



HAL
open science

Representing motion as a sequence of latent primitives, a flexible approach for human motion modelling

Mathieu Marsot, Stefanie Wuhrer, Jean-Sebastien Franco, Anne H el ene Olivier

► To cite this version:

Mathieu Marsot, Stefanie Wuhrer, Jean-Sebastien Franco, Anne H el ene Olivier. Representing motion as a sequence of latent primitives, a flexible approach for human motion modelling. Inria centre at the University Grenoble Alpes. 2022. hal-03715820v2

HAL Id: hal-03715820

<https://hal.science/hal-03715820v2>

Submitted on 16 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv es.



Distributed under a Creative Commons Attribution 4.0 International License

Representing motion as a sequence of latent primitives, a flexible approach for human motion modelling

Mathieu Marsot¹ Stefanie Wuhrer¹ Jean-Sébastien Franco¹ Anne-Hélène Olivier²

¹ Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP*, LJK, 38000 Grenoble, France

² Univ. Rennes, Inria, CNRS, IRISA, M2S, 35000 Rennes, France

firstname.lastname@inria.fr

Abstract

We propose a new representation of human body motion which encodes a full motion in a sequence of latent motion primitives. Recently, task generic motion priors have been introduced and propose a coherent representation of human motion based on a single latent code, with encouraging results for many tasks. Extending these methods to longer motion with various duration and framerate is all but straightforward as one latent code proves inefficient to encode longer term variability. Our hypothesis is that long motions are better represented as a succession of actions than in a single block. By leveraging a sequence-to-sequence architecture, we propose a model that simultaneously learns a temporal segmentation of motion and a prior on the motion segments. To provide flexibility with temporal resolution and motion duration, our representation is continuous in time and can be queried for any timestamp. We show experimentally that our method leads to a significant improvement over state-of-the-art motion priors on a spatio-temporal completion task on sparse pointclouds.

1. Introduction

3D human body motion modelling is an underlying issue in many computer vision and graphics problems like 3D pose estimation, 4D completion from sparse inputs, character animation or motion generation. These ill-posed problems often require prior knowledge on human motion to generate plausible solutions. Recently, several works have proposed to learn task generic priors of 3D human body motion by capturing information about pose changes over time [25, 11, 15] which have shown great performances on many applications. We propose to improve on two aspect of these repre-

sentations. First, we propose more flexibility in temporal resolution and motion duration, which makes our method easier to use as it does not need to be retrained from scratch for new inputs. Second, the reconstruction quality of existing priors tends to degrade quickly when considering longer motions. Combining temporal segments help to alleviate this degradation, but discontinuities in motion can appear due to motion averaging or normalization. To improve on this aspect, we learn a latent space on temporal segments jointly with segmentation parameters that allow for smooth transitions. This allows us to model a large variety of motion without restriction.

Our hypothesis is that motion is better represented as a sequence of latent primitives than in a single one. Making an analogy between actions and latent primitives, in a dataset of x different actions, a motion prior needs x^y different latent codes to represent all possible y element sequences of actions while a sequential motion prior requires only x latent codes. As y increases with longer motions, the number of latent codes grows exponentially with motion duration for classical motion priors but remains constant in a sequential representation.

Our model leverages a sequence-to-sequence (seq2seq) architecture that is flexible w.r.t. the sequence length of the input and allows to directly encode the motion into a sequence of latent primitives. We then decode the sequence of latent primitives using a decoder implicit in time that outputs a parametric 3D human body model for any given time instant.

We show that this sequential prior has better generalization capacity than a baseline using a single latent code on a synthetic dataset. Additionally, we show that our method generalizes to motion duration outside the training set, and provides state-of-the-art results for a spatio-temporal completion task on sparse unordered point cloud from data acquired in a multi-view studio.

In summary, our main contributions are

*Institute of Engineering Univ. Grenoble Alpes

- A novel motion representation using a sequence of latent primitives.
- An implicit representation of the temporal dimension allowing for flexible temporal resolution.
- An ablation and a comparative study showing significant improvement w.r.t. existing motion priors.

2. Related Work

Human motion has extensively been studied in disciplines including computer vision, animation, perception, and human-computer interaction with a variety of goals. We focus our review on works that propose motion priors for 3D human motion data densely captured in space and time.

First such models have focused on motion as a sequence of static poses. For instance, when generating dense human motion from sparse MoCap [2, 12, 14, 5] or from 2D video data [8, 26], these works use per frame input marker points or 2D images as input to reconstruct dense 4D motion data. In this way, a statistical human body model has been fitted to a corpus of motion, providing the community access to a large 4D dataset, which we leverage in our work [14].

Recently, there has been a surge of interest for data driven motion priors, with first approaches focusing on a few motion sequences or multiple sequences of the same actor [1, 3, 20, 21].

Following these works, there were attempts at generalizing to different body shapes and larger variety of motion. One line of work focuses on implicit spatio-temporal representations [16, 23], with the advantage of being flexible w.r.t. spatial and temporal sampling of the input data. These works have been specialized to human motions [6, 7], but as they do not reduce the spatial dimensionality of the input, they are restricted to small temporal spans.

