
MOTION-DVAE: UNSUPERVISED LEARNING FOR FAST HUMAN MOTION DENOISING

Guénolé Fiche¹, Simon Leglaive¹, Xavier Alameda-Pineda², and Renaud Séguier¹

¹CentraleSupélec, IETR UMR CNRS 6164, France

²Inria, Univ. Grenoble Alpes, CNRS, LJK, France

ABSTRACT

Pose and motion priors are crucial for recovering realistic and accurate human motion from noisy observations. Substantial progress has been made on pose and shape estimation from images, and recent works showed impressive results using priors to refine frame-wise predictions. However, a lot of motion priors only model transitions between consecutive poses and are used in time-consuming optimization procedures, which is problematic for many applications requiring real-time motion capture. We introduce Motion-DVAE, a motion prior to capture the short-term dependencies of human motion. As part of the dynamical variational autoencoder (DVAE) models family, Motion-DVAE combines the generative capability of VAE models and the temporal modeling of recurrent architectures. Together with Motion-DVAE, we introduce an unsupervised learned denoising method unifying regression- and optimization-based approaches in a single framework for real-time 3D human pose estimation. Experiments show that the proposed approach reaches competitive performance with state-of-the-art methods while being much faster. See the project page at <https://g-fiche.github.io/research-pages/motiondvae/>

1 Introduction

Human Motion Capture has become a key technology with many applications, such as character animation for the movie and video-game industries [1, 2, 3], performance optimization in sports [4, 5, 6], online shopping [7, 8], or even preservation of cultural heritage [9, 10]. With the rise of AR/VR technologies and the development of the metaverse [11, 12], we will need more accurate and realistic motion capture systems usable by everyone. Although giving good results, traditional marker-based systems can hardly be used on a large scale because they need time-consuming set-up and calibration. Recent advances have enabled motion capture from RGB images and videos [13, 14, 15]. This is a very promising solution to generalize the use of motion capture because it only requires a camera. However, current methods often produce unrealistic motions, especially in the case of occlusions or poor-quality images. The estimated noisy motion capture data needs post-processing before using it for real applications like motion recognition [16, 17] or scene understanding [18, 19].

There are numerous motion data post-processing methods, including temporal filtering [20], physical constraints [21], or statistical human motion priors [22, 23]. More recently, deep learning-based methods gained popularity with the pose priors like VPoser [24], or Pose-NDF [25], and motion priors like HuMoR [26] showing impressive results. Generative models are particularly attractive because they can be used in numerous applications, such as motion infilling or movement prediction. Despite impressive results, current pose and motion priors present several issues. First, many generative motion priors learn transitions between consecutive poses. These models are attractive because they have an excellent physical interpretation and generate realistic motions. However, they are subject to error accumulation in reconstruction, and motion dynamics are usually observable along multiple frames. Moreover, many pose and motion priors are used in optimization procedures, which are often prohibitively time-consuming. This is particularly problematic for multimedia applications like virtual and augmented environments, which require real-time motion capture of users [27]. Finally, very few methods model the noise in the observations.

To address these issues, we introduce Motion-DVAE, a generative motion prior using a dynamical variational autoencoder (DVAE) [28] to capture the short-term dynamics of human motion. While being a generative model like all VAEs [29, 30],

a DVAE introduces temporal relations among observed and latent variables using recurrent neural networks (RNNs). RNNs can accurately represent the dynamics of human motion, which cannot be summarized as a sequence of independent transitions.

Together with Motion-DVAE, we propose an unsupervised learned denoising method to refine 3D pose predictions in real time by using the proposed model in regression- or optimization-based procedures. For the learning step, the model parameters are estimated by maximizing the likelihood of the observations through an unsupervised optimization-based procedure. Contrary to prior works, we refine the Motion-DVAE encoder weights instead of the motion latent representation, enabling us to generalize to previously unseen data. Then, once learned on a database representative of the conditions in the test phase, denoising can be performed as a regression task (regression mode). We can also use the unsupervised optimization procedure for per-instance or per-dataset optimization (optimization mode).

In summary, our contributions are: (i) A generative motion prior, Motion-DVAE, representing short-term dependencies of human motion. (ii) A flexible unsupervised learning framework for real-time human motion denoising in regression- and/or optimization-based procedures. (iii) The proposed method is faster than state-of-the-art (SOTA) methods while showing competitive performance on pose estimation and motion denoising.

2 Related work

One of the main applications of the introduced method is to perform 3D human motion estimation from monocular videos. Therefore, we will review prior human pose and shape estimation works. We will then complete our study by reviewing works using human motion priors.

2.1 Human mesh estimation

Recent works have made significant progress in 3D human pose and motion estimation [14, 31, 32, 13, 26]. While earlier methods only focused on 3D joint locations [33, 34, 35, 36, 37], more and more works are model-based. This means that instead of working directly on joint coordinates, they estimate the parameters of a parametric model of the human body [38, 39, 24, 40, 41, 42]. Here we focus on model-based methods that estimate human pose and shape.

Optimization methods Optimization methods iteratively estimate the parameters of a body model given an image or a video. Most of the time, the optimization objective ensures that the projection of 3D predictions is consistent with a set of 2D cues. The first fully automatized method, SMPLify [31], fitted SMPL [39] to a set of 2D keypoints obtained with an off-the-shelf model [43]. SMPLify uses a set of pose and shape priors to guide the optimization so it converges toward realistic human bodies. Other works use different types of 2D information such as body silhouettes or part segmentation [44, 45, 46]. However, most contributions use 2D keypoints and focus on improving pose and shape priors [24, 47, 25], as well as the optimization process [48, 49, 46]. Various methods estimate motion by taking a sequence of observations as input. In that case, the optimization takes into account temporal motion priors [20, 50, 26], and some works jointly estimate moving camera position [51, 52]. Optimization methods usually give the most accurate results, but the inference is typically much slower than regression methods.

Regression methods In most cases, regression methods use a neural network to regress the parameters of a human body model, given an image as input. The canonical example of HMR [14] was the basis for several improvements [32, 13, 53, 54]. Recent advances took advantage of probabilistic modeling to propose multiple mesh predictions given an image [47, 55]. Improvements also include the development of video-based approaches. VIBE [15] and other methods [56, 57, 58] estimate motion directly from videos, while [59, 16] denoise predictions made by an off-the-shelf 3D prediction model with a single forward pass. The main problem for supervised regression methods is the lack of RGB data with 3D annotations, which are very hard to obtain. While [17, 60, 55] use synthetic data, another approach is to mix optimization and regression frameworks to learn in an unsupervised fashion. While HMR [14] uses a neural network to replace the optimization step, SPIN [32] initializes SMPLify [31] with a neural network and takes the output of the optimization as supervision.

In this work, we propose a unified framework that can be used in a regression- or optimization-based procedure. We will train Motion-DVAE on noisy data with an unsupervised optimization-based method and use it in regression or optimization modes on previously unseen data. This approach can be compared to the learning process of SPIN [32] as optimization is used to train a regression method. However, the proposed method is different since optimization is performed directly on the neural network’s weights instead of taking optimization results as pseudo-ground-truth for network training. Other works finetuned neural networks for per-instance optimization [61, 62], however to our knowledge, our approach is the first using unsupervised neural network training for per-dataset motion refinement.

2.2 Pose and motion priors

Pose and motion priors are crucial for accurate and realistic motion capture from noisy observations. The first proposed approaches relied on statistical models such as principal component analysis [22, 23] or Gaussian mixture models learned on pose data [31], and physical constraints [21]. With the rise of deep learning, new approaches emerged, using physics of motion [63, 64, 65], and adversarial training [14, 66, 67]. Recent pose [24] and motion [26, 16, 68, 69, 70, 71, 72, 3, 73] priors using VAEs [29, 30] showed very impressive results. In particular, ACTOR [16] learns an action-conditioned VAE-transformer prior, and HuMoR [26] uses a conditional VAE (CVAE) to learn transitions between consecutive poses. More recently, Pose-NDF [25] outperformed SOTA models [26, 24] for pose and motion denoising by learning a manifold of human poses.

In this work, we use a VAE-based model for motion-prior learning in line with recent works. RNNs used in DVAE enable us to model motion short-term dependencies efficiently.

3 Motion-DVAE

Notation. In all the following, we will use the notation $\mathcal{N}(\mu, \sigma)$ to denote a Gaussian distribution with mean μ and a diagonal covariance matrix with diagonal σ .

3.1 State representation

We want to represent the state of a moving person. A motion can be seen as a sequence of poses leading to translations and rotations in space. Thus, we take $r \in \mathbb{R}^3$ the global translation, $\phi \in \mathbb{R}^3$ the global orientation, and $\Theta \in \mathbb{R}^{21 \times 3}$ the pose in axis-angle representation. Those parameters can be fed to the SMPL model [39] to generate a human body mesh, given body shape parameters $\beta \in \mathbb{R}^{10}$. It will output a vertex mesh $V \in \mathbb{R}^{6890 \times 3}$ and a set of joints $J \in \mathbb{R}^{22 \times 3}$. In practice, we notice that giving the model parameters' velocities helps the DVAE to reconstruct human motions. Therefore, we also take \dot{r} , $\dot{\phi}$ and $\dot{\Theta}$ to represent the state. In the end, a moving person can be represented as a vector:

$$x = [r, \dot{r}, \phi, \dot{\phi}, \Theta, \dot{\Theta}] \in \mathbb{R}^{46 \times 3}. \quad (1)$$

We can obtain the meshes and joints associated with x using the function:

$$V, J = \mathcal{M}_\beta(x), \quad (2)$$

which corresponds to the SMPL forward pass computed from the appropriate entries of x and β .

3.2 Motion-DVAE model

We model human motion as a causal process: each state depends only on the past. We propose a DVAE [28] with a sequence of latent vectors $z_{1:T}$ to implicitly capture the temporal dependencies of human motion.

