using morphology and motion jointly rather than correlating two separate linear models. Furthermore, we employ a non-linear model, which allows to capture non-linear dependencies during training. We show experimentally that our model generalizes better than a linear one.

Furthermore, methods have been proposed that learn from 4D data (*e.g.* [33, 34, 35]). Unlike our work, these methods are subject-specific or train even on a single motion sequence. They can therefore not learn the interplay between morphology and motion. A recent work proposes a deep latent variable model for 4D human motion synthesis [36]. Like in our model, 4D motion sequences are mapped into a latent space, which allows to control the motion with sparse input. Unlike in our work, the focus is not on capturing the interaction between morphology and motion, but rather on modeling the probabilistic character of motion performed by a single subject.

Another recent work proposes using implicitly defined surfaces and correspondences over time to learn from 4D sequences [37]. From a set of 4D input sequences, the method learns a temporal flow of occupancy information. While this

work has been applied successfully to human motion data, it does not allow to decouple the influence of identity and motion information. Furthermore, while interpolations between static input frames is possible, interpolation between 4D sequences is not supported by this model.

## 3. Generative model of multi-frame sequences

Learning from spatially and temporally dense 4D human motion data is challenging. First, 4D data are high-dimensional consisting of hundreds of frames containing thousands of vertices each. In addition, raw capture data is unstructured, which makes even comparing individual frames difficult, to say nothing of multi-frame sequences. Learning from unstructured 4D data directly, *e.g.* using a network that consumes a sequence of captured scans, would require the network to learn how to compare frames while learning variation in morphology, pose, global displacement and temporal evolution. Consequently, training would require significant computational resources, and large datasets, which make this approach impractical. To overcome this challenge, we choose to leverage existing shape spaces of static 3D human bodies, which allow for a low-dimensional representation of multi-frame sequences by concatenating per-frame information. Unlike existing works that use such per-frame representations, we explicitly include information on global displacements and temporal information. This allows the network to learn variations and subsequently generalize along these dimensions. We choose a representation that explicitly decouples different factors of variation including pose, global displacement and temporal information, thereby helping the network to learn a structured latent space.

The second challenge when learning to generate multi-frame sequences is that the model needs to capture the influence of morphology on body motion. We propose the first solution to this problem based on a simple architecture. In particular, we propose a modified version of the Conditional Variational Autoencoder (CVAE) [38] that consumes multi-frame 4D human motion sequences, and conditions the decoder on morphology. This allows to learn a single latent *motion space* that encodes multi-frame 4D sequences. We demonstrate experimentally that this latent space is structured, and allows for meaningful interpolations along different axes of variation.

The following gives details on the representation, the architecture, and the training used to learn our model.

### *3.1. Representation for multi-frame 4D sequences*

We leverage existing static shape spaces developed for 3D human body surfaces to reduce the dimensionality of a multi-frame 4D sequence. Note that such a sequence always shows a single morphology deforming over time. This information can be exploited in the representation by building upon models that decouple the influence of morphology and pose for static data, and by holding morphology constant over the multi-frame sequence. Various models with this decoupling property have been proposed (*e.g.* [13, 1, 14]). They all represent a static body surface $f$ using three parameter vectors: morphology $\beta$, skeleton joint information $\theta$, and global translation $\gamma$. The models allow to generate static bodies with a function that takes input parameters $(\beta, \theta, \gamma)$. In our implementation, we leverage the commonly used SMPL model [14] as the AMASS dataset [9] we use for training is parameterized by this model.

Let $S = [f_1, f_2, \ldots, f_n]$ be a 4D human motion sequence consisting of $n$ 3D frames $f_i$. SMPL allows to represent $S$ using a single morphology vector $\beta$ along with per-frame pose and displacement vectors $\Theta = [\theta_1, \theta_2, \ldots, \theta_n]$ and $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]$, respectively. This representation alone does not include information on the temporal evolution of the motion. Since we wish to capture this information in our latent motion space to allow for generalization along this axis, we additionally include a sequence of per-frame timestamps $\Phi = [\phi_1, \phi_2, \ldots, \phi_n]$ in our representation. This results in a representation $[\beta, \chi]$ with motion information $\chi = [\Theta, \Gamma, \Phi]$ per sequence $S$. Note that in contrast to existing works, we model $\Gamma$ and $\Phi$ in the multi-frame sequence representation to retain variation in global displacement (*e.g.* walking backward or forward) and temporal evolution (*e.g.* walking or running), with the goal of generalizing along these dimensions. The time-stamped frames allow the network to place freely and on any time span length the reconstructed data.

In practice, we represent parameters $\beta$ and $\Gamma$ as in the original SMPL model. We chose to represent the rotations in $\Theta$ with a 6D vector that models rotations in a continuous manner and was shown to outperform other commonly used representation like quaternions or axis angles when training neural networks [39].

It remains to outline how to map a given raw motion sequence to this representation. To this end, we fit the SMPL model to the sequence, thereby spatially aligning the data. In our experiments, we use existing solutions to this problem [40, 9], and more details on the fitting are provided in §4. In the following, we denote such a sequence of spatially aligned meshes by $M = [m_1, m_2, \ldots, m_n]$.

All sequences used for training need to be comparable in terms of the action that is performed. To achieve this, all training sequences are segmented and pre-

processed as described in §4.1. We do not temporally align the training data in this work, as we found experimentally that an alignment with dynamic time warping [41] leads to similar results as uniformly sampling the sequences.

### *3.2. Architecture*

Our goal is to generate multi-frame 4D human motion. One interesting aspect is to learn the relationship between morphology and spatio-temporal motion patterns. To build generative models, variational autoencoder architectures were shown to be highly effective. Furthermore, the Conditional Variational Autoencoder (CVAE) architecture [38] allows to condition both encoder and decoder on input variables, thereby learning conditional distributions.