Another line of work closer to ours uses spatially aligned data, where the input is aligned on a statistical human body model. These works focus mostly on skeletal motion, which has a low dimensionality, and allows to represent longer temporal spans. Some priors were presented in a task specific setting, *e.g.* 3D pose estimation from monocular video [9, 27] or motion generation [22, 18, 4]. Two of these methods consider problems that are different from ours, but present ideas that inspired our work. Ghorbani *et al.* [4] synthesize motion variations by subdividing long motions into smaller segments. Subdividing motion into segments is an idea which we leverage in our work. Petrovich *et al.* [18] synthesize motions of a specific given action, and explicitly model the duration of a motion.

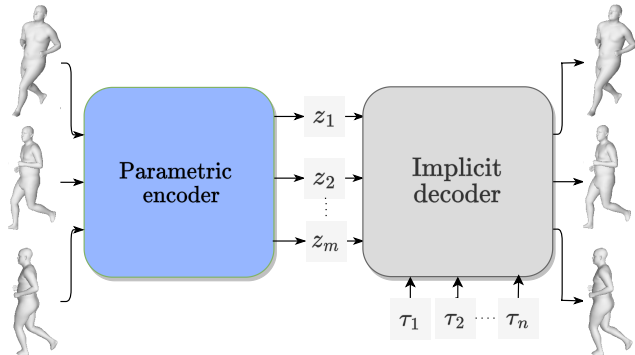


Figure 1. Method overview. Architecture consists of a seq2seq encoder (blue) that maps a human motion sequence into a sequence of latent primitives z_1, \dots, z_m , and a temporally implicit decoder (grey block) that decodes z_1, \dots, z_m and a series of timestamps τ_1, \dots, τ_n into a n -element sequence of parametric human body models.

We also explicitly model segment duration in our approach.

Finally some priors were presented as task generic [11, 25, 15] with encouraging results on different tasks with a single model. Closest to our work, these priors focus on encoding motion in a latent space and show that this representation can be leveraged in tasks such as 3D pose estimation or 4D motion completion. To encode motion, these methods either consider motion of fixed duration and framerate [11, 25] or temporally aligned motion using a dynamic time warping criterion [15]. These priors can model up to 3s motion when considering global displacement [15] and 4s motion without considering global displacement [25], but they do not generalize easily to longer motion sequences. We show experimentally that our approach outperforms two of these methods for the task of motion completion [11, 25]; the third method [15] is limited to motions performing a cyclic hip motion and cannot be applied in our scenario of various motion.

3. Overview

Figure 1 provides a visual overview of our method. We use a data driven approach to learn our motion representation. To deal with the high dimensionality of the spatio-temporal data used for training, we leverage a low-dimensional representation for each frame based on a human body model.

Given as input a sequence of human motion with variable duration and number of frames, our goal is to encode the input directly in a sequence of latent primitives. To address this sequence to sequence problem, we choose the transformer architecture [24]. This choice was guided by the transformer capability to bet-

ter retain long term correlations than its recurrent network counterparts.

To generate coherent output motions at arbitrary temporal resolution, we decode each latent primitive independently in a temporally implicit way, thereby converting the latent sequence to a sequence of parametric human body models for any given time instant. Decoding the primitives independently allows the model to build a single common latent space of motion. We want the model to allocate each primitive to a temporal segment of the motion, without requiring handcrafted segmentation parameters, so the per-primitive outputs are combined using a weighted average computed using temporal masks. The temporal masks ensure that each primitive is allocated to a continuous temporal window of the input motion.

The model is trained as a conditional variational autoencoder (CVAE) using the shape conditioning proposed in prior works [15] to retain correlations between body shape and motion.

4. Method

This section provides a formal description of the representation and architecture proposed to learn a sequence of latent primitives that represent human motion of variable duration, and provides details on the training.

4.1. 4D sequence representation

We are interested in modelling a large variety of motions performed by different subjects. To do so, we leverage the AMASS dataset [14]. AMASS is an aggregation of multiple synthetic motion capture data for which a parametric body model is provided.

We consider body shape to remain constant over time, which allows to represent a motion sequence using a set of body shape parameters β , the joint rotations of a skeleton $\theta(\tau)$, and a 3D coordinate vector characterizing the displacement of the root joint $\gamma(\tau)$, where τ is the parameter controlling time. For our implementation, we use the SMPL body model [13] provided with the dataset. More details are given in 5.1.

We call $M(\theta(\tau), \gamma(\tau), \beta)$ the function that outputs the template aligned mesh corresponding to the parametric representation $\theta(\tau), \gamma(\tau), \beta$. A discretized motion sequence consisting of n frames is then characterized by the sequence $\{M(\theta(\tau_i), \gamma(\tau_i), \beta), \tau_i\}_{i=1}^n$, where τ_i are the time stamps corresponding to the meshes.