Generative model The joint distribution of the introduced Motion-DVAE is defined as:

$$p_\theta(x_{1:T}, z_{1:T}|x_0) = p_{\theta_z}(z_1|x_0)p_{\theta_x}(x_1|z_1, x_0) \times \prod_{t=2}^T p_{\theta_z}(z_t|z_{1:t-1}, x_0)p_{\theta_x}(x_t|z_{1:t}, x_0), \quad (3)$$

where θ_x and θ_z represent the decoder and the prior network parameters and $\theta = \theta_z \cup \theta_x$. From the DVAE perspective, x_0 is fixed and can be considered a parameter. Indeed we model the dynamics of motion, and x_0 is just the starting point of it. The generative model of the DVAE is set as follows:

$$p_{\theta_x}(x_t|z_{1:t}, x_0) = \mathcal{N}(x_t; \mu_{\theta_x}(z_{1:t}, x_0), Id), \quad (4)$$

where $\mu_{\theta_x}(z_{1:t}, x_0)$ is the output of the decoder network. We can interpret $z_{1:t}$ as a sequence of transitions allowing us to go from x_0 to x_t . z_t should then depend on x_0 and previous transitions. Hence, we assume $z_{1:0} = \emptyset$ and define the prior model as follows:

$$p_{\theta_z}(z_t|z_{1:t-1}, x_0) = \mathcal{N}(z_t; \mu_{\theta_z}(z_{1:t-1}, x_0), \sigma_{\theta_z}(z_{1:t-1}, x_0)). \quad (5)$$

The generative model will be fixed during denoising learning. Implementation details are given in Appendix A.1.

One can notice that given the latent sequence $z_{1:t}$, the DVAE does not use past states $x_{1:t-1}$ to predict x_t . Thus, contrary to most motion models like [26], we do not need to alternate between sampling x_t and z_t to sample a motion sequence. Furthermore, models that use previous predictions $x_{1:t-1}$ for predicting x_t are more likely to be subject to posterior collapse [74, 75, 76].

Inference model From the joint distribution Eq. (3), D-separation [28] yields the following approximate posterior distribution:

$$q_\phi(z_{1:T}|x_{0:T}) = q_\phi(z_0|x_{0:T}) \prod_t q_\phi(z_t|z_{1:t-1}, x_{t:T}, x_0). \quad (6)$$

We define the inference model as follows:

$$q_\phi(z_t|z_{1:t-1}, x_{t:T}, x_0) = \mathcal{N}(z_t; \mu_\phi(z_{1:t-1}, x_{t:T}, x_0), \sigma_\phi(z_{1:t-1}, x_{t:T}, x_0)), \quad (7)$$

where μ_ϕ and σ_ϕ are the output of the encoder network and $z_{1:0} = \emptyset$.

Implementation details of the inference model are given in Appendix A.2.

Loss functions The loss is computed using the evidence lower bound (ELBO) [28] derived from the inference and generative models. We also add a regularization term at the output of the SMPL model to enforce the final human mesh to be as close as possible to the original mesh, consisting of a squared reconstruction error on meshes and joints. More information about loss functions is available in Appendix A.3, and learning settings are detailed in Appendix A.4.

4 Unsupervised learned denoising

In this section, we propose to exploit Motion-DVAE to refine the pose estimate in the context of monocular videos or to recover motion from noisy 3D data. We introduce a robust noise model which can be tuned depending on the use case.

4.1 Problem definition and modeling

Suppose that we have 3D noisy observations in SMPL format:

$$y_t^{raw} = [r_t, \phi_t, \Theta_t, \beta_t]. \quad (8)$$

Those parameters' velocities can be obtained through finite differences in a deterministic way, leading to $y_t = [r_t, \dot{r}_t, \phi_t, \dot{\phi}_t, \Theta_t, \dot{\Theta}_t]$. We take as observations the SMPL-H [77] body joints $J_t \in \mathbb{R}^{22 \times 3}$ and a subset of mesh vertices $V_t \in \mathbb{R}^{43 \times 3}$ which can be seen as virtual motion capture markers [78]:

$$y_t^{3d} = [J_t, V_t] \in \mathbb{R}^{65 \times 3}. \quad (9)$$

From the 3D noisy observations, our goal is to infer the initial state x_0 , the sequence of latent variables $z_{1:T}$, and the body shape β that correspond to the ground-truth clean motion.

Obtaining 3D observations from an RGB video. As stated in Eq. (8), the proposed unsupervised learned denoising works on 3D global human motion noisy data. In order to obtain noisy motion capture data from RGB videos, we propose SPIN-t, a procedure built on SPIN's [32] frame-wise local pose estimations (in frame coordinates) to obtain global motion (in world coordinates). Details about SPIN-t are available in Appendix F.

Joint distribution. We define the generative model of the noisy observations with the following joint distribution:

$$p_\theta(y_{0:T}, v_{0:T}, z_{1:T}|\beta, x_0) = p(v_0)p(y_0|x_0, \beta, v_0) \times \prod_{t=1}^T p_\theta(z_t|z_{1:t-1}, x_0)p(v_t)p_\theta(y_t|z_{1:t}, \beta, v_t, x_0), \quad (10)$$

with:

$\triangleright p(y_0|x_0, \beta, v_0) = \mathcal{N}(y_0^{3d}; \mathcal{M}_\beta(x_0), v_0)$. This likelihood distribution is equivalent to considering that y_0 is equal to the output of the SMPL model applied on the initial state x_0 plus a heteroscedastic Gaussian noise with zero mean and variance coefficients v_0 . Note that the distribution is defined on the 3D predictions instead of SMPL parameters. This is more convenient since we do not need to define a probabilistic distribution over \mathcal{SO}^3 [55].

$\triangleright p_\theta(y_t|z_{1:t}, \beta, v_t, x_0) = \mathcal{N}(y_t^{3d}; \mathcal{M}_\beta(\mu_{\theta_x}(z_{1:t}, x_0)), v_t)$. This likelihood distribution is very similar to the previous one, except that we will use the Motion-DVAE decoder μ_{θ_x} to parameterize x_t by x_0 and a sequence of latent variables $z_{1:t}$.

$\triangleright p_\theta(z_t|z_{1:t-1}, x_0) = \mathcal{N}(z_t; \mu_{\theta_z}(z_{1:t-1}, x_0), \sigma_{\theta_z}(z_{1:t-1}, x_0))$. Here, we feed the initial state and past latent variables to the Motion-DVAE prior network $(\mu_{\theta_z}, \sigma_{\theta_z})$ to obtain the current latent variable z_t . We use the same prior distribution for $p_\theta(z_1|x_0)$ by assuming $z_{1:0} = \emptyset$.

$\triangleright p(v_t) = \prod_{j,d} p(v_{t,j,d})$ where $j \in \{1 : 22\}$ stands for the joint index, $d \in \{1 : 3\}$ is the spatial dimension (x , y , or z) and $p(v_{t,j,d}) = \mathcal{IG}(v_{t,j,d}; \frac{\lambda}{2}, \frac{\lambda}{2})$. This is a confidence model defined on 3D observations. \mathcal{IG} stands for the Inverse-Gamma distribution, the larger λ the higher confidence in observations. This variance model, along with the above-defined likelihood model $p_\theta(y_t|z_{1:t}, \beta, v_t, x_0)$ is equivalent to assuming that the 3D observations' positional error follows a Student-t distribution, which is more robust and adaptable than the Gaussian distribution [79].

Approximate Posterior We derive the approximate posterior of the latent variables given the noisy observations from the previous joint distribution and D-separation [80, 81] (see Appendix B):

$$q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T}|y_{0:T}) = q(\beta|y_{0:T})q_\omega(x_0|y_{0:T})q_\gamma(v_0|y_0, x_0) \times \prod_{t=1}^T \left[q_\gamma(v_t|z_{1:t}, y_t, x_0)q_\phi(z_t|z_{1:t-1}, y_{t:T}, x_0) \right], \quad (11)$$

where:

$\triangleright q(\beta|y_{0:T})$ is modeled as a Dirac delta function centered on the observations' average shape.

$\triangleright q_\phi(z_t|z_{1:t-1}, y_{t:T}, x_0)$ corresponds to the approximate posterior distribution of the DVAE described in Sec. 3.2. Hence, we will finetune the encoder weights during denoising learning.

$\triangleright q_\omega(x_0|y_{0:T})$ and $q_\gamma(v_t|z_{1:t}, y_t, x_0)$ are respectively Gaussian and Inverse Gamma distributions and will be implemented by two new neural networks (initial state and noise predictors) learned during the denoising learning step.

This inference model will be used to estimate the latent variables from the noisy observations, particularly the initial state x_0 and the latent motion $z_{1:T}$, which encode the ground truth motion $x_{1:T}$.

4.2 Unsupervised denoising learning

The aim of unsupervised denoising learning is to learn the parameters of the inference model defined in Eq. (11). The objective is to push the approximate posterior $q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T}|y_{0:T})$ towards the intractable exact posterior distribution $p_\theta(x_0, \beta, z_{1:T}, v_{0:T}|y_{0:T})$. We can show that this is equivalent to maximizing an ELBO [82], yielding the following objective function (see Appendix C):

$$\mathcal{L}(\phi, \gamma, \omega) = \mathcal{L}_{rec} + \mathcal{L}_{KL}^{DVAE} + \mathcal{L}_{KL}^{noise}, \quad (12)$$

where

$$\mathcal{L}_{rec} = -\frac{1}{2} \mathbb{E}_{q_{\phi, \gamma, \omega}} \left[\sum_{j,d} \frac{1}{v_{0,j,d}} (y_{0,j,d} - \mathcal{M}_\beta(x_0)_{j,d})^2 + \sum_{t,j,d} \frac{1}{v_{t,j,d}} (y_{t,j,d} - \mathcal{M}_\beta(\mu_{\theta_x}(x_0, z_{1:t}))_{j,d})^2 \right] \quad (13)$$

is a negative MSE weighted by the inverse of the estimated noise variance for each observation and

$$\mathcal{L}_{KL}^{DVAE} = -\sum_{t=1}^T \mathbb{E}_{q_\omega q_\phi} \left[D_{KL}(q_\phi(z_t|z_{1:t-1}, y_{t:T}, x_0) \| p_\theta(z_t|x_0, z_{1:t-1})) \right]; \quad (14)$$

$$\mathcal{L}_{KL}^{noise} = -\sum_{t=1}^T \mathbb{E}_{q_\omega q_\phi} \left[D_{KL}(q_\gamma(v_t|z_{1:t}, y_t, x_0) \| p(v_t)) \right] \quad (15)$$

push the approximate posteriors towards the priors.