Our architecture is inspired by CVAE, and shown in Fig. 2. Note that unlike prior works, our architecture consumes multi-frame sequences, thereby learning a latent *motion space*. In particular, the motion vector $\chi$ is encoded into a low-dimensional latent vector $z$, and the morphology representation $\beta$ is used as condition. However, unlike the original CVAE, we do not condition the encoder with $\beta$ and only learn the distribution of $(z \mid \chi)$, as in a standard VAE. The decoder takes both $\beta$ and $z$ as input to reconstruct $\chi$ to learn the distribution of $(\chi \mid z, \beta)$, as in CVAE. The reason for this change is that conditioning the encoder on $\beta$ caused a loss in performance. We believe this is because unlike in CVAE, where conditions are discrete, our morphology condition is continuous. For training, we have only few motion sequences for the same $\beta$, yet many different values of $\beta$, and hence learning a latent distribution for each $\beta$ separately is not feasible. The decoder on the other hand is still conditioned by the morphology information, thereby allowing to capture the dependencies between $\chi$ and $\beta$.

As usual for VAEs, encoder outputs are interpreted as mean $\mu$ and standard deviation $\sigma$ of the prior distribution of the latent space, and the corresponding latent vector $z$ is randomly sampled as $z = \mu + \epsilon \times \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$.

The decoder takes as input $z \in \mathbb{R}^{64}, \beta \in \mathbb{R}^{8}$, and directly outputs $\hat{\chi} = (\hat{\Theta}, \hat{\Gamma}, \hat{\Phi})$ which are converted back to a sequence of meshes by applying SMPL to each frame.

### *3.3. Training*

The network is trained as a classical VAE with a two term loss: a reconstruction term which represents the difference between the input and output vectors, and a regularization term to constrain the latent variables to follow a known prior distribution. The training is divided into two phases. First, we consider a reconstruction loss on the spatio-temporal representation $\chi$ and second, we replace this

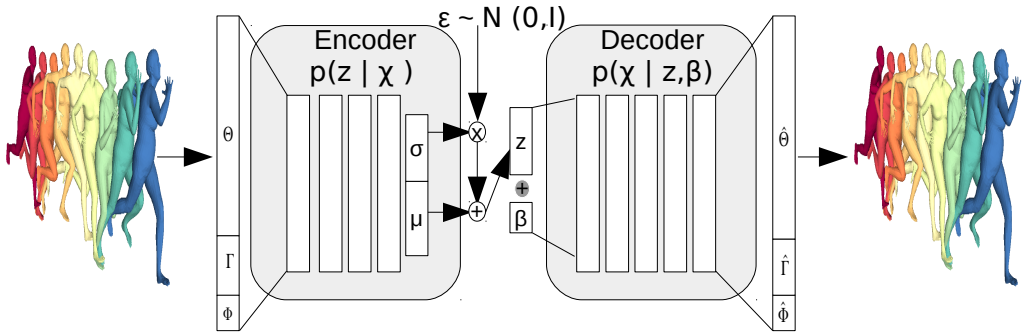


Figure 2: Architecture: Input and output are 4D sequences, mapped to a *single* latent vector $z$. The decoder is conditioned by the morphology $\beta$. $\chi$ is the concatenation of timestamp $\phi$, joint rotations $\Theta$ and translation $\Gamma$. Encoder and decoder are a succession of fully connected layers with ReLU activations.

loss by a loss computed directly on the sequence of meshes $M$ in $\mathbb{R}^3$. Considering a reconstruction loss on $\chi$ first allows for a fast and memory efficient initialization.

*Reconstruction loss on $\chi$.* The standard reconstruction term would be the squared $L_2$ distance between $\chi$ and its reconstruction $\hat{\chi}$. To balance the influence of the different types of information captured by the motion representation, we divide this loss into three terms: one on pose $\mathcal{L}_{pose} = \left(\Theta - \hat{\Theta}\right)^2$, one on translation $\mathcal{L}_{trans} = \left(\Gamma - \hat{\Gamma}\right)^2$, and one on time $\mathcal{L}_{time} = \left(\Phi - \hat{\Phi}\right)^2$.

To minimize these three losses, which do not have the same numerical magnitude, we use adaptive weights to trade off their relative influence [42]. In particular, the weights assigned to the partial losses are updated during training based on the norm of the gradient of the partial loss and a learning rate. This ensures that the losses are decreasing in similar proportions. This gives a total reconstruction loss

$$\mathcal{L}_{rec} = \omega_{pose}\mathcal{L}_{pose} + \omega_{trans}\mathcal{L}_{trans} + \omega_{time}\mathcal{L}_{time}, \tag{1}$$

where $\omega_{pose}, \omega_{trans}$ and $\omega_{time}$ are the respective adaptive weights for the reconstruction losses.

*Reconstruction loss in 4D.* The second reconstruction loss that we consider is the squared $L_2$ loss between the sequence of meshes $M = [m_1, m_2, \ldots, m_n]$ and its reconstruction $\hat{M}$. Therefore the contributions of the losses $\mathcal{L}_{pose}$ and $\mathcal{L}_{trans}$ are merged into one spatial loss :

$$\mathcal{L}_{spatial} = \left(M - \hat{M}\right)^2, \tag{2}$$

which gives the $4D$ reconstruction term

$$\mathcal{L}_{rec4D} = \omega_{spatial}\mathcal{L}_{spatial} + \omega_{time}\mathcal{L}_{time}, \tag{3}$$

where $\omega_{spatial}$ is a new adaptive weight of the spatial loss. Optimizing this loss leads to more accurate reconstruction of the 4D multi-frame sequences because it uses the full surface information, but it comes at the cost of higher computation time.

*Regularization loss.* The regularization term we use is the squared Kullback-Leibler (KL) divergence between the learned prior distribution $\mathcal{N}(\mu, \sigma)$ of our latent variable $z$ and the standard normal distribution $\mathcal{N}(0, 1)$, denoted by $\mathcal{L}_{KL}$ in the following.

*Optimization.* A common problem when training VAEs is the weighting of the regularization loss versus the reconstruction loss. We chose to use a fixed weight factor $\omega_{KL}$ to trade off these losses. In the end, the training first optimizes the loss

$$\mathcal{L}_{init} = \mathcal{L}_{rec} + \omega_{KL}\mathcal{L}_{KL} \tag{4}$$

to provide a good initialization and optimizes

$$\mathcal{L} = \mathcal{L}_{rec4D} + \omega_{KL}\mathcal{L}_{KL} \tag{5}$$

in a second phase to refine the model by using surface information.

