

Adversarial Parametric Pose Prior

Andrey Davydov *

andrey.davydov@epfl.ch

Anastasia Remizova

anastasia.remizova@epfl.ch

Victor Constantin

victor.constantin@epfl.ch

Sina Honari

sina.honari@epfl.ch

Mathieu Salzmann

mathieu.salzmann@epfl.ch

Pascal Fua

pascal.fua@epfl.ch

CVLab, EPFL

Abstract

The Skinned Multi-Person Linear (SMPL) model represents human bodies by mapping pose and shape parameters to body meshes. However, not all pose and shape parameter values yield physically-plausible or even realistic body meshes. In other words, SMPL is under-constrained and may yield invalid results.

We propose learning a prior that restricts the SMPL parameters to values that produce realistic poses via adversarial training. We show that our learned prior covers the diversity of the real-data distribution, facilitates optimization for 3D reconstruction from 2D keypoints, and yields better pose estimates when used for regression from images. For all these tasks, it outperforms the state-of-the-art VAE-based approach to constraining the SMPL parameters. The code will be made available at https://github.com/cvlab-epfl/adv_param_pose_prior.

1. Introduction

The SMPL model [17] is now widely used to parameterize body poses and shapes [15, 21]. However, it offers no guarantee to produce realistic human bodies when random values are passed as its inputs. This complicates its usage within an optimization, regression, or generative frameworks, where it is desirable that any sample drawn be plausible.

To mitigate this issue, several approaches have been used. In [2], this is addressed by introducing a Gaussian Mixture Model (GMM) learned on the SMPL pose. Unfortunately, due to its unbounded nature, it still allows poses far away from any training example and potentially unrealistic. In SMPL-X [22], a Variational Autoencoder

(VAE) is used instead to learn a low-dimensional representation of the SMPL parameters. This choice was motivated by the ability of VAEs to model the distribution of valid data samples in the latent space as a multivariate Gaussian, which was shown to better approximate the data distribution than classical models, such as GMMs, while also facilitating sampling at test time. In both approaches, the learned prior is then used together with other losses in an optimization-based framework that aims at finding plausible human meshes. Unfortunately, VAEs have drawbacks. First, their learned prior tends to be mean-centered and to discard part of the original data distribution that are far away from it. Furthermore, its Gaussian prior is unbounded, like that of GMMs. Hence, one can also sample latent values far away from any in the training set and produce unrealistic bodies. Adversarial training has been used to bound the parameter prediction of SMPL in regression-based frameworks [5, 11]. However, this requires balancing the adversarial loss with other losses. More importantly, no explicit prior has been learned in such cases, as this training needs to be repeated for each new task.

In short, these approaches make it necessary to balance different losses and do not bound the inputs of the SMPL model. In contrast, we aim at learning a prior, that once learned can be used in an optimization or learning-based frameworks, without the requirement of enforcing constraints on it. In other words, the learned prior should be integrated as part of the SMPL model and the model can be optimized only on the target loss, where the learned prior is not added as an extra constraint. To this end, we learn an explicit prior, that constrains the input of the SMPL model to be realistic poses via adversarial training. This has to be done only once and independently of the target application so that no further adversarial training is needed. Hence, it does not require balancing multiple losses in the downstream tasks.