To learn the inference model, we will jointly optimize ϕ , γ , and ω , which are the weights of the Motion-DVAE encoder initially learned on AMASS [83], noise predictor, and initial state predictor networks. No other parameter will be optimized, so there will not be any mandatory optimization process when performing denoising on new data. This enables dramatic speed-up of denoising new data. Note that Motion-DVAE prior and decoder networks are fixed during the denoising training process. This enables keeping the motion prior unchanged and focusing on filling the gap between noisy observations and clean motion capture data. Implementation details are given in Appendix E.

4.3 Denoising new data

For denoising new data, the final prediction is the posterior mean estimate of $x_{0:T}^{3d}$. First, we need to predict the initial state:

$$\hat{x}_0^{3d} = \mathbb{E}_{q_\omega(x_0|y_{0:T})} [\mathcal{M}_\beta(x_0)] \simeq \mathcal{M}_\beta(\mu_\omega(y_{0:T})). \quad (16)$$

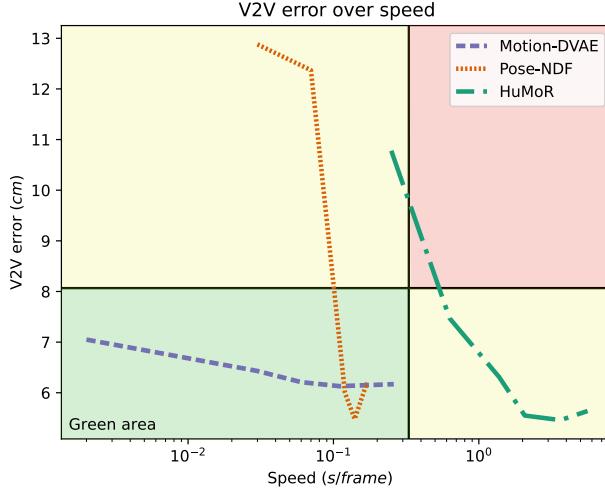


Figure 1: Evolution of the accuracy in V2V versus speed.

where μ_ω is the mean vector of $q_\omega(x_0|y_{0:T})$. To infer \hat{x}_0 , we have to feed the initial state predictor with observations and take the mean of the output distribution. For predicting $x_{1:T}$ we will use:

$$\hat{x}_{1:T} = \mathbb{E}_{q_{\phi,\gamma,\omega}} \left[\mathbb{E}_{p_\theta(x_{1:T}|x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T})} [x_{1:T}] \right]. \quad (17)$$

This leads to the following estimate:

$$\hat{x}_{t,j,d}^{3d} = \mathbb{E}_{q_{\phi,\gamma,\omega}} \left[\frac{v_{t,j,d} \mathcal{M}_\beta(\mu_{\theta_x}(x_0, z_{1:t-1})_{j,d}) + y_{t,j,d}^{3d}}{v_{t,j,d} + 1} \right]. \quad (18)$$

More detailed calculations are available in Appendix D. We assume that $\mathcal{M}_\beta(x_t)$ follows a Gaussian distribution with mean $\mathcal{M}(\mu_{\theta_x}(x_0, z_{1:t-1})_{j,d})$ and unit variance. This is an approximation because x_t being a Gaussian distribution does not mean that $\mathcal{M}_\beta(x_t)$ is also Gaussian and the variance of x_t and $\mathcal{M}_\beta(x_t)$ are not equal. However, this approximation is used in state-of-the-art methods [25] and works well in practice.

We can intuitively interpret the final clean motion estimate in Eq. (18): if the confidence in observations is very high, $v_{t,j,d}$ will be a small value, and the final predictions will be very close to $y_{t,j,d}^{3d}$. Conversely, if $v_{t,j,d}$ is large, the final prediction will rely more on the Motion-DVAE output. The process to estimate $\hat{x}_{1:T}$ compromises observations and the Motion-DVAE output weighted by the estimated variance. It needs recursive Monte Carlo sampling on x_0 and $z_{1:T}$ to calculate the expectation of the noise for the initial state and the motion. Despite this necessary sampling, denoising can be made in real time because each sample only consists of a forward pass in a neural network, which is almost instantaneous in our case.

5 Experimental results

This section evaluates the proposed unsupervised learned denoising framework with Motion-DVAE for human motion data denoising. Experiments on RGB videos and an ablation study are available as supplementary material (see Appendix H). We evaluate the proposed method for each experiment and compare it to the state-of-the-art in terms of accuracy and speed. All tests have been run on a single Nvidia GeForce GTX 1070.

For experiments on human motion data denoising we use AMASS [83]. The AMASS dataset [83] is a large database of human motion that unifies several motion capture databases by representing them in SMPL format. It contains more than 300 subjects and 10k motions, covering various shapes and poses. We sub-sample AMASS to 30Hz and follow the recommended training, validation, and testing splits.

5.1 Denoising noisy motion capture data

First, we evaluate the unsupervised learned denoising on noisy motion capture data. To generate noisy mocap data, we use the same procedure as Pose-NDF [25]: we add a Gaussian noise to the rotation of each joint in the axis-angle

Table 1: Motion denoising on noisy AMASS: We compare our method (bottom part) and SOTA (middle part) performance for 3 different noise values.

| | Speed ↓ | V2V ↓ | | | Acceleration | | |
|---------------------|--------------|-------------|-------------|-------------|--------------|-------------|-------------|
| Ground truth | - | - | | | 6.32 | | |
| Observations | - | 8.68 | 12.89 | 16.93 | 137.44 | 202.56 | 264.28 |
| HuMoR [26] | 5.60 | 5.61 | 6.31 | 7.97 | 4.36 | 5.55 | 8.76 |
| Pose-NDF [25] | 0.17 | 5.89 | 6.90 | 7.89 | 9.64 | 10.03 | 10.84 |
| Regression (ours) | 0.001 | 6.25 | 7.69 | 9.36 | 7.64 | 7.99 | 8.57 |
| Optimization (ours) | <u>0.06</u> | 5.02 | <u>6.74</u> | 8.72 | <u>8.09</u> | 9.98 | 9.49 |

representation. We test our method on sequences of 2 seconds. We set the noise standard deviation to 0.1, 0.15, and 0.2, corresponding to initial per-vertex errors of 8.68, 12.89, and 16.93 cm. The observations are the noisy vertices (virtual Mocap markers) and joints, and we initialize the denoising with the corresponding noisy SMPL parameters.

Starting from the proposed Motion-DVAE trained on AMASS, we use the unsupervised learning process described in Sec. 4.2 on the noisy AMASS training data. We then use our model in both regression and optimization modes. The inference model is fixed for regression mode, and predictions are made through a single forward pass in Motion-DVAE. In optimization mode, the inference model is optimized on test samples in an unsupervised fashion for a fixed number of iterations using Sec. 4.2. Note that all optimization methods, such as HuMoR [26] and Pose-NDF [25], perform optimization on the test set. The difference between the proposed approach and those methods is that we can use Motion-DVAE in regression mode and that optimization variables are neural networks parameter, enabling us to optimize the inference model for the whole test set at once independently of the number of sequences.

Instead of using Eq. (18) for denoising, we use the initial state predictor and Motion-DVAE output as the prediction. This saves us from learning to predict the noise on every 6890 vertices, which would be complicated in practice. Then, using the proposed Student-t noise model for training and optimization would not make as much sense since we do not use it for final predictions. Thus, we use very high values for noise parameter λ , which is equivalent to choosing a Gaussian prior distribution for the noise. Adding a Gaussian noise to the SMPL pose parameter does not give a Gaussian noise in 3D observations; however, this approximation was also made in the SOTA approach [25] and works well in practice. We compare our results in regression and optimization modes with public implementations of SOTA methods [25, 26] in Tab. 1, regarding speed (in $s/frame$), per-vertex-error (V2V in cm), and joints acceleration (in m/s^2). Joints acceleration measures the motion plausibility, the best joint acceleration is the closest to the ground truth value (6.32).

The proposed method is much faster than the SOTA methods. In the regression mode, the proposed method is 170 times faster than Pose-NDF and 5000 times faster than HuMoR. Regarding per-vertex error, the proposed approach in optimization mode outperforms the SOTA methods for an introduced noise of 8.68 cm. Our performance is similar to the SOTA models for an added noise of 12.89 cm in observations. However, the introduced method is less efficient than others for an added noise of 16.93 cm. This performance degradation for our approach when the added noise is large is unsurprising. Indeed, we never optimize the inputs of Motion-DVAE, which are always noisy observations. When the introduced noise is significant, correcting the error through a single forward pass becomes hard. In terms of joint acceleration, HuMoR produces the most realistic motions. However, it sometimes produces over-smoothed predictions, and the mean joint acceleration highly depends on the noise value. Pose-NDF produces unnatural motions according to the joint acceleration criteria, which is unsurprising since its only temporal model is a smoothing term over time. Our method produces reasonably smooth motions and is not very sensitive to the noise value. One can notice that optimization decreases the per-vertex error but also increases joint acceleration. This observation is expected: when Motion-DVAE is trained on an extensive training set and used in regression mode, it tends to produce smooth motions. In the optimization mode, the proposed model is adapted to specific noisy motion samples, which improves the prediction in terms of per-vertex error at the expense of smoothness.

Qualitative results are shown on Fig. 2. For Motion-DVAE, optimization was performed on the whole noisy AMASS test set, not only on the visualized sequence. Qualitative results on videos are available on the project website.