## 4. Evaluation

All results are also shown as supplemental videos.

### *4.1. Data*

*Training data.* Training our model requires a dataset of human motion sequences densely captured in 4D. Currently, few such datasets exist (*e.g.* [6, 40]), and they do not contain enough sequences to train our model. For this reason, we base our training set on the recently published AMASS data [9]. This dataset regroups a large set of MoCap recordings and fits the SMPL model with additional soft-tissue motions to all data. This dataset is semi-synthetic in the sense that while the data used for training is based on real captures of sparse markers, the dense 3D geometry per frame is synthesized using a model.

In particular, we construct our training dataset as subset of AMASS by extracting subsequences containing at most a cycle of leg motion. We automatically find segments in the sequences of AMASS that contain a single cycle by considering the rotation of the hips joints. We manually prepare two reference sequences, a gait cycle starting with the left foot, and another one starting with the right foot. We then compare the hip joint rotations from each subsequence of an AMASS sequence to the reference cycle using dynamic time warping as distance. If the distance is below a threshold, we consider the subsequence as motion cycle. As post-processing, we prune segments with a duration above 3 seconds or below 0.3 seconds; they are the results of bad cuts resulting in incomplete or multiple cycles. This provides us with 13121 sequences of motion cycles of various duration and motion types. The majority of our motions comes from walking sequences: walking forward, backwards, turning left or right. But our dataset also encompasses

sidestep, jogging, running, front slits or dancing motions as the threshold was permissive in the segmentation process. As the segmentation does not discriminate on arm motion, all these motions can have various arm movements like walking while carrying something on the head or walking using a handrail. Our training set includes hundreds of different subjects performing different kinds of motions.

To allow for efficient learning, the sequences are spatially aligned by zeroing the initial translation and we use the identity rotation as initial rotation of the root joint to be an invariant in the ground plane.

*Test data.* We consider two test datasets. One called *AMASS test set* in the following, contains 1027 sequences. AMASS originating from a collection of MoCap datasets, we took care that sequences from a dataset were not included both in training and testing splits as per author recommendation[1], we used "MPI_mosh", "SFU"and "TotalCapture" for testing, all the others for training. The second one, called *Kinovis test set* in the following, contains 4D motion sequences captured using the Kinovis multi-view platform. This dataset is an extension of the dataset used in [40]. It allows to evaluate the generalization of the model on a dataset based on densely captured 4D data where SMPL fittings are not obtained solely from motion capture data. This smaller dataset contains 37 gait cycles extracted from walking and running sequences. To pre-process this data, we first fit SMPL to the available 4D sequences and segment individual gait cycles using the technique described in §4.1. Some sequences from this test set contain less than $100$ frames, so we augmented the data to $100$ frames using linear interpolation between the 6D rotations.

### *4.2. Implementation Details*

To build our motion representation $\chi$, we discard the 2 foot joints of the skeleton because in AMASS they always have a constant rotation so they carry no information. It leaves us with a total of 20 joints. Our representation $\chi$ consists of $100$ time-stamped frames, each of which is represented by $124$ parameters (120 for $\theta$, 3 for $\gamma$ and 1 for $\phi$). To normalize the data, we normalize the translation $\gamma$ in $[-1, 1]^3$, the timestamp $\phi$ is normalized in $[0, 1]$ and we remove the identity rotation $[1, 0, 0, 0, 1, 0]$ from the 6D representation. This centering of the rotation lead to a significant gain in reconstruction accuracy compared to the classic scaling of data $\frac{\theta - \mu_\theta}{\sigma_\theta}$. We believe it is due to some components having low standard devi-

[1] https://github.com/nghorbani/amass

ation so centering the data without scaling prevents instabilities due to a division by a small number.

For both training phases we train respectively for 5000 epoch with $\mathcal{L}_{init}$ and 200 epoch with $\mathcal{L}$. We use $\omega_{kl} = 0.01$ for both steps to weigh the regularization loss and we chose a latent dimension for the motion space $\chi$ of 64, and a dimension of 8 for $\beta$. We argue these choices in section 4.3 , we believe it to be the best compromise between smooth interpolations and good reconstruction capacity.

We train the network using the pytorch library with the ADAM optimizer. When optimizing $\mathcal{L}_{init}$, we use a learning rate of $1e^{-3}$ and a batch size of $256$. When optimizing $\mathcal{L}$, we use a batch size of $16$ for memory reasons and a learning rate of $1e^{-4}$.

We used the SMPL body model implementation proposed by the AMASS authors which is an implementation of the SMPL model with hands and dynamic components (SMPL-H + DMPL [2]). However, we did not consider hands joints and DMPL as we wanted to focus on larger spatio-temporal features of the motion.

### *4.3. Ablation studies*

We now investigate the reconstruction capability and latent space regularization of the model. The reconstruction capability of the model characterizes its ability to reconstruct examples unseen during training. A reconstruction is defined by the output sequence $\hat{\chi}$ obtained from our model combined with the ground truth morphology parameters $\beta$. The latent space regularization allows for semantically meaningful interpolations between latent points and the generation of new unseen sequences.

*Reconstruction.* The reconstruction capability of the model is mainly controlled by the latent space dimension.

Fig. 3 shows the impact of the dimension of $z$ on the reconstruction error on the AMASS test set. As expected, the bigger the latent space dimension, the smaller the error. However, the error quickly diminishes until $dim(z) = 64$, then it starts to stagnate. Therefore we chose to use a latent dimension for $z$ of 64.

*Latent space regularisation.* The regularisation of the latent space is a major factor as the model should be able to generalize to interpolations between existing motions. This regularisation heavily depends on the coefficient $\omega_{KL}$ which weighs the influence of latent space regularization at the cost of reconstruction accuracy.