Furthermore, one can use a bounded distribution, such as

*This work was supported in part by the Swiss National Science Foundation

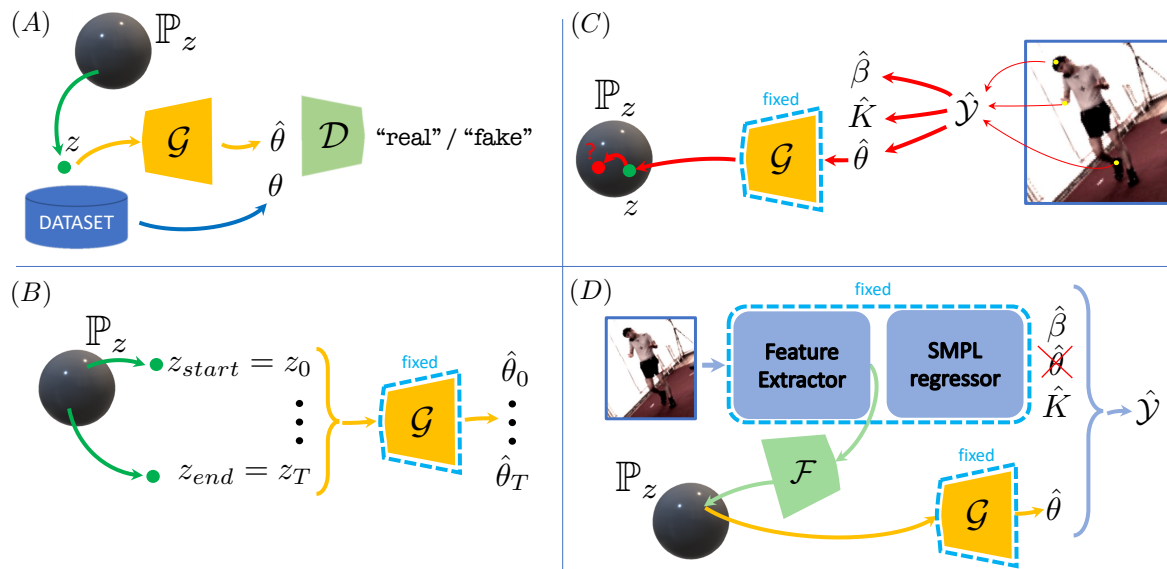


Figure 1. Multiple uses of the proposed approach. (A) The adversarial game between generator \mathcal{G} and discriminator \mathcal{D} guarantees that the former provides realistic body poses. (B) Given “start” and “end” latent vectors, one can render the whole sequences of plausible and smooth body interpolations. (C) To optimize for corresponding poses given target keypoints \mathcal{Y} , akin to VPoser [22], the pretrained generator \mathcal{G} can be used as an implicit pose prior. (D) The pretrained \mathcal{G} can be dropped in as a pose prior in a pretrained off-the-shelf human mesh regressor. It improves the prediction quality of the regressor.

uniform or spherical ones, in the input space of the learned prior, which facilitates plausible sample generation and also its integration in regression frameworks, as by limiting the output of the preceding component that is passed as input to the learned prior, one is always guaranteed to have a plausible human representation. Once trained, our model can be used in many different settings without further retraining as shown in Fig. 1. We introduce GAN-based pose prior learning technique that consistently outperforms the VAE-based state-of-the-art approach for both optimization- and regression-based approaches to human body pose recovery [22]. Also, we make a comparison between different choices of latent spaces, out of which the spherical one brings the most benefit.

2. Related Work

2.1. Body Representation

There have been many attempts at modeling the human body. The earliest ones split the body into several simpler shapes and combine them into a unified model. The introduction of several datasets consisting of diverse body scans [25] has ushered the age of learnable body models. The SMPL body model [17] constitutes one of the most successful and easy-to-use models. It uses a combination of PCA coefficients to model the shape and a regressor that poses the body from the joints angles. Several extensions have since then been proposed. SMPL-H [26] includes a more detailed hand model, thus removing one of the limitations of the original model. More recently, SMPL-X [22],

adds facial expressions to the previous models. Instead of using a mesh representation, NASA [4] encodes the human body as a signed distance function. Instead of learning an explicit prior, [24] first predicts the 3D pose and then constrains it using physics-based optimization. This must be done for every video, rather than being a part of a prior integrated into a model. The “LIMP” model [3] has been proposed and evaluated directly on meshes, while we learn a prior for SMPL input parameters. In this paper we focus only on SMPL as it is widely used in the community. Here, we focus on the SMPL model as we are interested in modeling the human body itself, and favor a mesh representation, which inherently provides correspondences across, e.g., video frames.