5.2 Methods accuracy versus speed

The previous experiment fixed the number of iterations for the proposed and state-of-the-art optimization-based methods. The proposed inference model was optimized for 50 iterations. We used the implementation default number of iterations for [26] and [25]. We are now interested in investigating the methods’ accuracy as a function of the speed. For this purpose, we vary the number of iterations. For the proposed method, we use the regression mode (0 optimization

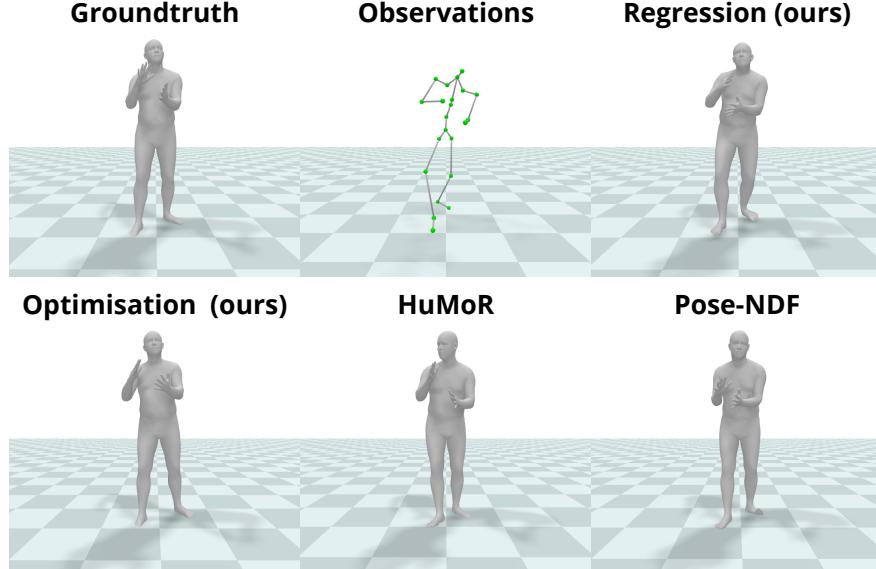


Figure 2: Qualitative results for motion denoising.

iterations) as well as the optimization mode for a number of steps between 20 and 250. The methodology for reducing [26] and [25] optimization processes as well as experiments on the output plausibility as a function of speed are exposed in the supplementary material (see Appendix H).

Experiments were performed on a randomly selected subset of 54 sequences from the test set of noisy AMASS, with a noise standard deviation of 0.15. We arbitrarily define acceptability thresholds for each metric, these could differ depending on the use case. For the speed, we consider that a method should not be more than 10 times slower than the frame rate. Beyond that, we consider that the method becomes difficult to use on a large scale. For accuracy, we expect the method to decrease the error of the noisy observations by at least 33 percent.

Results are shown in Fig. 1. HuMoR obtains good accuracy performance, but its slowness makes it unusable in practice. Pose-NDF with 5 steps satisfies all the criteria, however Motion-DVAE is the best choice to obtain good accuracy in a reasonable amount of time.

6 Conclusion

We introduced Motion-DVAE, a human motion prior based on the dynamical variational autoencoder framework for modeling short-term dependencies of human movement. Together with Motion-DVAE, we proposed an unsupervised learned denoising method allowing for robust noise modeling. This procedure can be used for motion denoising in regression- or optimization-based frameworks and adapts to new data in real time. Our method is more than 100 times faster than SOTA approaches for motion denoising while showing similar accuracy and plausibility. It is state-of-the-art for local motion estimation from videos, while being about 50 times faster than SOTA optimization methods.

Future work includes modeling environmental interactions, such as ground and object contact. This would probably help improve the global predictions, which is one of the weaknesses of Motion-DVAE. Another potential development would be improving noise modeling, for example, by learning visibility.

7 Acknowledgements

This study is part of the EUR DIGISPORT project supported by the ANR within the framework of the PIA France 2030 (ANR-18-EURE-0022). This work was performed using HPC resources from the “Mésocentre” computing center of CentraleSupélec, École Normale Supérieure Paris-Saclay, and Université Paris-Saclay supported by CNRS and Région Île-de-France.

References

- [1] Jenna Ng. Seeing movement: On motion capture animation and james cameron’s avatar. *Animation*, 7, 2012.
- [2] Sebastian Starke, Ian Mason, and Taku Komura. DeepPhase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics*, 41, 2022.
- [3] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Transactions on Graphics*, 39, 2020.
- [4] Moritz Einfalt, Dan Zecha, and Rainer Lienhart. Activity-conditioned continuous human pose estimation for performance analysis of athletes using the example of swimming. In *Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [5] William Robert Johnson, Jacqueline Alderson, David Lloyd, and Ajmal Mian. Predicting athlete ground reaction forces and moments from spatio-temporal driven cnn models. *IEEE Transactions on Biomedical Engineering*, 66, 2019.
- [6] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [7] Yuzhao Liu, Yuhang Liu, Shihui Xu, Kelvin Cheng, Soh Masuko, and Jiro Tanaka. Comparing vr- and ar-based try-on systems using personalized avatars. *Electronics*, 9, 2020.
- [8] Na Zheng, Xuemeng Song, Zhaozheng Chen, Linmei Hu, Da Cao, and Liqiang Nie. Virtually trying on new clothing with arbitrary poses. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [9] Ricardo M Rodriguez, Rafael Aguilar, Santiago Uceda, and Benjamín Castañeda. 3d pose estimation oriented to the initialization of an augmented reality system applied to cultural heritage. In *Digital Cultural Heritage: Final Conference of the Marie Skłodowska-Curie Initial Training Network for Digital Cultural Heritage, ITN-DCH 2017, Olimje, Slovenia, May 23–25, 2017, Revised Selected Papers*, 2018.
- [10] Efstrathios Stavrakis, Andreas Aristidou, Maria Savva, Stephania Loizidou Himona, and Yiorgos Chrysanthou. Digitization of cyprriot folk dances. In *Progress in Cultural Heritage Preservation: 4th International Conference, EuroMed 2012, Limassol, Cyprus, October 29–November 3, 2012. Proceedings* 4, 2012.
- [11] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai. Metaverse for social good: A university campus prototype. In *Proceedings of the 29th ACM international Conference on Multimedia*, 2021.
- [12] Lik-Hang Lee, Tristan Braud, Pengyuan Zhou, Lin Wang, Dianlei Xu, Zijun Lin, Abhishek Kumar, Carlos Bermejo, and Pan Hui. All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda. *arXiv preprint arXiv:2110.05352*, 2021.
- [13] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *European Conference on Computer Vision (ECCV)*, 2020.
- [14] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [15] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [16] Mathis Petrovich, Michael J. Black, and Gü̈l Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, 2021.
- [17] Gü̈l Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [18] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *International Conference on Computer Vision (ICCV)*, 2019.
- [19] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [20] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [21] Marcus Brubaker, David Fleet, and Aaron Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87, 2010.

- [22] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision (ECCV)*, 2000.
- [23] Dirk Ormoneit, Hedvig Sidenbladh, Michael Black, and Trevor Hastie. Learning and tracking cyclic human motion. In *Neural Information Processing Systems (NeurIPS)*, 2000.
- [24] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [25] Garvita Tiwari, Dimitrije Antic, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision (ECCV)*, 2022.
- [26] 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 (ICCV)*, 2021.
- [27] Chuanhang Yan, Yu Sun, Qian Bao, Jinhui Pang, Wu Liu, and Tao Mei. Woc: A handy webcam-based 3d online chatroom. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.
- [28] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends® in Machine Learning*, 15, 2021.
- [29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [30] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.
- [31] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, 2016.
- [32] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *International Conference on Computer Vision (ICCV)*, 2019.
- [33] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision (ICCV)*, 2017.
- [34] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *International Conference on Computer Vision (ICCV)*, 2017.
- [35] Xiaowei Zhou, Menglong Zhu, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [36] Yu Cheng, Bo Yang, Bo Wang, Yan Wending, and Robby Tan. Occlusion-aware networks for 3d human pose estimation in video. In *International Conference on Computer Vision (ICCV)*, 2019.
- [37] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [38] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *ACM Transactions on Graphics*, 24, 2005.
- [39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34, 2015.
- [40] Ahmed A A Osman, Timo Bolkart, and Michael J. Black. STAR: A sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, 2020.
- [41] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [42] Ahmed A. A. Osman, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. SUPR: A sparse unified part-based human representation. In *European Conference on Computer Vision (ECCV)*, 2022.
- [43] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [44] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [45] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [46] Taosha Fan, Kalyan Vasudev Alwala, Donglai Xiang, Weipeng Xu, Todd Murphy, and Mustafa Mukadam. Revitalizing optimization for 3d human pose and shape estimation: A sparse constrained formulation. In *International Conference on Computer Vision (ICCV)*, 2021.
- [47] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *International Conference on Computer Vision (ICCV)*, 2021.
- [48] Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. Learned vertex descent: A new direction for 3d human model fitting. In *European Conference on Computer Vision (ECCV)*, 2022.
- [49] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *European Conference on Computer Vision (ECCV)*, 2020.
- [50] Buzhen Huang, Yuan Shu, Jingyi Ju, and Yangang Wang. Occluded human body capture with self-supervised spatial-temporal motion prior. *arXiv preprint arXiv:2207.05375*, 2022.
- [51] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [52] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. *arXiv preprint arXiv:2302.12827*, 2023.
- [53] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regressor for 3D human body estimation. In *International Conference on Computer Vision (ICCV)*, 2021.
- [54] Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, and Zhenan Sun. Danet: Decompose-and-aggregate network for 3d human shape and pose estimation. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.
- [55] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Hierarchical kinematic probability distributions for 3d human shape and pose estimation from images in the wild. In *International Conference on Computer Vision (ICCV)*, 2021.
- [56] Zhengyi Luo, S. Alireza Golestaneh, and Kris M. Kitani. 3d human motion estimation via motion compression and refinement. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Asian Conference on Computer Vision (ACCV)*, 2021.
- [57] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [58] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.
- [59] Do-Yeop Kim and Ju-Yong Chang. Attention-based 3d human pose sequence refinement network. *Sensors*, 21, 2021.
- [60] Zhongang Cai, Mingyuan Zhang, Jiawei Ren, Chen Wei, Daxuan Ren, Jiatong Li, Zhengyu Lin, Haiyu Zhao, Shuai Yi, Lei Yang, et al. Playing for 3d human recovery. *arXiv preprint arXiv:2110.07588*, 2021.
- [61] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [62] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *International Conference on 3D Vision (3DV)*. IEEE, 2021.
- [63] Buzhen Huang, Liang Pan, Yuan Yang, Jingyi Ju, and Yangang Wang. Neural mocon: Neural motion control for physically plausible human motion capture. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [64] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [65] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics*, 39, 2020.
- [66] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