[2] https://github.com/nghorbani/human_body_prior/tree/master/

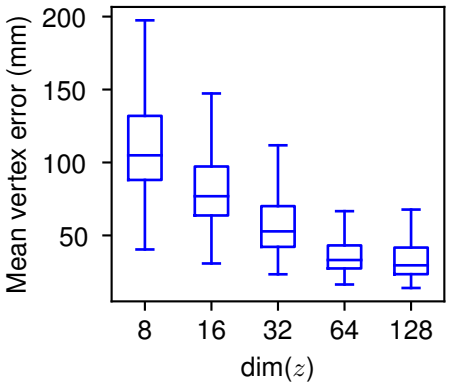


Figure 3: Plot of the mean vertex error over a sequence on the AMASS test set for models trained with different $z$ dimension. The boxes show the first, second and third quartiles of the error, the whiskers are defined following Tukey's method [43].

Fig. 4 shows the reconstruction error on models trained with different values for $\omega_{KL}$. Note that the smaller $\omega_{KL}$, the smaller the reconstruction error. However, with $\omega_{KL} = 0.001$, the model no longer allows for plausible interpolations in latent space, a problematic interpolation is shown in the supplementary material.

Therefore, we chose $\omega_{KL} = 0.01$ for our final model. Note that this value is directly connected to the scale of $\mathcal{L}_{rec4D}$ as this coefficient is not dynamic and that we used the mesh vertex positions in meters during training.

### *4.4. Comparison to baseline model*

We further evaluate the reconstruction performance of our model and compare it to a linear baseline. The baseline is a principal component analysis (PCA) fitted on the training data $[\chi, \beta]$. Note that unlike our model, PCA has access to morphology information in the "encoder" and reconstructs both $\hat{\chi}$ and $\hat{\beta}$. To provide a fair comparison, we consider the original $\beta$ instead of $\hat{\beta}$ in PCA reconstructions and set the PCA latent dimension to $dim(z) + dim(\beta)$.

Fig. 5 shows the statistics of the mean vertex error per motion sequence for PCA, our model after initialization, and our fully trained model. The PCA provides good reconstructions results which we did not expect from a linear model on our data. But our fully trained model leads to lower errors by a large margin on

`human_body_prior/body_model`

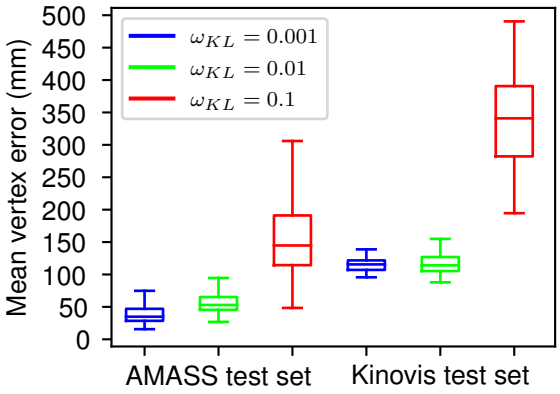


Figure 4: Plot of the mean vertex error over a sequence for three different models trained with different $\omega_{KL}$. The boxes are defined like in Fig. 5.

both test sets, demonstrating that a simple non-linear conditional model greatly outperforms the linear baseline in terms of reconstruction capabilities. Furthermore, our model improves over its initialization, demonstrating that optimizing a loss in 4D leads to more faithful results than operating on a sparse skeletal representation.

### *4.5. Motion space structure and interpolation*

Fig. 1 (top right) visualizes some samples in motion space by linearly reducing its dimensionality to two. It shows that the learned motion space is strongly structured. To further evaluate the structure of the learned latent space along with the quality of generated sequences, we interpolate between different motions chosen manually from the AMASS dataset. Both $z$ and $\beta$ are interpolated linearly. To evaluate how the model behaves spatially and temporally, we interpolate along all the different axes captured by our motion representation $\chi$, namely temporal information, global displacement, and joint poses. Fig. 6 shows the corresponding results, where a subset of all frames of the 4D sequence are rendered simultaneously and color-coded by the time at which they appear.

*Interpolation of temporal information.* To inspect the temporal information learned by our model, we interpolate between a running and a walking motion, which have both different durations and dynamics. We observe that the duration given

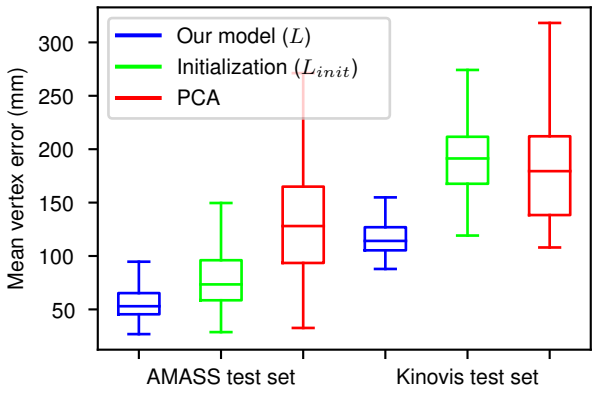


Figure 5: Plot of the mean vertex error over a sequence. The boxes are defined like in Fig. 3.