4.2. Transformer encoder

We represent a 4D human motion sequence using two independent factors: a sequence of m latent primitives $\{z_i\}_{i=0}^m$, and body shape β . To allow for flexible

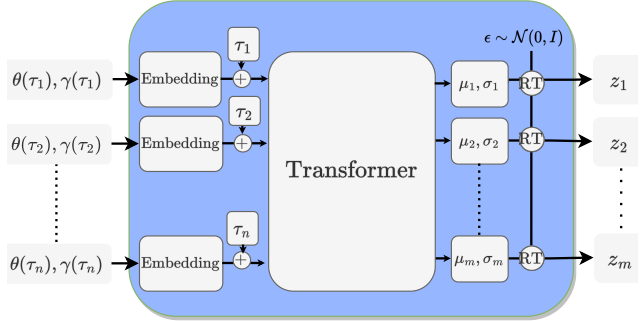


Figure 2. Encoder mapping an input sequence into a sequence of latent motion primitives z_1, \dots, z_m . The embedding is a one layer perceptron. Time stamps τ_i are added as positional encoding. Transformer outputs a sequence latent distributions μ_i, σ_i from which z_i are sampled using the reparametrization trick (RT).

duration and frame rates, we allow an arbitrary number of input frames per sequence. The latent primitives are obtained by a transformer encoder that maps an input sequence to a sequence of latent primitives. The transformer block of the encoder operates on a reduced embedding representation of the input frames and is similar to the original transformer [24] that was used for language translation. The encoder architecture is shown in Figure 2.

The difference between language translation and our setting is that we do not have ground truth for the latent primitives. To learn this part in an unsupervised setting, we fix the number m of latent primitives but retain temporal flexibility by learning the duration δ_i of each motion primitive using a differentiable cost function.

The encoder considers a sequence of parameters $\{\theta(\tau_i), \gamma(\tau_i), \tau_i\}_{i=1}^n$. To adjust the dimensionality of this representation, for each frame $\theta(\tau_i), \gamma(\tau_i)$ are passed through one perceptron layer, and τ_i is subsequently concatenated to this representation. Note that the perceptrons applied to all frames share weights.

The output sequences of the transformer encoder are interpreted as a sequence of Gaussians defined by a sequence of means and standard deviations $\{\mu_i, \sigma_i\}_{i=1}^m$ from which the latent primitives $\{z_i\}_{i=1}^m$ are sampled using Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ such that $z_i = \mu_i + \epsilon\sigma_i$. This is similar to the interpretation of latent spaces of VAEs and known to allow for generalization.

4.3. Temporally implicit decoder

Our decoder operates in two stages by first decoding individual latent primitives and by subsequently combining them. This allows learning a motion prior on the latent primitives which each characterize a mo-

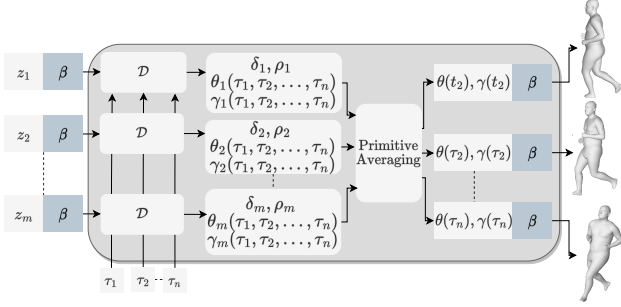


Figure 3. Implicit decoder. Given a sequence of m latent primitives z_i , a body shape β and a sequence of n timestamps τ_j , the decoder outputs a sequence of body meshes parameterized by β, θ, γ . The z_i are decoded independently into segment parameters that include duration δ_i and a rigid transformation ρ_i , and subsequently combined to decode a dense 4D motion.

tion segment, while still enforcing coherence between neighboring segments. The architecture of the implicit decoder is shown in Figure 3.

Primitive decoding First, given a body shape β , the latent primitives are decoded individually using an implicit primitive decoder $\mathcal{D}(z_i, \beta, \tau)$ that outputs per-segment motion parameterized by $\theta_i(\tau), \gamma_i(\tau)$, its duration δ_i and a rigid transformation ρ_i . $\theta_i(\tau)$ and $\gamma_i(\tau)$ characterize the global motion on the temporal segment $[\Delta_i, \Delta_i + \delta_i]$, where $\Delta_i = \sum_{j < i} \delta_j$. This ensures that each latent primitive encodes information of a continuous temporal segment of the input motion. For invariance w.r.t. the initial orientation and displacement of a segment, \mathcal{D} also outputs a rigid transformation ρ_i which is used as a transition from segment space to input space. The architecture of \mathcal{D} consists of two MLPs. The first MLP outputs per segment parametric representation $\theta_i(\tau), \gamma_i(\tau)$. The second MLP does not consider time and outputs the per segment parameters ρ_i and δ_i .

Primitive combination To combine the segment representations into motion $\theta(\tau), \gamma(\tau)$, a weighted average of the per-segment representations using temporal masks is computed. For each primitive, the corresponding Gaussian mask is $G_i(\tau) = e^{-\left(\frac{\tau - (\Delta_i + \delta_i/2)}{\delta_i/2}\right)^2}$ such that:

$$\theta(\tau) = \frac{\sum_i G_i(\tau)(\rho_i * \theta_i(\tau))}{\sum_i G_i(\tau)}, \quad (1)$$

$$\gamma(\tau) = \frac{\sum_i G_i(\tau)(\rho_i * \gamma_i(\tau))}{\sum_i G_i(\tau)}. \quad (2)$$

We denote by $\rho_i * \theta_i(\tau)$ and $\rho_i * \gamma_i(\tau)$ the operation of applying the rigid transformation ρ_i to the corresponding body model parameters. This transformation

consists of rotating the root joint for parameters θ , and rotating and translating the global displacements γ .