- [67] Kedi Lyu, Zhenguang Liu, Shuang Wu, Haipeng Chen, Xuhong Zhang, and Yuyu Yin. Learning human motion prediction via stochastic differential equations. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.
- [68] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Transactions on Graphics*, 39, 2020.
- [69] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joseph Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *British Machine Vision Conference (BMVC)*, 2017.
- [70] Yujun Cai, Yiwei Wang, Yiheng Zhu, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Chuanxia Zheng, Sijie Yan, Henghui Ding, Xiaohui Shen, Ding Liu, and Nadia Magnenat Thalmann. A unified 3d human motion synthesis model via conditional variational auto-encoder. In *International Conference on Computer Vision (ICCV)*, 2021.
- [71] Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. Task-generic hierarchical human motion prior using vaes. In *International Conference on 3D Vision (3DV)*, 2021.
- [72] Xiaoyu Bie, Wen Guo, Simon Leglaise, Lauren Girin, Francesc Moreno-Noguer, and Xavier Alameda-Pineda. Hit-dvae: Human motion generation via hierarchical transformer dynamical vae. *arXiv preprint arXiv:2204.01565*, 2022.
- [73] Chuan Guo, Xinxi Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [74] Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations (ICLR)*, 2019.
- [75] Bin Dai, Ziyu Wang, and David Wipf. The usual suspects? reassessing blame for vae posterior collapse. In *International Conference on Machine Learning (ICML)*, 2020.
- [76] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019.
- [77] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36, 2017.
- [78] Matthew Loper, Naureen Mahmood, and Michael J. Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 33, 2014.
- [79] Zhiqi Kang, Radu Horaud, and Mostafa Sadeghi. Robust face frontalization for visual speech recognition. In *International Conference on Computer Vision Workshops (ICCVW)*, 2021.
- [80] Dan Geiger, Thomas Verma, and Judea Pearl. Identifying independence in bayesian networks. *Networks*, 20, 1990.
- [81] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [82] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 1998.
- [83] 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 (ICCV)*, 2019.

Supplementary material

A Motion-DVAE training

A.1 Implementation of the generative model

We recall the generative model of Motion-DVAE:

$$p_{\theta}(x_{1:T}, z_{1:T}|x_0) = p_{\theta}(z_1|x_0)p_{\theta}(x_1|z_1, x_0)\prod_{t=2}^T p_{\theta}(z_t|z_{1:t-1}, x_0)p_{\theta}(x_t|z_{1:t}, x_0), \quad (1)$$

We implement $p_{\theta}(x_t|z_{1:t}, x_0) = \mathcal{N}(x_t; \mu_{\theta_x}(z_{1:t}, x_0), Id)$ with:

- $h_0^x = d_{ini}(x_0)$,
- $h_t^x = d_{h^x}(h_{t-1}^x, z_t)$,
- $\mu_{\theta_x}(h_t^x) = d_x(h_t^x)$.

For $p_{\theta}(z_t|z_{1:t-1}, x_0) = \mathcal{N}(z_t; \mu_{\theta_z}(z_{1:t-1}, x_0), \sigma_{\theta_z}(z_{1:t-1}, x_0))$, we take:

- $h_0^z = d_{ini}(x_0)$
- $h_t^z = d_{h^z}(h_{t-1}^z, z_t)$
- $[\mu_{\theta_z}(h_t^z), \sigma_{\theta_z}(h_t^z)] = d_z(h_t^z)$

d_{h^x} and d_{h^z} are recurrent neural networks and are both implemented with the same LSTM. d_{ini} , d_x , and d_z are MLPs with 2 hidden layers using group normalization with 16 groups and ReLU activations. The latent dimension is 48, while the LSTM hidden state dimension is 1024.

A.2 Implementation of the approximate posterior

The inference model is:

$$q_{\phi}(z_{1:T}|x_{0:T}) = q_{\phi}(z_0|x_{0:T})\prod_t q_{\phi}(z_t|z_{1:t-1}, x_{t:T}, x_0). \quad (2)$$

$q_{\phi}(z_t|z_{1:t-1}, x_{t:T}, x_0) = \mathcal{N}(z_t; \mu_{\phi}(z_{1:t-1}, x_{t:T}, x_0), \sigma_{\phi}(z_{1:t-1}, x_{t:T}, x_0))$ is implemented by:

- $h_t^{glob} = [h_{t-1}^{lat}, h_t^{data}]$,
- $h_0^{lat} = e_{ini}(x_0)$,
- $h_t^{lat} = e_h^{lat}(h_{t-1}^{lat}, z_t)$,
- $h_0^{data} = 0$,
- $h_t^{data} = e_h^{data}(h_{t+1}^{data}, x_t)$,
- $[\mu_{\phi}(h_t^{glob}), \sigma_{\phi}(h_t^{glob})] = e_{glob}(h_t^{glob})$.

e_h^{lat} and e_h^{data} are LSTM neural networks, while e_{ini} and e_{glob} are MLPs. Implementations of LSTM and MLP are similar to the generative model.

A.3 Loss function

From Eq. (3) and Eq. (6), following [84], the ELBO of Motion-DVAE is:

$$\mathcal{L}(\theta, \phi; x_{0:T}) = \sum_{t=1}^T \mathbb{E}_{q_{\phi}}[\log p_{\theta}(x_t|z_{1:t}, x_0)] - \sum_{t=1}^T \mathbb{E}_{q_{\phi}}[D_{KL}(q_{\phi}(z_t|z_{1:t-1}, x_{t:T}, x_0)||p_{\theta}(z_t|z_{1:t-1}, x_0))]. \quad (3)$$

We define the loss function of Motion-DVAE:

$$\mathcal{L}^{train}(\theta, \phi; x_{0:T}) = \mathcal{L}_{rec}(\theta, \phi; x_{0:T}) + \mathcal{L}_{KL}(\theta, \phi; x_{0:T}) + \mathcal{L}_{reg}(\theta, \phi; x_{0:T}). \quad (4)$$

The first term can be developed as:

$$\begin{aligned} \mathcal{L}_{rec}(\theta, \phi; x_{0:T}) &= -\sum_{t=1}^T \mathbb{E}_{q_\phi} [\log p_\theta(x_t | x_0, z_{1:t})] \\ &= -\sum_{t=1}^T \mathbb{E}_{q_\phi} [\log \mathcal{N}(x_t; \mu_{\theta_x}(z_{1:t}, x_0), Id)] \\ &= \sum_{t=1}^T \mathbb{E}_{q_\phi} [\|\mu_{\theta_x}(z_{1:t}, x_0) - x_t\|_2^2]. \end{aligned}$$

This is a reconstruction term between the input and the output of Motion-DVAE.

The second term is a Kullback-Leibler divergence pushing the posterior distribution towards the prior:

$$\mathcal{L}_{KL}(\theta, \phi; x_{0:T}) = \sum_{t=1}^T \mathbb{E}_{q_\phi} [D_{KL}(q_\phi(z_t | z_{1:t-1}, x_{t:T}, x_0) || p_\theta(z_t | z_{1:t-1}, x_0))].$$

We also add a regularization term using the SMPL model to enforce the final human mesh to be as close as possible to the original mesh. It consists of a squared reconstruction error on meshes and joints:

$$\mathcal{L}_{reg}(\theta, \phi; x_{0:T}) = \sum_{t=1}^T \mathbb{E}_{q_\phi} [\|\mathcal{M}_\beta(\mu_{\theta_x}(z_{1:t}, x_0)) - \mathcal{M}_\beta(x_t)\|_2^2].$$

A.4 Learning settings

We train Motion-DVAE with sequences of 30 frames from the AMASS [85] dataset, previously downsampled to 30Hz. Then, learning sequences last 1 second. We choose this duration because we argue that even if human motion can last more than 1 second, direct dependencies between poses rarely exceeds 1 second. To ease learning, following [86], we align the first frame of each sequence in the canonical coordinate frame, meaning that translation r_0 and the first two components of root-orient Φ_0 are 0. This enables focusing on learning spatial-temporal dependencies independently from the starting point of the motion, which makes the motion prior more general.

We use batches of 64 sequences and train Motion-DVAE for 200 epochs. Similar to HuMoR [86], we use Adamax [87] with the same settings and learning rate decays. We also use KL-annealing [88] during the first 50 epochs. However, since our model does not use past predictions for current state prediction, we do not need to perform scheduled sampling.