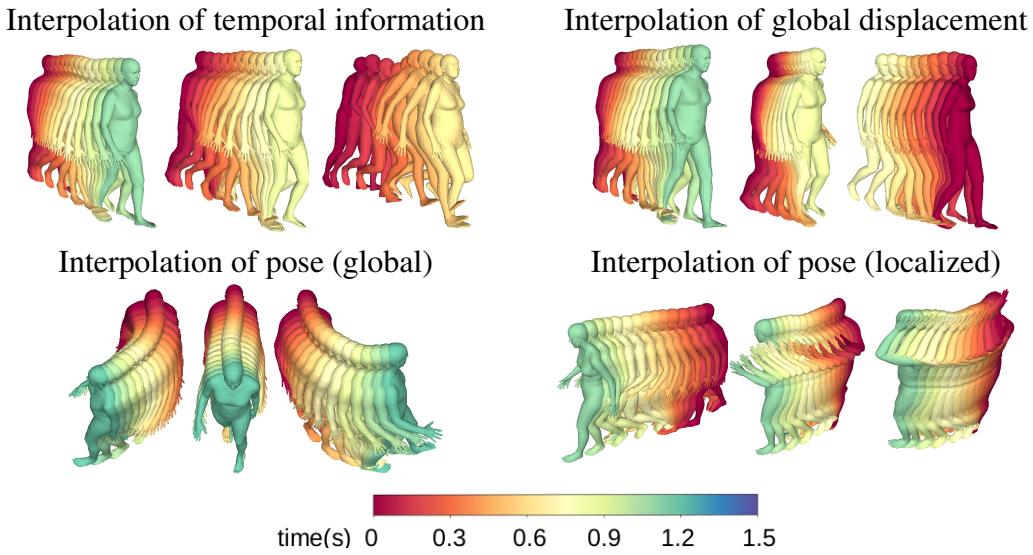


Figure 6: Each figure shows the linear interpolation in motion space (middle) between a start (left) and a target (right) 4D motion sequence color-coded by the time at which the rendered frames appear in the sequence. Each sequence corresponds to a single latent vector $z$ in motion space. Interpolations along all axes lead to semantically meaningful interpolations. **Top-Left** Interpolation between running and walking along temporal axis. **Top-Right** Interpolation between walking backward and forward along global displacement axis. **Bottom-Left** Interpolation between left and right turn along pose axis (global). **Bottom-Right** Interpolation between a walk and a walk carrying an object on the head. Arm motion is interpolated (localized).

by $\sum_{j=0}^{100} \phi_j$ of the intermediate sequences monotonically decreases when going from running to walking. The intermediate sequences are realistic in terms of their dynamic behaviour as shown in Fig. 6 (top left), which shows that the motion space has captured information on the temporal evolution of the motion and is able to generate interesting new 4D samples.

*Interpolation of global displacement.* To inspect global displacement, we interpolate between a forward and a backward walk. We observe that an intermediate sequence corresponds to a really small step as shown in Fig. 6 (top right). There were no steps this small in the training set. The latent space has captured information on $\Gamma$ and is able to generate interesting new scenarios unseen during training.

*Interpolation of pose.* To inspect the learned information of pose, we distinguish between global pose and pose articulation of the body. First, we interpolate between sequences of turning left and turning right while walking, exhibiting mostly global pose change. The intermediate sequences gradually change from a left turn to a right turn as shown in Fig. 6 (bottom left).

Second, we interpolate between a walking motion and a walking motion while carrying an object on the head. The intermediate sequence gives a realistic intermediate position for the arms, and gradually elevates them to the head level as shown in Fig. 6 (bottom right). While the interpolated motion is not natural as there is no reason to walk with the arm half-raised, the result shows that the latent space has captured information on $\Theta$, and can generate interesting motions not seen during training.

## 5. Application to 3D human motion prediction

As application, we propose to use our model for dense 4D human motion prediction. Note that human motion prediction has received considerable interest in the case of spatially sparse data (*e.g.* [26, 4]) and dense predictions from 2D video [30], as discussed in §2. However, to the best of our knowledge, we are the first to study this problem for spatially dense 4D human motion data.

Given a short sequence of spatially dense motion data, we leverage our model to predict the full sequence. We consider an observed sequence of an incomplete gait cycle $S = [f_1, ...f_k]$, and use our model to predict the following frames $[f_{k+1}, ...f_n]$.

The motion prediction optimizes latent vectors $[z, \beta]$, such that the first $k$ frames of the sequence $\hat{S}$ decoded using $[z, \beta]$ minimize the Euclidean distance to the given input sequence $S$. In this way, the first $k$ frames of the reconstruction $\hat{S}$

match $S$, and the remaining frames complete the gait cycle. We perform a simple gradient-based optimization using the adam optimizer from pytorch and the $L^2$ distance between the the first $k$ frames and their reconstruction.

To the best of our knowledge, no prior work exists on this problem, thus we adapt a state-of-the-art solution for skeletal human motion prediction [4] to our scenario as follows. We replace the skeleton used by this method by the SMPL skeleton, and run the Residual sup. method which is based on a recurrent architecture. We then combine the predicted skeletal motion with the *ground truth* morphology $\beta$, and use SMPL to obtain per-frame reconstructions. Note that this advantages [4] in two ways. First, we extend the method to make densely sampled predictions in space, which the original method does not propose. Second, we use ground truth morphology information for this extension, while our method optimizes for $\beta$.

Both models use the full training data. To feed the data to the recurrent architecture, we convert it to a fixed framerate of 100fps. For both models, we give 0.25s of motion as input. Note that this test scenario is adapted for both [4] and ours, since the recurrent model can only predict future frames while our model can complete sequences (*i.e.* add past and future frames) as long as the given input frames are labelled with a consistent set of time-stamps.

Fig. 7 shows the mean vertex error over time between the original and the reconstructed sequences. The mean at time $t$ is computed over all the vertices of frame $t$ for all test sequences of length at least $t$. Our model outperforms the recurrent model for this application.

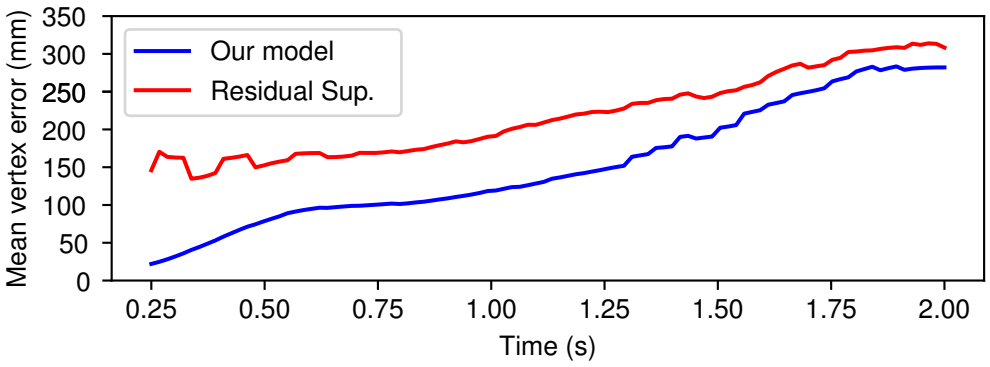


Figure 7: Comparison to Residual Sup. [4] on the AMASS test set. The first 0.25s are given as input.