The temporally implicit nature of \mathcal{D} alleviates the problem of averaging segments that may not be temporally aligned according to a predefined frame rate. The averaging of rotations is done in 6D representation space [28]. This qualitatively leads to naturally combined results.

4.4. Training

The model is trained in a variational autoencoder setting with a reconstruction loss, and a Kullback–Leibler (KL) divergence loss to constrain the prior distribution to a normal distribution. We also added a regularization loss on the segment duration for faster convergence and to prevent local minima. The total loss is

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{KL} \mathcal{L}_{KL} + \lambda_{reg} \mathcal{L}_{reg}, \quad (3)$$

where

$$\mathcal{L}_{rec} = \mathcal{L}_{global} + \mathcal{L}_{segment}, \quad (4)$$

$$\mathcal{L}_{KL} = \frac{1}{m} \sum_{j=1}^m KL(\mathcal{N}(\mu_j, \sigma_j), \mathcal{N}(0, I)), \quad (5)$$

$$\mathcal{L}_{reg} = \sum_{j=1}^m (\delta_j - 1/m)^2. \quad (6)$$

The reconstruction loss is divided into two terms. The first term \mathcal{L}_{global} acts as a global reconstruction term between the input and the reconstructed output, including both a per vertex distance to capture fine details and a distance in the parametric representation.

$$\begin{aligned} \mathcal{L}_{global} = & \frac{1}{n} \sum_{i=1}^n \left(\left\| \theta(\tau_i) - \theta_{GT}(\tau_i) \right\|^2 \right. \\ & + \left\| \gamma(\tau_i) - \gamma_{GT}(\tau_i) \right\|^2 \\ & \left. + \lambda_{3D} \left\| M(\tau_i) - M_{GT}(\tau_i) \right\|^2 \right) \end{aligned} \quad (7)$$

with $\|\cdot\|$ the L2-norm, θ_{GT}, γ_{GT} the ground truth body model parameters and λ_{3D} a weighting coefficient that controls the relative influence of the per vertex distance.

The second term $\mathcal{L}_{segment}$ acts as a per segment reconstruction loss, which guarantees that each segment represents a realistic motion and allows for realistic reconstructions where segments are overlapping.

$$\begin{aligned} \mathcal{L}_{segment} = & \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m G_j(\tau_i) \left\| \rho_j * \theta_j(\tau_i) - \theta_{GT}(\tau_i) \right\|^2 \\ & + \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m G_j(\gamma_i) \left\| \rho_j * \gamma_j(\tau_i) - \gamma_{GT}(\tau_i) \right\|^2 \end{aligned} \quad (8)$$

5. Evaluation of the motion prior

We start by outlining the implementation and data used to build our model. To evaluate the motion prior, we test the generalization to sequences of duration outside of the training set, evaluate the influence of the sequential latent representation, and the segmentation learning. Finally, we leverage our prior on a spatio-temporal completion task to evaluate its benefits against state of the art motion priors. More qualitative results are provided in supplementary material.

5.1. Implementation and data

Implementation details Our method is implemented using pytorch and the Adam optimizer is used for optimization. SMPL represents a static body by 22 skeleton joints and 16 body shape parameters. We discard the foot joints which have constant rotation in AMASS, which results in 20 skeleton joints. Each joint is represented in 6D using its relative rotation to its parent joint [28]. For body shape, we use the first 8 shape components. While our method is flexible w.r.t. the number of input frames n , we fix $n = 100$ for training and train on motions of 3 – 5s by randomly sampling subsequences from the training set. During training, global displacements are normalized to $[-1, 1]$ for each direction and timestamps are scaled to $[0, 1]$. Unless stated otherwise, we set $m = 8$ and each latent vector has dimension $D = 256$. When training the motion prior, we initially use a learning rate of $1e-4$, which is reduced to $1e-5$ after 20 epochs without improvement of the training loss, and further decreased to $1e-6$ after 20 more epochs without improvement. This is done using the plateau scheduler of pytorch. During the training phase we use $\lambda_{KL} = 0.0001, \lambda_{reg} = 0.01$ and set $\lambda_{3D} = 0$ for the first 500 epochs because the 3D term slows down training significantly. Once we obtain good convergence after 500 epochs, we set $\lambda_{3D} = 1$ for 500 epochs. This significantly increases the pressure on trajectory reconstruction and gives a hierarchical importance to the joints, greatly reducing the reconstruction error in mm . We use a batch size of 16. The training phase takes between 1 and 2 days on a Geforce RTX 2080Ti with 12GB RAM.

Data splits AMASS [14] is a collection of different datasets parameterized by SMPL that contains a variety of motions and body shapes. We leave some collections of this dataset for validation ensuring that all experiment are evaluated on unseen motions and unseen body shapes.