B Unsupervised learned denoising posterior distribution

Let's recall the joint distribution in the context of motion denoising:

$$p_\theta(y_{0:T}, v_{0:T}, z_{1:T} | \beta, x_0) = p(v_0)p(y_0 | x_0, \beta, v_0) \prod_{t=1}^T p_\theta(z_t | z_{1:t-1}, x_0)p(v_t)p_\theta(y_t | z_{1:t}, \beta, v_t, x_0). \quad (5)$$

We want to express the posterior distribution $p_\theta(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})$:

$$\begin{aligned} p_\theta(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) &= p(\beta | v_{0:T}, z_{1:T}, x_0, y_{0:T})p_\theta(v_{0:T}, z_{1:T}, x_0 | y_{0:T}) \\ &= p(\beta | v_{0:T}, z_{1:T}, x_0, y_{0:T})p(v_{0:T} | z_{1:T}, x_0, y_{0:T})p_\theta(z_{1:T} | x_0, y_{0:T})p(x_0 | y_{0:T}) \\ &= p(\beta | v_{0:T}, z_{1:T}, x_0, y_{0:T}) \prod_{t=1}^T [p(v_t | z_{1:t}, y_t, x_0)p_\theta(z_t | z_{1:t-1}, y_{t:T}, x_0)]p(v_0 | y_0, x_0)p(x_0 | y_{0:T}) \\ &\simeq p(\beta | y_{0:T}) \prod_{t=1}^T [p(v_t | z_{1:t}, y_t, x_0)p_\theta(z_t | z_{1:t-1}, y_{t:T}, x_0)]p(v_0 | y_0, x_0)p(x_0 | y_{0:T}). \end{aligned}$$

As can be seen in the above calculations, we approximate the posterior of β . We choose to ignore the noise and the latent motion to simplify the model. In practice, β is computed as the average of the observed sequence of body shapes. As a reminder, we define the following approximate posterior:

$$q(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) = q(\beta | y_{0:T}) \prod_{t=1}^T \left[q_\gamma(v_t | z_{1:t}, y_t, x_0) q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \right] q_\gamma(v_0 | y_0, x_0) q_\omega(x_0 | y_{0:T}). \quad (6)$$

C ELBO derivation and loss functions

For finetuning Motion-DVAE, we aim to minimize the Kullback-Leibler divergence:

$$\min_{\phi, \gamma, \omega} D_{KL}(q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) || p_\theta(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})). \quad (7)$$

We will do it by maximizing the ELBO:

$$\mathcal{L}(\phi, \gamma, \omega) = \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log p_\theta(y_{0:T}, v_{0:T}, z_{1:T}, x_0, \beta) - \log q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})]. \quad (8)$$

C.1 ELBO decomposition

One can notice that the first term of the ELBO involves the joint distribution defined in Eq. (5). Taking the logarithm and expectation of this joint distribution, we obtain:

$$\mathbb{E}_{q_{\phi, \gamma, \omega}} [\log p(v_0) + \log p(y_0 | x_0, \beta, v_0)] + \sum_{t=1}^T \mathbb{E}_{q_{\phi, \gamma, \omega}} [\log p_\theta(z_t | x_0, z_{1:t-1}) + \log p(v_t) + \log p_\theta(y_t | x_0, z_{1:t}, \beta, v_t)].$$

Similarly, the second term involves:

$$q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) = q(\beta | y_{0:T}) \prod_{t=1}^T \left[q_\gamma(v_t | z_{1:t}, y_t, x_0) q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \right] q_\gamma(v_0 | y_0, x_0) q_\omega(x_0 | y_{0:T}),$$

which becomes after taking the logarithm and expectation:

$$\begin{aligned} & \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q(\beta | y_{0:T}) + \log q_\gamma(v_0 | y_0, x_0) + \log q_\omega(x_0 | y_{0:T})] \\ & + \sum_{t=1}^T \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q_\gamma(v_t | z_{1:t}, y_t, x_0) + \log q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0)]. \end{aligned}$$

The decomposed ELBO is then:

$$\begin{aligned} \mathcal{L}(\phi, \gamma, \omega) = & \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} \left[\log p(y_0 | x_0, \beta, v_0) + \sum_{t=1}^T \log p_\theta(y_t | z_{1:t}, \beta, v_t, x_0) \right] \\ & + \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [-\log q_\omega(x_0 | y_{0:T})] \\ & + \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} \left[\sum_{t=1}^T \log p_\theta(z_t | z_{1:t-1}, x_0) - \sum_{t=1}^T \log q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \right] \\ & + \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} \left[\sum_{t=0}^T \log p(v_t) - \sum_{t=1}^T \log q_\gamma(v_t | z_{1:t}, y_t, x_0) \right] \\ & + \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [-\log q(\beta | y_{0:T})], \end{aligned}$$

which can be rewritten as:

$$\begin{aligned}
\mathcal{L}(\phi, \gamma, \omega) &= \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} \left[\log p(y_0 | x_0, \beta, v_0) + \sum_{t=1}^T \log p_\theta(y_t | z_{1:t}, \beta, v_t, x_0) \right] \\
&\quad - \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q_\omega(x_0 | y_{0:T})] \\
&\quad - \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, v_{0:T} | z_{1:T}, y_{0:T})} \left[D_{KL} \left(\prod_{t=1}^T q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \| \prod_{t=1}^T p_\theta(z_t | z_{1:t-1}, x_0) \right) \right] \\
&\quad - \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T} | v_{0:T}, y_{0:T})} \left[D_{KL} \left(\prod_{t=0}^T q_\gamma(v_t | z_{1:t}, y_t, x_0) \| \prod_{t=0}^T p(v_t) \right) \right] \\
&\quad - \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q(\beta | y_{0:T})].
\end{aligned}$$

C.2 ELBO terms calculation

C.2.1 Data commitment term

The first term is a data commitment term. We have:

$$\begin{aligned}
&\mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} \left[\log p(y_0 | x_0, \beta, v_0) + \sum_{t=1}^T \log p_\theta(y_t | x_0, z_{1:t}, \beta, v_t) \right] \\
&= \mathbb{E}_{q_{\phi, \gamma, \omega}} \left[\log \mathcal{N}(y_0; \mathcal{M}_\beta(x_0), v_0) + \sum_{t=1}^T \log \mathcal{N}(y_t; \mathcal{M}_\beta(\mu_{\theta_x}(x_0, z_{1:t})), v_t) \right] \\
&= \frac{1}{2} \mathbb{E}_{q_{\phi, \gamma, \omega}} \left[\sum_{j,d} \frac{1}{v_{0,j,d}} (y_{0,j,d} - \mathcal{M}_\beta(x_0)_{j,d})^2 + \sum_{t,j,d} \frac{1}{v_{t,j,d}} (y_{t,j,d} - \mathcal{M}_\beta(\mu_{\theta_x}(x_0, z_{1:t}))_{j,d})^2 \right]
\end{aligned}$$

As expected, those are Mean Squared Errors weighted by the inverse of the variance.

C.2.2 Initial state predictor term

The second term is linked to the initial state predictor:

$$\mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q_\omega(x_0 | y_{0:T})] = \mathbb{E}_{q_\omega(x_0 | y_{0:T})} [\log q_\omega(x_0 | y_{0:T})].$$

This term corresponds to the negative entropy of $q_\omega(x_0 | y_{0:T})$. We chose not to use it during learning because it would increase the initial state predicted variance, which is not a desired behavior.

C.2.3 Motion prior term

The third term of the ELBO uses Motion-DVAE to implement a motion prior:

$$\begin{aligned}
&\mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, v_{0:T} | z_{1:T}, y_{0:T})} \left[D_{KL} \left(\prod_{t=1}^T q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \| \prod_{t=1}^T p_\theta(z_t | z_{1:t-1}, x_0) \right) \right] \\
&= \sum_{t=1}^T \mathbb{E}_{q_\omega q_\phi} [D_{KL}(q_\phi(z_t | z_{1:t-1}, y_{t:T}, x_0) \| p_\theta(z_t | x_0, z_{1:t-1}))] \\
&= \sum_{t=1}^T \mathbb{E}_{q_\omega q_\phi} [D_{KL}(\mathcal{N}(z_t; \mu_\phi(z_{1:t-1}, x_{t:T}, x_0), \sigma_\phi(z_{1:t-1}, x_{t:T}, x_0)) \| \mathcal{N}(z_t; \mu_{\theta_z}(z_{1:t-1}, x_0), \sigma_{\theta_z}(z_{1:t-1}, x_0))] \\
&= -\frac{1}{2} \sum_{t=1}^T \mathbb{E}_{q_\omega q_\phi} \left[\log \frac{\sigma_{\mu_{\theta_z}}}{\sigma_\phi} - 1 + \frac{\sigma_\phi}{\sigma_{\mu_{\theta_z}}} + \frac{(\mu_{\mu_{\theta_z}} - \mu_\phi)^2}{\sigma_{\mu_{\theta_z}}} \right].
\end{aligned}$$

C.2.4 Noise prior term

The fourth term of the ELBO is a noise prior term:

$$\begin{aligned}
& \mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T} | v_{0:T}, y_{0:T})} \left[D_{KL} \left(\prod_{t=0}^T q_{\gamma}(v_t | z_{1:t}, y_t, x_0) \| \prod_{t=0}^T p(v_t) \right) \right] \\
&= \sum_{t=1}^T \mathbb{E}_{q_{\omega} q_{\phi}} [D_{KL}(q_{\gamma}(v_t | z_{1:t}, y_t, x_0) \| p(v_t))] \\
&= \sum_{t,j,d} \mathbb{E}_{q_{\omega} q_{\phi}} \left[D_{KL} \left(\mathcal{IG}(v_{t,j,d}, \alpha_{\gamma}(z_{1:t}, y_{t,j,d}, x_0), \beta_{\gamma}(z_{1:t}, y_{t,j,d}, x_0)) \| \mathcal{IG} \left(v_{t,j,d}, \frac{\lambda}{2}, \frac{\lambda}{2} \right) \right) \right] \\
&= \sum_{t,j,d} \mathbb{E}_{q_{\omega} q_{\phi}} \left[\left(\alpha_{\gamma} - \frac{\lambda}{2} \right) \psi(\alpha_{\gamma}) - \log \Gamma(\alpha_{\gamma}) + \log \Gamma \left(\frac{\lambda}{2} \right) + \frac{\lambda}{2} \left(\log \beta_{\gamma} - \log \frac{\lambda}{2} \right) + \alpha_{\gamma} \frac{\frac{\lambda}{2} - \beta_{\gamma}}{\beta_{\gamma}} \right],
\end{aligned}$$

where Γ and ψ are the Gamma and Digamma functions.

C.2.5 Body shape term

The last term is about body shape:

$$\mathbb{E}_{q_{\phi, \gamma, \omega}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [\log q(\beta | y_{0:T})] = \mathbb{E} [\delta(\beta - \bar{\beta}_{SPIN}(y_{0:T}))]$$

Since that term does not depend on any learned parameter, we ignore it during training.

D Final predictions from observations

D.1 Initial state

We start by predicting the initial state:

$$\hat{x}_0^{3d} = \mathbb{E}_{q_{\omega}(x_0 | y_{0:T})} [\mathcal{M}_{\beta}(x_0)] \simeq \mathcal{M}_{\beta}(\mu_{\omega}(y_{0:T})), \quad (9)$$

where μ_{ω} is the mean vector of $q_{\omega}(x_0 | y_{0:T})$.