## 6. Application to motion completion from spatio-temporally sparse input

This section demonstrates our model's performance for spatio-temporal completion. Given a set of sparse, unmatched and temporally inoherent set of points, our model can retrieve a spatio-temporally aligned sequence by leveraging its learned representation of human surface motion priors.

### *6.1. Completion dataset*

We introduce a new dataset of cyclic human motion (CHUM). The data was captured using a 4D modeling platform with 68 RGB cameras and a standard Qualisys motion capture system. Data consists of dense scans of approximately 10000 points acquired at 50 frame per second with synchronised MoCap data for 16 markers.

We recorded 4 actors with different morphologies (2 men and 2 women) performing various cyclic motions like walking, running, side-stepping of skipping. For each motion, we provide RGB images, 3D textured meshes and synchronised MoCap data with 16 markers.

For our experiment, we segmented 4 gait cycles manually for each original sequence and we found an initial 3D transformation (rotation + translation) to align each segment. Finally, we sampled 100 frames from each sequence. For sequences with less than 100 frames we uniformly duplicate frames until we reach 100 frames. Note that for this application, we do not need to fit SMPL to the dense scans.

### *6.2. Completion methodology*

The input considered is an incomplete sequence of frames $S = [f_1, ..., f_n]$, where each frame is a (sparse) set of 3D points without any temporal correspondence, with associated time stamps. An optional input can be given in the form of motion capture markers that provide correspondence information over time. We wish to find latent vectors $(z, \beta)$ that describe the observations well, which automatically completes missing information in space and time.

We follow a similar methodology as in Section 5 and optimize for a latent vector $(z, \beta)$ that best fits the available data. To do so, we minimize the loss

$$\mathcal{L}_{completion} = \omega_{dense}\mathcal{L}_{dense} + \omega_{mocap}\mathcal{L}_{mocap} + \omega_{time}\mathcal{L}_{time}, \tag{6}$$

where $\mathcal{L}_{dense}$ is the Chamfer distance between the sequence generated by the model and $S$, $\mathcal{L}_{mocap}$ is the $L^2$ distance between the captured motion capture markers and 16 manually selected vertices of the SMPL template, and $\mathcal{L}_{time}$ is

the $L^2$ loss on the timestamps. The weights $\omega_{dense}$, $\omega_{mocap}$ and $\omega_{time}$ are adaptive [42]. The weight $\omega_{mocap}$ is set to zero when no markers are available and the weight $\omega_{dense}$ is set to zero when no dense data is available. In the following, this variation of $\omega_{mocap}$ allows to evaluate the benefit of having corresponding points over time for the completion task.

In the following, we evaluate the accuracy of our model to reconstruct dense 4D data based on partial sparse input data. The reconstruction error is computed using the Chamfer distance between our result and the densely captured 4D sequences.

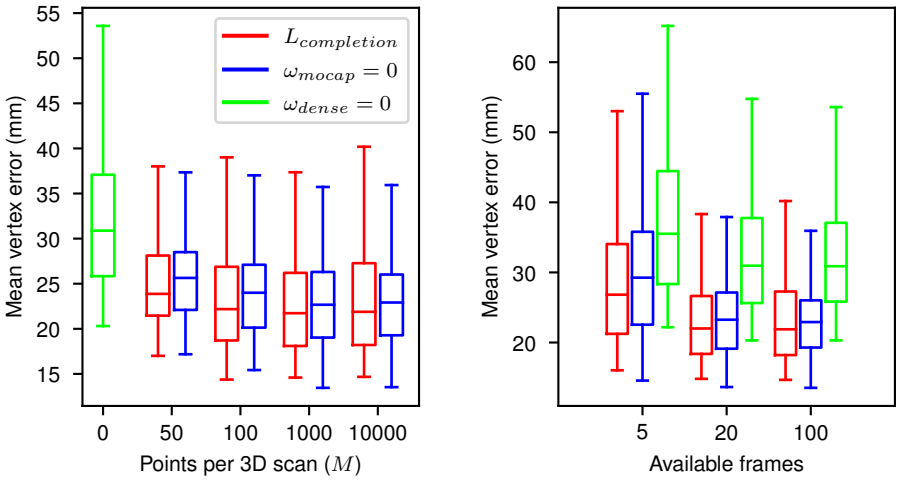


Figure 8: Mean chamfer distance between completed sequence and dense scans. In red, MoCap information and scans are used in $L_{completion}$, in blue, no MoCap information is used and in green, no dense information is used. The boxes are defined like in Fig. 3.

*Spatial completion.* We first evaluate the quality of spatial completion by varying the number of 3D points used in the loss $\mathcal{L}_{dense}$ and by measuring the influence of activating the loss $\mathcal{L}_{mocap}$ in Equation 6. To compute $\mathcal{L}_{dense}$, we randomly sampled a fixed number of points $M$ per captured scan to simulate spatially sparse data. Note that the sampled points are not in correspondence across time. Fig. 8 shows the evolution of the reconstruction error when varying $M$. The red plots show the results obtained when using the loss of Equation 6 for varying numbers of $M$. The green plot shows the case for $\omega_{dense} = 0$. The blue plots show the case where $M$ varies and $\omega_{mocap} = 0$. While the results improve overall for increasing observations (*i.e.* larger $M$), for $M = 50$, the reconstruction error is

already around $25mm$. This quality is achieved because our model optimizes for all frames simultaneously, so few points per scan are sufficient to find a plausible solution. An example of the completion task with $M = 100$ is shown in Figure 9.