5.2. Generalization

Our model is learned using sequences with duration 3 – 5s and $m = 8$ latent primitives. We analyze how well this model generalizes to sequences of different duration by applying it to duration 0.2–8s. To process sequences longer than 5s with our method, they are virtually accelerated by scaling the timestamps to $[0, 1]$. We perform a forward pass on test sequences of different duration and consider

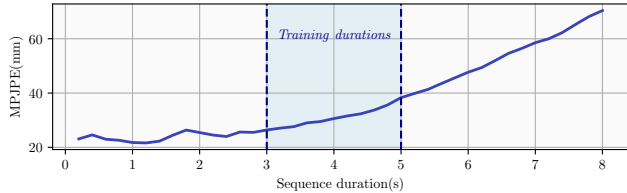


Figure 4. Generalization to sequence duration outside training set (for training, duration of 3 – 5s). Blue line shows evolution of MPJPE (lower is better) of our method for different sequence duration.

the mean per joint position error (MPJPE) between input and output joints, which is a standard metric introduced in [9]. This error is averaged over the sequence. Figure 4 shows the evolution of MPJPE for different sequence duration. Our model generalizes well to sequences of shorter duration than those used during training, and the error degrades gracefully for sequences of longer duration.

5.3. Sequential latent representation

To evaluate the value of learning a sequence of latent primitives instead of a single latent vector, we compare our method to a baseline result when setting $m = 1$. To compare the same total number of latent dimensions, we consider two models in this experiment: one trained with $m = 4$ where each segment has $D = 256$ latent dimensions and one with $m = 1$ and $D = 1024$. For evaluation, we consider the same generalization plot to sequences of different duration as before. Note that m is reduced in this setting for practical reasons as training for $m = 1$ with high latent dimension is costly. Figure 5 shows that $m = 4, D = 256$ leads to significantly lower MPJPE than $m = 1, D = 1024$ and generalizes significantly better to sequences of different duration. This shows that a sequential representation better generalizes to varying motions and duration.

5.4. Segmentation learning

To evaluate the influence of learning segment durations, we compare our method to a baseline trained with fixed segmentation parameters $\delta_i = 1/m$. As in the previous experiment, we set $m = 4, D = 256$ for both models and consider the generalization plot to sequences of different duration. Figure 5 shows that the differences between the two models are minor. Allowing for flexible segments slightly degrades performance for longer sequences; the virtual acceleration of sequences is more detrimental to the model when learning the segmentation parameters as they are more heavily influenced by timestamp variations. However, flexible segments slightly improve the model performance in the training interval.

5.5. Comparative evaluation

We now provide a comparative evaluation on the task of spatio-temporal motion completion w.r.t. a strong baseline based on a static parametric human body model and two recent state of the art motion priors [25, 11]. A common

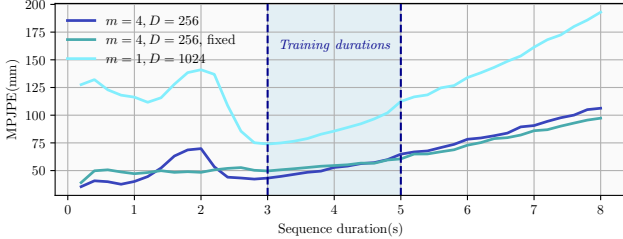


Figure 5. Value of using sequences of latent primitives and flexible segmentation. Lines show evolution of MPJPE (lower is better) of our method for different sequence duration (3 – 5s sequences used for training). Note that using latent sequences with flexible segmentation $m = 4, D = 256$ performs best in training interval and generalizes well.

task to evaluate that motion prior learnt a space of plausible human motion is to perform data completion from sparse 4D inputs by optimizing a latent representation. As most existing methods cannot process sequences of arbitrary duration, our evaluation focuses on how well the methods perform when the input signals are degraded spatially and/or temporally for a given duration. In this experiment, we consider unordered pointcloud inputs at increasingly sparse spatial and temporal resolutions.

Evaluation protocol To prevent any biases from having learned on AMASS, all comparative evaluations are performed on a *multi-view test set*, which was acquired by a multi-view camera system at 50 frames per second and for which no parametric representation is available. The dense per-frame pointclouds of roughly 10,000 points were obtained by a multi-view stereo method [10]. The data consists of 4 subjects (2 males, 2 females) performing different motions including boxing, kicking, sidestepping and various types of walking and running and one cartwheel sequence. In total, 170 motion sequences are used for testing. To evaluate the robustness of the methods w.r.t. degraded input signals, we downsample the pointcloud sequences spatially to 100 and 1000 points per frame, and temporally to 5 and 10 fps. For each experiment, we reconstruct coherent 4D sequences at 30 fps which is the proposed framerate in [25] and [11] and evaluate the error by the mean Chamfer distance over all frames of all test sequences. As the closest state of the art methods consider sequences of fixed duration (4s and 2s, respectively), we perform our evaluation of sequences of duration 4s, optimizing two latent vectors for [11]. All comparisons are based on code and pre-trained models provided with the respective publications.