D.2 Motion prediction

Then we need to compute $\hat{x}_{1:T} = \mathbb{E}_{p_{\theta}(x_{1:T} | y_{0:T})} [x_{1:T}]$ where

$$\begin{aligned}
p(x_{1:T} | y_{0:T}) &= \int p_{\theta}(x_{1:T}, x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) d_{x_0} d_{\beta} d_{z_{1:t}} d_{v_{0:T}} \\
&= \int p_{\theta}(x_{1:T} | x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T}) p_{\theta}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T}) d_{x_0} d_{\beta} d_{z_{1:t}} d_{v_{0:T}} \\
&= \mathbb{E}_{p_{\theta}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})} [p_{\theta}(x_{1:T} | x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T})].
\end{aligned}$$

Approximating the posterior $p_{\theta}(x_0, \beta, z_{1:T}, v_{0:T} | y_{0:T})$ with $q_{\phi, \gamma, \omega}$ we obtain:

$$\hat{x}_{1:T} = \mathbb{E}_{q_{\phi, \gamma, \omega}} \left[\mathbb{E}_{p_{\theta}(x_{1:T} | x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T})} [x_{1:T}] \right]. \quad (10)$$

We need to calculate $p_{\theta}(x_{1:T} | x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T})$:

$$\begin{aligned}
\log p_{\theta}(x_{1:T} | x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T}) &\stackrel{c}{=} \log p_{\theta}(x_{1:T}, x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T}) \\
&\stackrel{c}{=} \sum_t \log p(y_t | x_t, v_t) p_{\theta}(x_t | z_{1:T}, x_0) \\
&= \sum_t \log \mathcal{N}(y_t; x_t, \text{diag}(v_t)) \mathcal{N}(x_t; \mu_{\theta_x}, \sigma_{\theta_x}),
\end{aligned}$$

where $\stackrel{c}{=}$ denotes equality up to an additive constant that does not depend on $x_{1:T}$. We can further develop the last line and identify:

$$\log p_\theta(x_{1:T}|x_0, \beta, z_{1:T}, v_{0:T}, y_{0:T}) = \sum_{t,j,d} \log \mathcal{N}\left(x_{t,j,d}; \frac{v_{t,j,d}(\mu_{\theta_x})_{j,d} + (\sigma_{\theta_x})_{j,d}y_{t,j,d}}{v_{t,j,d} + (\sigma_{\theta_x})_{j,d}}, \frac{v_{t,j,d}(\sigma_{\theta_x})_{j,d}}{v_{t,j,d} + (\sigma_{\theta_x})_{j,d}}\right).$$

This finally leads to:

$$\hat{x}_{t,j,d}^{3d} = \mathbb{E}_{q_{\phi,\gamma,\omega}} \left[\frac{v_{t,j,d} \mathcal{M}_\beta(\mu_{\theta_x}(x_0, z_{1:t-1})_{j,d}) + y_{t,j,d}^{3d}}{v_{t,j,d} + 1} \right]. \quad (11)$$

E Unsupervised denoising learning

E.1 Neural networks implementation

During the unsupervised denoising training, we introduce 2 neural networks: an initial state predictor and a noise predictor. Those 2 models were not necessary for training Motion-DVAE on clean motion capture data since x_0 was known and there was no noise in the observations.

As described in the main paper, the initial state predictor implements $q_\omega(x_0|y_{0:T}) = \mathcal{N}(x_0; \mu_\omega(y_{0:T}), \sigma_\omega(y_{0:T}))$. It is implemented by:

- $h_0 = 0$
- $h_t = r_h(h_{t+1}, y_t)$
- $\hat{x}_0 = e_{ini}(h_T)$

r_h is an anticausal LSTM neural network. We want it to take the observations $y_{0:T}$ backward in time because x_0 should depend more on y_0 than on y_T . e_{ini} is an MLP.

For the noise predictor, we implement $q_\gamma(v_t|z_{1:t}, y_t, x_0) = \prod_{j,d} \mathcal{IG}(v_{t,j,d}; \alpha_\gamma(z_{1:t}, y_t, x_0), \beta_\gamma(z_{1:t}, y_t, x_0))$ as follows:

- $h_0^z = d_{ini}(x_0)$
- $h_t^z = d_{h^z}(h_{t-1}^z, z_t)$
- $[\alpha_\gamma, \beta_\gamma] = e_b(h_t, y_t)$

d_{ini} and d_{h^z} are the MLP and the LSTM neural networks previously defined for Motion-DVAE. e_b is an MLP only used by the noise predictor.

E.2 Learning settings

Unsupervised denoising training for regression mode is performed on training data for 500 epochs, with early stopping when the validation loss does not improve for 10 epochs. Note that for noisy AMASS [85] data, the training converges in about 50 epochs only, probably due to a large amount of training data. In optimization mode, we fix the number of iterations (200 iterations on i3DB [89], and 50 for AMASS).

During unsupervised denoising learning, the decoder and prior networks are fixed, preserving the motion prior. We optimize the weights of the encoder, initial state predictor, and noise predictor networks. As for training the original Motion-DVAE, we use KL-annealing [88].

F SPIN-t

SPIN-t is a procedure built on SPIN's [90] frame-wise local pose estimations (in frame coordinates) to obtain global motion (in world coordinates). SPIN provides SMPL pose and shape parameters Θ and β , as well as orientation ϕ relative to the camera. To obtain motion in global coordinates, we need the global translation r . For smooth and realistic 3D trajectories, we decide to optimize both global translation and rotation. We define v as the output of the SMPL model forward passes with optimization variables r and ϕ and SPIN [90] predictions β and Θ . During the optimization, v it only depends on the global translation and rotation. We use L-BFGS [91] to solve the following optimization problem:

$$\min_{r,\phi} \lambda_{data} \xi_{data}(r, \phi) + \lambda_{smooth} \xi_{smooth}(r, \phi), \quad (12)$$

where $\xi_{data}(r, \phi) = \sum_t \sum_j \sigma_{t,j} \rho(\Pi(v_{t,j}) - y_{t,j})$, with t the temporal index, j the observed joint, $y_{t,j}$ and $\sigma_{t,j}$ the Openpose [92] 2D detection and its associated confidence, ρ the robust Geman McClure function [93], and Π the pinhole projection; and $\xi_{smooth} = \|v_{1:T} - v_{0:T-1}\|_2^2$. ξ_{data} ensures that the projection of the predicted 3D joints corresponds to the 2D detections, while ξ_{smooth} smooths the motion over time. λ_{data} and λ_{smooth} are two hyperparameters.

G Ablation of SOTA optimization processes

The optimization process of HuMoR can be divided into 2 parts. A first part called VPoser-t minimizes the projection error of 3D predictions relative to 2D keypoints [92], with a pose prior [94] and regularization terms. The second part of optimization uses a CVAE motion prior to predict more plausible motions. We first decrease the number of iterations of the last part, which is the most time-consuming and the less important for the final results. We then decreased the number of iterations for VPoser-t, leading to significant degradation in performance.

Pose-NDF uses 5 steps of 50 iterations. At each iteration, the observation reconstruction weight becomes less and less important compared to the pose prior and smoothing term. Then, changing the number of iterations per step would change the balance of the loss functions, and we thus reduce the number of steps instead.

H Additional experiments

H.1 Plausibility versus speed

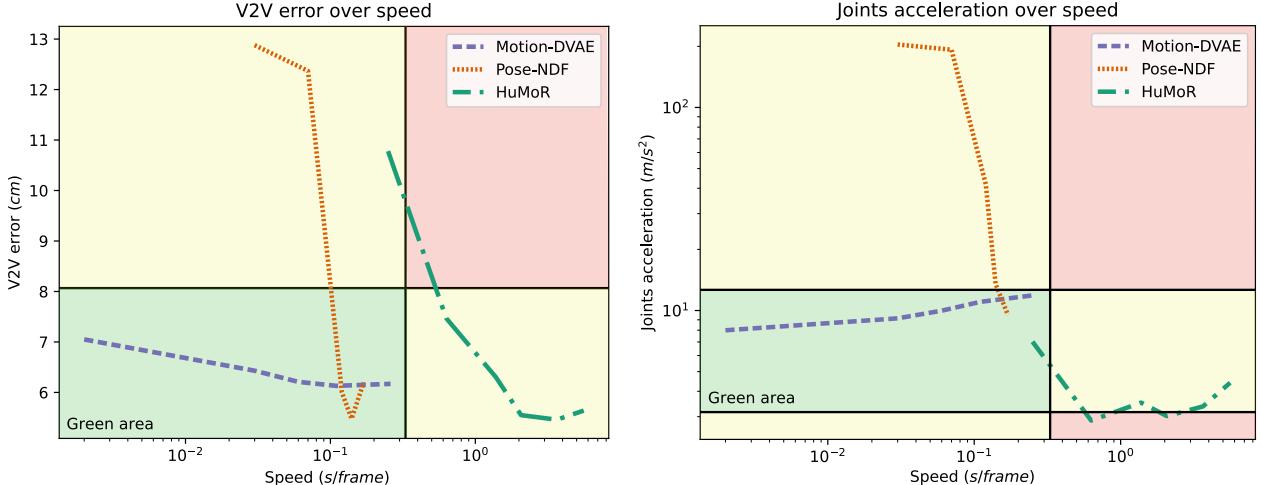


Figure 1: Evolution of the performance versus speed. Vertex-to-vertex (V2V) error is an accuracy metric, while joints acceleration measures plausibility.

This experiment is similar to the "Accuracy versus Speed" experiment in the main paper. We consider that joint acceleration should be between half and double of the ground-truth joint acceleration. Results are shown in Fig. 1.

HuMoR obtains satisfying plausibility performance, but it is very slow. Pose-NDF produces motions that are not smooth enough. Considering plausibility and speed Motion-DVAE is clearly the best choice. We observe that the more optimization iterations are performed with the proposed method, the better accuracy, but also the less smooth predictions. This result is not surprising since optimizing the model for specific samples leads to a loss of generalization.