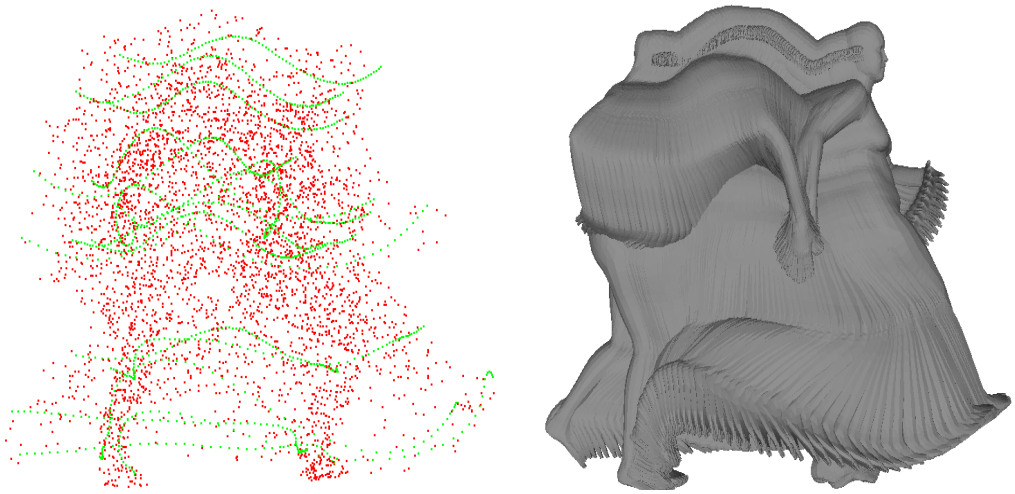

Figure 9: On the left, the sparse scan for $M = 100$ (red) and the MoCap data (green) are shown, . On the right, the completed sequence obtained when optimizing $L_{completion}$ is shown. The original sequence is a kick sequence from CHUM.

*Temporal completion.* Second, we evaluate the quality of temporal completion by varying the number of input frames used in all losses. We again measure the influence of $\mathcal{L}_{mocap}$ in this scenario. To vary the number of input frames, we uniformly sample a fixed number of frames from each test sequence. Fig. 8 shows the evolution of the reconstruction error. The 100 frame completion task which includes all the frames is given as a reference, the model extrapolates with almost no loss of precision when given only a fifth of the frames (20 frames) and the error is still low even when extrapolating from a twentieth of the frames (5 frames). This shows that our spatio-temporal model allows for accurate completions from captures acquired at low frame rates.

*Spatio-temporal completion.* Finally, we evaluate our model's completion quality for data captured sparsely in both time and space. To this end, we downsample our test sequences both spatially and temporally, and optimize the losses. Table 1 summarizes the reconstruction errors. For reference, the result when optimizing with $\omega_{dense} = 0$ is shown in green font, and constant over varying numbers of $M$. The result when setting $\omega_{mocap} = 0$ is shown in blue font, and the result

with the full loss including motion capture markers in red font. Note that even when completing both spatially and temporally, the model still performs well. The motion capture information gives useful guidance when there are few points per scan but this is not necessary for denser scans.

Table 1: Mean error over the test set (in $mm$) for the different combinations and optimization losses. In red, $L_{completion}$ was used with dense and mocap data, in blue, $\omega_{mocap}$ was set to 0 and in green, $\omega_{dense}$ was set to 0

| Points per scan $M$ \ Number of frames | 5 | 20 | 100 |
|---|---|---|---|
| 0 | 48 | 44 | 43 |
| 50 | 33, 36 | 26, 27, | 26, 26 |
| 100 | 31, 33 | 25, 25 | 24, 24 |
| 1000 | 29, 30, | 24, 23 | 24, 23 |
| 10000 | 30, 26 | 25, 21 | 24, 21 |

## 7. Conclusions and future work

This work presented a first 4D human body motion generator that represents multi-frame sequences in a single latent space. We demonstrated that this latent motion space is structured and allows to synthesize meaningful new motions by interpolation. We further applied this model to motion prediction.

For future work, it will be interesting to study how our model can be leveraged for different applications. For instance, our model can directly be fit to spatially and temporally sparse data for completion applications by optimizing using the learned decoder. Examples include reconstructing dense 4D human motion from sparse MoCap data available for few frames. Furthermore, as the model captures the relationship between morphology and motion, it can potentially be leveraged to advance the state-of-the-art in tasks such as motion transfer.

This study focused on locomotions with cyclic leg movements. In the future, we would like to investigate the inclusion on a wider variety of action to take full advantage of the variety of motions available in the AMASS dataset.

## 8. Acknowledgements

We thank Jinlong Yang and Jiabin Chen for providing us the Kinovis test set, Joao Regateiro for helpful discussions, and the Kinovis platform at Inria Grenoble,

the engineers Laurence Boissieux and Julien Pansiot, and our volunteer subjects for help with the 4D data acquisition. This work was partially funded by ANR grant 3DMOVE - 19-CE23-0013-01.

## References

[1] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, B. Schiele, Building statistical shape spaces for 3d human modeling, Pattern Recognition 67 (2017) 276–286. 2, 3, 5, 8

[2] N. Troje, Retrieving information from human movement patterns, in: Understanding Events: From Perception to Action, Oxford University Press, 2008, pp. 308–334. 2, 5

[3] L. Sigal, D. Fleet, N. Troje, M. Livne, Human attributes from 3d pose tracking, in: European Conference on Computer Vision, 2010. 2

[4] J. Martinez, M. Black, J. Romero, On human motion prediction using recurrent neural networks, in: Conference on Computer Vision and Pattern Recognition, 2017. 2, 5, 18, 19

[5] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, J. Davis, SCAPE: shape completion and animation of people, ACM Transactions on Graphics 24 (3) (2005) 408–416, proceedings of SIGGRAPH. 3, 5, 6

[6] G. Pons-Moll, J. Romero, N. Mahmood, M. Black, DYNA: a model of dynamic human shape in motion, ACM Transactions on Graphics 34 (4) (2015) 120:1–14. 3, 5, 12

[7] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, C. Sminchisescu, GHUM & GHUML: Generative 3d human shape and articulated pose models, in: Conference on Computer Vision and Pattern Recognition, 2020. 3, 5

[8] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. J. Black, Smpl: A skinned multi-person linear model, ACM transactions on graphics (TOG) 34 (6) (2015) 1–16. 3

[9] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, M. J. Black, Amass: Archive of motion capture as surface shapes, in: Proceedings of the IEEE