Parametric baseline VPoser+SLERP The first baseline relies on the state of the art static pose prior VPoser [17]. In this approach, a latent pose representation is output per frame. As the global displacement is not encoded in VPoser, we additionally optimize per frame displacement. To increase the temporal resolution, we linearly interpolate between observed frames for displacement and using spherical linear interpolations (SLERP) for body pose rotations. We call this baseline VPoser+SLERP in the

following.

Motion prior with frequency guidance [25] We compare to two parametric human motion priors. The first one uses frequency guidance and was trained for motions of fixed duration (4s) at 30 fps. This prior does not encode global displacements, so we optimize them per input frame and interpolate linearly for the remaining frames.

Hierarchical motion prior [11] The second parametric prior uses a hierarchical approach to encode motions of fixed duration (2s) at 30 fps. As we consider sequences of 4s in our comparisons, we optimize for two segments by using two non-overlapping sliding windows of 2s.

Latent optimization Given a sequence of sparse pointclouds $\{P_i(\tau_i), \tau_i\}_{i=1}^n$ we optimize for a latent representation $\{\{z_j\}_{j=1}^m, \beta\}_{i=1}^n$ that best explains the observation. To initialize the latent representation, we randomly sample from the prior distribution in latent space for Xu *et al.* [25], Li *et al.* [11] and VPoser+SLERP.

For our method, a sequence characterizing a plausible motion cannot be initialized by independently sampling from the latent prior per latent primitive. Therefore, we use an initialization encoder which is trained as a mapping function from point cloud sequences to the sequential latent space. This initialization encoder is trained using pointcloud representations of the AMASS sequences. The initialization encoder has a similar architecture to the encoder network of our model shown in Figure 2 but the embedding layer has been replaced by a PointNet [19]. We use the output of this pointcloud encoder as initialization.

Then we solve for $\operatorname{argmin}_{\{z_j\}_{j=1}^m, \beta} (\mathcal{L}_{comp})$ with:

$$\mathcal{L}_{comp} = \mathcal{L}_{chamfer}(M(\mathcal{D}(\{z_j\}_{j=1}^m, \beta, \tau_i), P_i(\tau_i)) + \lambda_{prior} \mathcal{L}_{prior}(\{z_j\}_{j=1}^m), \quad (9)$$

where $\mathcal{L}_{chamfer}$ is the average Chamfer distance between input pointclouds and the vertices of the output meshes. \mathcal{L}_{prior} constrains the latent primitives to stay in the latent distribution. For Xu *et al.* [25], Li *et al.* [11] and VPoser+SLERP, it constrains the latent representation to stay close to the origin. In our method it constrains the latent representation to stay close to its initialization. We set $\lambda_{prior} = 0.01$ for all methods.

Results and discussion Table 1 reports the results obtained when completing from 5 and 10fps motion to a

# points per frame	100		1000		10000	
	5	10	5	10	5	10
VPoser+SLERP	27	25	24	20	24	20
Xu <i>et al.</i> [25]	28	26	26	24	26	24
Li <i>et al.</i> [11]	83	79	52	48	42	40
Ours	21	17	17	13	17	13

Table 1. Comparison to state of the art using average Chamfer distance (*mm*) (lower is better). Motion completion from different spatial (# points) and temporal (fps) resolutions.