H.2 Experiments on i3DB

We provide supplementary experiments on i3DB [89]. The i3DB dataset [89] contains in-the-wild videos with medium to heavy occlusions due to person-environment interaction. It provides global 3D annotations on joint coordinates that we use for computing evaluation metrics. We initialize our method using an off-the-shelf 2D pose estimation [92] and SPIN [90], a method providing frame-wise SMPL pose estimates.

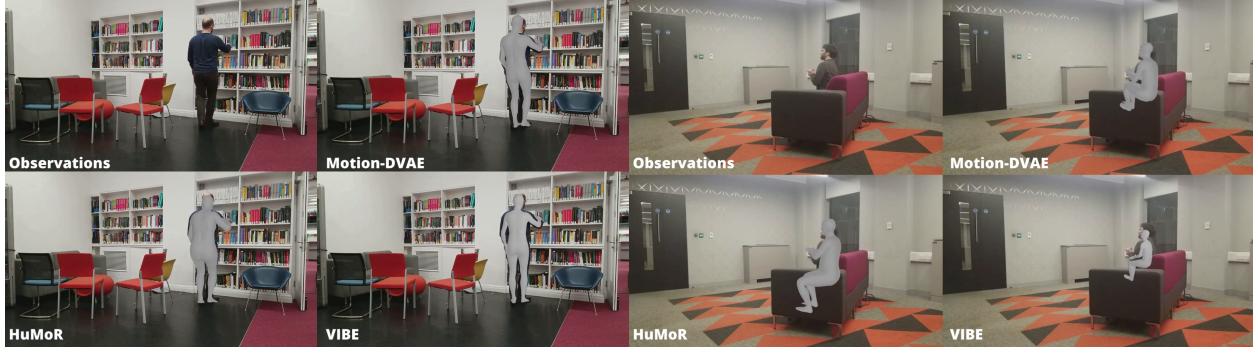


Figure 2: Qualitative results for motion estimation from videos.

H.2.1 Evaluation on RGB videos

We evaluate the proposed method on i3DB [89]. Following [86], we only use six scenes with accurate annotations (Scenes 05, 07, 10, 11, 13, and 14). Due to this dataset’s low number of videos, we evaluate our approach in a cross-validation setting, with each fold: 4 videos for training, 1 for validation, and 1 for testing. As in the previous experiment, the method is evaluated in the two regression and optimization modes.

As for the motion denoising experiment, we use very high values for noise parameter λ for learning the denoising process on the noisy training set. However, we use a robust learned noise model in the optimization mode. This gives a Student-t prior distribution for the noise on the 3D observations. Fitting a robust noise model on noisy observation sequences allows us to adapt to the observed noise in an unsupervised manner, which is expected to be beneficial in terms of prediction accuracy. Indeed, as explained in the section 4.3 of the main paper, this unsupervised adaptation allows us to find a compromise between the noisy observations and the Motion-DVAE output, depending on the estimated noise variance. A lower variance will give more importance to the observations and vice versa.

We use the open-source implementations of SPIN [90], VIBE [95], and HuMoR [86] for comparing with SOTA algorithms. We also include MVAE [96] in the comparison, but taking the results given in the paper of [86]. We do not have the running time for this method, but we can assume that it is similar to HuMoR since MVAE can be thought of as ablation of the HuMoR CVAE using the same optimization procedure. The running time of 2D keypoints detection by off-the-shelf models is not considered in speed calculation since it is used by all methods. However, we add SPIN execution time to our method for a fair comparison with SPIN [90] and other methods. Results are presented in Tab. 1.

Table 1: Results on i3DB: We compare the proposed method (bottom part) and SOTA (top part) performance in terms of global and local Mean Per Joint Positional Error (MPJPE) (cm) and execution time (sec/frame) on visible (Vis) and occluded (Occ) joints. We achieve an accuracy competitive with the SOTA methods while showing reasonably high speed.

| | Speed ↓ (sec/frame) | G-MPJPE ↓ | | | MPJPE ↓ | | |
|---------------------|------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | All | Vis | Occ | All | Vis | Occ |
| SPIN [90] | <u>0.05</u> | - | - | - | 14.95 | 11.89 | 23.90 |
| VIBE [95] | 0.02 | 116.46 | 90.05 | 192.55 | 15.08 | 12.06 | 23.78 |
| MVAE [96] | - | 40.91 | 37.54 | 50.63 | 19.17 | 16.00 | 28.32 |
| VPoser-t [86] | 1.05 | <u>31.88</u> | <u>28.84</u> | <u>40.63</u> | 16.36 | 12.81 | 26.57 |
| HuMoR [86] | 11.37 | 28.68 | 26.01 | 36.37 | 15.29 | 12.57 | 23.10 |
| SPIN-t (ours) | 0.25 | 42.45 | 36.54 | 59.71 | 14.28 | 11.66 | 21.94 |
| Regression (ours) | 0.26 | 41.41 | 36.55 | 55.45 | 13.82 | <u>11.54</u> | 20.40 |
| Optimization (ours) | 0.56 | 41.09 | 36.01 | 55.78 | <u>14.08</u> | 11.48 | <u>21.58</u> |

We can notice that the proposed SPIN-t improves the SPIN predictions regarding the MPJPE by adjusting the global rotation. SPIN-t outperforms all SOTA methods in terms of MPJPE. However, VPoser-t is more accurate globally (i.e., in terms of G-MPJPE) by about 8 cm.

Motion-DVAE outperforms all SOTA models by more than 1cm in terms of MPJPE. Regarding the global error, Motion-DVAE is similar to MVAE and significantly outperforms VIBE, which gives inaccurate results for occluded joints. HuMoR obtains better global predictions but is prohibitively slow for real-world applications.

Table 2: Ablation study: MPJPE and G-MPJPE are in cm.

| | | G-MPJPE | | MPJPE | |
|----------------|--------------|---------|-------|-------|-------|
| Full | HuMoR [86] | Vis | Occ | Vis | Occ |
| | Regression | 26.01 | 36.37 | 12.57 | 23.10 |
| | Optimization | 36.55 | 55.45 | 11.54 | 20.40 |
| NN output | Regression | 36.01 | 55.78 | 11.48 | 21.58 |
| | Optimization | 39.54 | 56.64 | 13.69 | 22.32 |
| No prior | Regression | 36.42 | 55.33 | 11.80 | 21.95 |
| | HuMoR [86] | 37.51 | 58.77 | 11.64 | 20.90 |
| | Optimization | 36.75 | 58.68 | 11.56 | 21.74 |
| No noisy train | Regression | 68.19 | 81.72 | 32.30 | 47.56 |
| | Optimization | 39.97 | 57.64 | 12.26 | 22.49 |
| Gaussian noise | Optimization | 35.84 | 56.93 | 11.52 | 21.76 |

Qualitative results are shown in Fig. 2. When there are no occlusions, all methods perform great visually. However, with occlusions, there are clear differences between predictions. VIBE fails to estimate the global translation, and the arms pose is not as accurate as other methods. Motion-DVAE and HuMoR give good local predictions, with more realistic floor contacts for the latter. HuMoR seems to underestimate the distance between the person and the camera for global predictions as the prediction looks closer than the person on the image, while Motion-DVAE predicted translation is a bit too far relative to the camera. Qualitative results for videos are made available in the supplementary material.

H.2.2 Ablation study

We also perform an ablation study to evaluate the impact of the different components of the denoising procedure learned in an unsupervised fashion. We chose i3DB [89] for the ablation study because this dataset has many occlusions, enabling us to evaluate the model components for dealing with occlusions. Results are shown in Tab. 2. Most ablations are tested in regression (Reg) and optimization modes (Opt), except for the last experiment, which only concerns the optimization mode. Results are compared with the performance without any ablation (Full).

The first ablation experiment, "NN output", predicts the initial state, the latent motion, and the noise by taking directly the expectations outputted by the associated inference model networks successively instead of sampling the approximate posterior distributions as explained in the section 4.3 of the main paper. This approximation is made in most works, but neural network decoders are not linear functions. Thus taking the expectation of the latent representation and decoding it does not give the expectation of the network output. Results obtained without the proposed inference scheme are less accurate by about 1.5 cm in the regression mode and 0.5 cm in the optimization mode.

Next, we remove $\mathcal{L}_{KL}^{DVAE}(\phi, \omega)$ from the equation (13) of the main paper for both training and optimization ("No prior"). We compare our results with HuMoR [86] by setting the motion prior weight to 0 in optimization mode. Surprisingly, when removing the motion prior, HuMoR obtains better results for occluded and visible joints. On the contrary, removing the motion prior penalizes our method in regression and optimization modes. In the optimization mode, the results get better on every metric using the motion prior. This shows that the proposed motion prior is efficient for unsupervised optimization.

In "No noisy train", we skip the unsupervised training step on noisy data: we perform direct inference and optimize from Motion-DVAE learned on AMASS. As expected, the results deteriorate significantly in the regression mode because we use a model trained on clean data for processing noisy observations. In optimization mode, we do not reach the accuracy of the regression mode with no ablation. This demonstrates the efficiency of the introduced unsupervised learning procedure for denoising new unknown motions.

Finally, in "Gaussian noise" we test the proposed approach in optimization mode using a Gaussian noise prior distribution instead of Student-t. The results are improved on visible joints but deteriorate on occlusions. However, the results in the regression mode are still better than the results in the optimization mode, which shows the excellent generalization of the unsupervised learning approach.

Supplementary References

- [84] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends® in Machine Learning*, 15, 2021.

- [85] 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 (ICCV)*, 2019.
- [86] 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 (ICCV)*, 2021.
- [87] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [88] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.
- [89] Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J. Mitra. Imapper: Interaction-guided scene mapping from monocular videos. *ACM Transactions on Graphics*, 38, 2019.
- [90] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *International Conference on Computer Vision (ICCV)*, 2019.
- [91] Jorge Nocedal and Stephen J Wright. Nonlinear equations. *Numerical Optimization*, 2006.
- [92] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [93] Stuart Geman. Statistical methods for tomographic image reconstruction. *Bulletin of International Statistical Institute*, 4, 1987.
- [94] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, 2016.
- [95] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [96] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Transactions on Graphics*, 39, 2020.