International Conference on Computer Vision, 2019, pp. 5442–5451. 4, 6, 8, 12

[10] B. Allen, B. Curless, Z. Popović, The space of human body shapes: reconstruction and parameterization from range scans, ACM Transactions on Graphics 22 (3) (2003) 587–594, proceedings of SIGGRAPH. 5

[11] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, H.-P. Seidel, A statistical model of human pose and body shape, Computer Graphics Forum (Special Issue of Eurographics 2008) 2 (28) (2009) 337–346. 5

[12] Y. Chen, Z. Liu, Z. Zhang, Tensor-based human body modeling, in: Conference on Computer Vision and Pattern Recognition, 2013, pp. 105–112. 5

[13] A. Neophytou, A. Hilton, Shape and pose space deformation for subject specific animation, in: 3DV, 2013. 5, 8

[14] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. Black, SMPL: A Skinned Multi-Person Linear Model, Transactions on Graphics 34 (6) (2015) 248:1–248:16, proceedings of SIGGRAPH ASIA. 5, 8

[15] H. Joo, T. Simon, Y. Sheikh, Total capture: A 3d deformation model for tracking faces, hands, and bodies, in: Conference on Computer Vision and Pattern Recognition, 2018. 5

[16] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, M. J. Black, Expressive body capture: 3d hands, face, and body from a single image, in: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10975–10985. 5

[17] I. Santesteban, E. Garces, M. A. Otaduy, D. Casas, SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans, Computer Graphics Forum (Proc. Eurographics) (2020). 5

[18] B. Jiang, J. Zhang, J. Cai, J. Zheng, Disentangled human body embedding based on deep hierarchical neural network, IEEE Transactions on Visualization and Computer Graphics (2020). 5

[19] L. Cosmo, A. Norelli, O. Halimi, R. Kimmel, E. Rodolà, Limp: Learning latent shape representations with metric preservation priors, in: European Conference on Computer Vision, 2020. 5

[20] N. Troje, Decomposing biological motion: A framework for analysis and synthesis of human gait patterns, Journal of Vision 2 (2002) 371–387. 5

[21] C. Rose, M. F. Cohen, B. Bodenheimer, Verbs and ad-verbs: multidimensional motion interpolation, IEEE Computer Graphics and Applications 18 (1998) 32–40. 5

[22] D. Holden, J. Saito, T. Komura, A deep learning framework for character motion synthesis and editing, Transactions on Graphics 35 (4) (2016) #138. 5

[23] H. S. Koppula, A. Saxena, Learning spatio-temporalstructure from rgb-d videos for human activity detection andanticipation, in: ICML, 2013. 5

[24] R. Urtasun, D. J. Fleet, P. Fua, 3d people tracking with gaussian process dynamical models, in: CVPR, 2006. 5

[25] L. Kovar, M. Gleicher, F. Pighin, Motion graphs, Transactions on Graphics 21 (2002). 5

[26] K. Fragkiadaki, S. Levine, P. Felsen, J. Malik, Recurrent network models for human dynamics, in: Conference on Computer Vision and Pattern Recognition, 2015. 5, 18

[27] M. Kaufmann, E. Aksan, J. Song, F. Pece, R. Ziegler, O. Hilliges, Convolutional autoencoders for human motion infilling, in: Conference on 3D Vision, 2020. 5

[28] M. Loper, N. Mahmood, M. Black, Mosh: Motion and shape capture from sparse markers, ACM Transactions on Graphics 33, proceedings SIGGRAPH Asia (2014). 6

[29] J. Y. Zhang, P. Felsen, A. Kanazawa, J. Malik, Predicting 3d human dynamics from video, in: The IEEE International Conference on Computer Vision (ICCV), 2019. 6

[30] A. Kanazawa, J. Y. Zhang, P. Felsen, J. Malik, Learning 3d human dynamics from video, in: Computer Vision and Pattern Recognition (CVPR), 2019. 6, 18

[31] A. Jain, T. Thormählen, H.-P. Seidel, C. Theobalt, MovieReshape: tracking and reshaping of humans in videos, ACM Transactions on Graphics 29 (2010) 148:1–10, proceedings of SIGGRAPH Asia. 6

[32] A. Kuznetsova, N. Troje, B. Rosenhahn, A statistical model for coupled human shape and motion synthesis, in: Conference on Computer Graphics Theory and Applications, 2013. 6

[33] A. Boukhayma, E. Boyer, Surface motion capture animation synthesis, Transactions on Visualization and Computer Graphics (2018). 6

[34] J. Regateiro, A. Hilton, M. Volino, Dynamic surface animation using generative networks, in: International Conference on 3D Vision, 2019. 6

[35] I. Akhter, T. Simon, S. Khan, I. Matthews, Y. Skeikh, Bilinear spatiotemporal basis models, ACM Transactions on Graphics 31 (2012) #17:1–12. 6

[36] S. Ghorbani, C. Wloka, A. Etemad, M. Brubaker, N. Troje, Probabilistic character motion synthesis usin a hierarchical deep latent variable model, in: Symposium on Computer Animation, 2020. 6

[37] M. Niemeyer, L. Mescheder, M. Oechsle, A. Geiger, Occupancy flow: 4d reconstruction by learning particle dynamics, in: International Conference on Computer Vision, 2019, pp. 5379 – 5389. 6

[38] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in: Advances in neural information processing systems, 2015, pp. 3483–3491. 7, 9

[39] Y. Zhou, C. Barnes, J. Lu, J. Yang, H. Li, On the Continuity of Rotation Representations in Neural Networks, in: Proceedings of CVPR, IEEE, Long Beach, CA, 2019, p. 9. 8

[40] J. Yang, J.-S. Franco, F. Hétroy-Wheeler, S. Wuhrer, Estimation of human body shape in motion with wide clothing, in: European Conference on Computer Vision, 2016. 8, 12, 13

[41] D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: International Conference on Data Mining and Knowledge Discovery, 1994, pp. 359–370. 9

[42] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, Stockholmsmässan, Stockholm Sweden, 2018, pp. 794–803. 11, 21

[43] J. W. Tukey, Box-and-whisker plots, Exploratory data analysis (1977) 39–43. 15