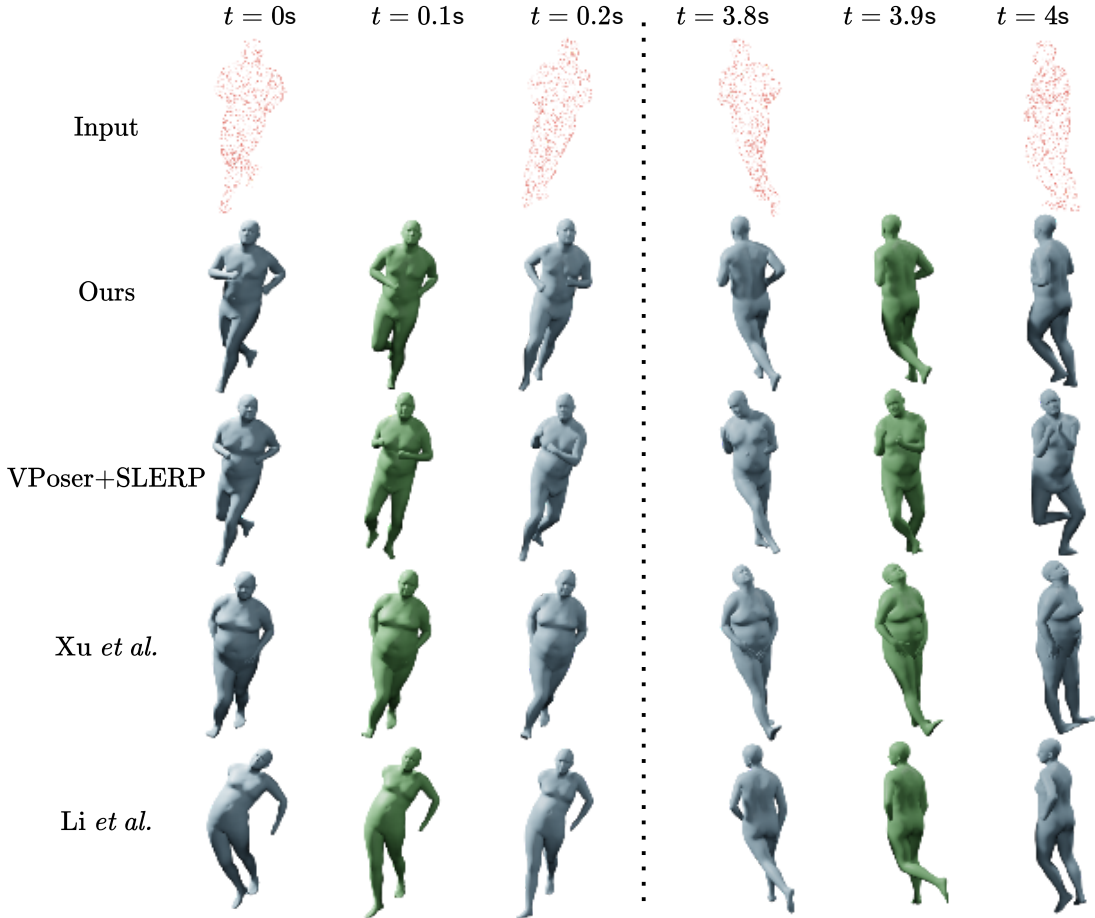


Figure 6. Comparison to state of the art on a challenging example. Completion results on a sequence of a person running in circle; we show frames close to the beginning and the end of the sequence. Our method estimates pose more precisely than other strategies. Blue meshes approximate input frames, green meshes are interpolated by the motion priors.

30fps motion, when considering input sequences increasingly sparsely sampled in space and time. Note that our method outperforms existing methods by a large margin, especially when considering very sparsely sampled input, and degrades gracefully for decreasing input resolutions.

Figures 6 and 7 show qualitative results on a sequence of a person running in a circle and a challenging cartwheel sequence when comparing the different methods. While our method generates a resulting motion very close to the sparsely sampled input point clouds, VPOSER+SLERP and Xu *et al.* fail to capture the global orientation of the motion, because they do not encode global displacement in their latent representation. While Li *et al.* finds the correct global orientation for the running sequence, the poses towards the beginning of the sequence are unrealistic. In contrast, our method is robust to global orientation and translation. As it was trained to predict the rigid transformation between segments ρ_i and encodes global displacement in latent space, it does not suffer from discontinuities at segment transitions. It also learned detailed motion features thanks to the small temporal windows covered per

segment, which is especially visible on the cartwheel motion where our model correctly captured arm positions while other methods could not generalize to this challenging input.

6. Conclusion and future works

This work presented a temporally implicit spatio-temporal representation of motion using a sequence of latent primitives. We showed that using latent primitives characterizing temporal segments of motion allows for a gain in precision which outweighs the gain of adding more latent dimensions. We also showed that our method learned plausible human motion and outperformed state-of-the-art motion priors on a completion task on unstructured point-clouds.

Future work will investigate sequential learning which takes into account dependencies between motion primitives. It could prove useful for motion synthesis and be promising to generate a coherent sequence of primitives from scratch.

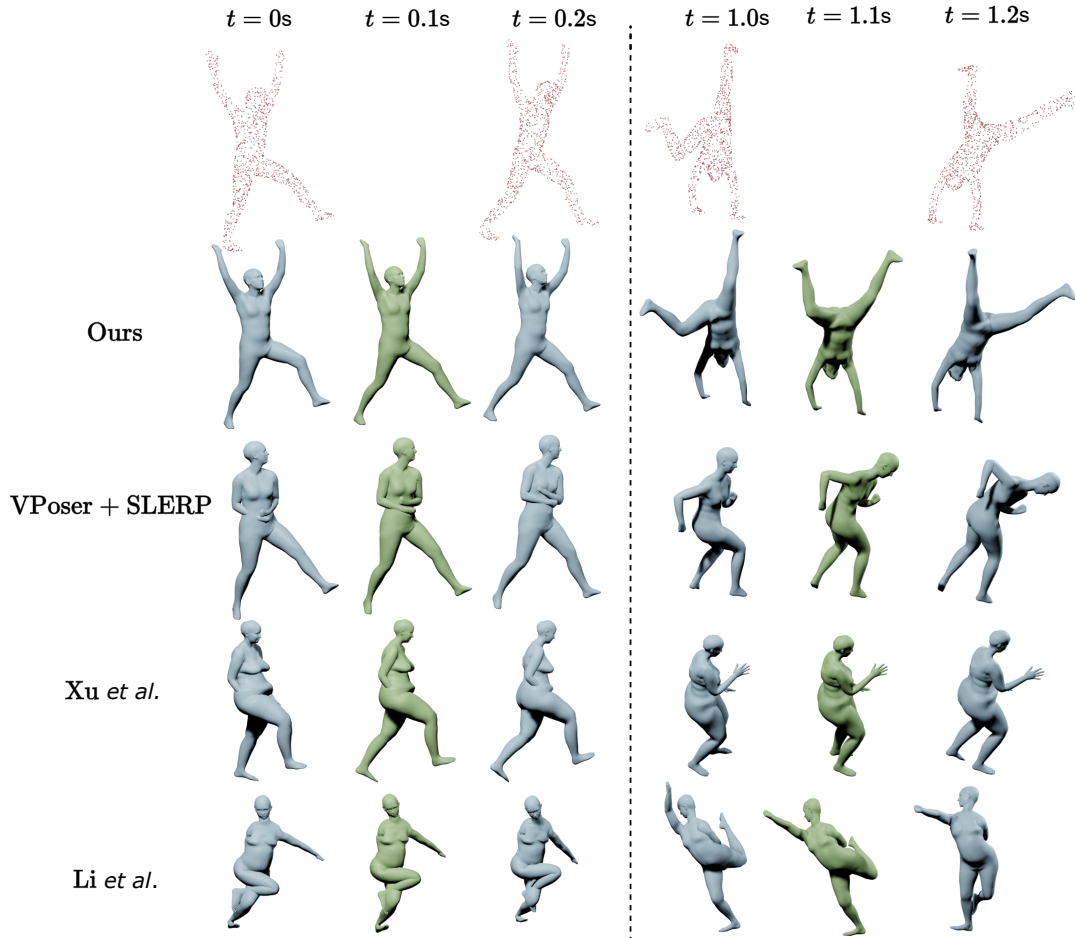


Figure 7. Comparison to state of the art on the most challenging example of the test set: a cartwheel; same color coding as Figure 6.

Potential negative societal impact

This work presents a novel representation of human motion sequences using a sequence of latent primitives, and an approach that allows to recover detailed human data from very sparse input. As detailed 3D human motion data is highly personal, such an approach could be used maliciously. Furthermore, our model might be used to re-target motions to different body shapes by changing β , which could be used to generate disinformation.

Acknowledgements

We thank Diego Thomas and Raphaël Dang Nhu for interesting discussions about the method design and paper redaction. We also thank Laurence Boissieux, Julien Panriot and the volunteer subjects for help in the acquisition and data processing of the multi-view test set. This work was supported by the National French Research Agency under grants 19-CE23-0013 (3DMOVE) and ANR-21-ESRE-0030 (CONTINUUM).

References

- [1] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yeaser Skeikh. Bilinear spatiotemporal basis models. *TOG*, 31:#17:1–12, 2012. 2
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. *Transactions on Graphics*, 24(3):408–416, 2005. 2
- [3] A. Boukhayma and E. Boyer. Surface motion capture animation synthesis. *TVCG*, 2018. 2
- [4] Saeed Ghorbani, Calden Wloka, Ali Etemad, Marcus A Brubaker, and Nikolaus F Troje. Probabilistic character motion synthesis using a hierarchical deep latent variable model. In *Computer Graphics Forum*, volume 39, pages 225–239, 2020. 2
- [5] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *TOG*, 40(4):94:1–16, 2021. 2

- [6] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ODE. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2
- [7] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. H4D: human 4d modeling by learning neural compositional representation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2
- [8] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, 2019. 2
- [9] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5
- [10] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *European Conference on Computer Vision*, pages 781–796, 2018. 6
- [11] Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. Task-generic hierarchical human motion prior using vaes. *Conference on 3D Vision*, 2021. 1, 2, 5, 6
- [12] Matthew Loper, Naureen Mahmood, and Michael Black. MoSh: motion and shape capture from sparse markers. *ToG*, 33, 2014. 2
- [13] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: a skinned multi-person linear model. *Transactions on Graphics*, 34(6):1–16, 2015. 3
- [14] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, 2019. 2, 3, 5
- [15] Mathieu Marsot, Stefanie Wuhrer, Jean-Sebastien Franco, and Stephane Durocher. A structured latent space for human body motion generation, 2021. 1, 2, 3
- [16] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, 2019. 2
- [17] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Conference on Computer Vision and Pattern Recognition*, 2019. 6
- [18] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer VAE. In *International Conference on Computer Vision*, 2021. 2
- [19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: deep learning on point sets for 3d classification and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2017. 6
- [20] J. Regateiro, A. Hilton, and M. Volino. Dynamic surface animation using generative networks. In *3DV*, 2019. 2
- [21] João Regateiro, Marco Volino, and Adrian Hilton. Deep4d: A compact generative representation for volumetric video. *Front. Virtual Reality*, 2:739010, 2021. 2
- [22] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision*, 2021. 2
- [23] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. CASPR: learning canonical spatiotemporal point cloud representations. *Advances in neural information processing systems*, 33:13688–13701, 2020. 2
- [24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3
- [25] Jiachen Xu, Min Wang, Jingyu Gong, Wentao Liu, Chen Qian, Yuan Xie, and Lizhuang Ma. Exploring versatile prior for human motion via motion frequency guidance. In *Conference on 3D Vision*, 2021. 1, 2, 5, 6
- [26] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *ICCV*, 2019. 2
- [27] Siwei Zhang, Yan Zhang, Federica Bogo, Pollefeys Marc, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *International Conference on Computer Vision (ICCV)*, Oct. 2021. 2
- [28] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2019. 4, 5