2.2. SMPL Parameter Estimation

Since the introduction of the SMPL body model, many approaches have aimed to estimate the SMPL parameters given either an image [5, 9, 11, 23], some labels, such as 2D or 3D pose [1, 2, 22], or body silhouettes [14]. Depending on whether they are optimization- or regression-based, they can be divided into three categories.

Optimization Models. The first category consists of optimizing the SMPL parameters so as to minimize an objective function defined in terms of different pose or image descriptors. Such descriptors can be 2D and/or 3D joint locations [1, 2, 22], silhouettes [14], or dense correspondences [7]. SMPLify [2] constitutes one of the first such methods. It uses a GMM to model the pose space and optimizes the SMPL parameters so as to match 2D joint loca-

tions. The unboundedness of the GMM prior may result in the optimization producing unrealistic poses. In [22], the GMM is replaced by a VAE to model the pose space distribution. While a VAE can model more complicated distributions than a GMM, it remains unbounded. Furthermore, the mean-centered nature of VAEs makes it cover the original data distribution only partially, because it poorly represents data samples away from its distribution’s means. As we will show later, our approach learns a better and smoother coverage of the data while addressing the unbounded nature of these approaches. In [28], a normalizing flow (NF) is used to model a body prior. The mapping from the latent distribution to the SMPL pose is invertible by construction, which makes it suitable for weakly- or self-supervised optimization. On the other hand, our generative model does not have any constraints on the architecture and the training procedure is less demanding. Moreover, this NF model [28] explores only a Gaussian distribution in the latent space, while our approach is distribution-agnostic, as we show in the experiments.

Regression Models. The second category consists of directly regressing the SMPL parameters given an input image. Human Mesh Recovery (HMR) [11] is one of the initial methods that applies such a technique using deep neural networks. Since then it has been used in several other works, such as [5, 9, 23]. These methods minimize an adversarial prior together with other target losses. Therefore, the resulting representation is only usable within the learned model, since no explicit prior is learned. By contrast, we learn an explicit bounded prior, which needs to be trained only once. Then, the weights of this learned prior can be frozen and it can be used in any regression approach by mapping a feature space to the learned prior latent space.

Combined Models. The two previous categories are compatible with each other and can be used together. SPIN [13] mixes the two by fine-tuning the regression estimate with an optimization procedure. EFT [10] takes the pretrained regression network of [13] and uses its weights as an implicit body prior. It fine-tunes the weights of the network for every sample in a weakly-annotated dataset to obtain the body parameters. Although we demonstrate our method separately on optimization and regression-based tasks, it can be used in the combined approach, as these models merge the individual components from optimization and regression-based approaches.

3. Method

To constrain the SMPL poses we rely on a GAN approach [6]. It involves two competing networks, a *generator* \mathcal{G} and a *discriminator* \mathcal{D} . The generator samples vectors \mathbf{z} , known as *latent vectors*, from a set $\mathbb{P}_z \subseteq \mathbb{R}^d$ and generates a SMPL pose vector $\hat{\Theta} = \mathcal{G}(\mathbf{z})$, which can be passed to the

SMPL decoder ϕ to generate a body mesh $\mathcal{B} = \phi(\hat{\Theta}, \beta)$, where β denotes the SMPL body shape parameters. The task of the discriminator is to distinguish poses generated in this manner from those of a large dataset of poses known to be realistic. By contrast, the generator is trained to produce poses that fool the discriminator. This process is shown in Fig. 1 (A).

Constraining shape and pose. Training our models, we leave the SMPL shape parameters β untouched to be able to compare with other models, i.e. VPoser [22], as they only learn a prior for pose. Moreover, the shape part of the model is already data-driven (with PCA). However, the PCA weights for the shape are also unbounded by the model and eliminating this problem is also worth further research. We trained a model in such combined fashion, more information on this can be found in the supplementary material.

3.1. Distribution over Latent Vectors

GAN-based approaches [12, 20, 27] have used several types of distributions from which to draw their latent vectors, including Gaussian, Uniform, and Spherical distributions. To test all three, we learn three different sets of latent vectors:

- GAN-N: $\mathbf{z}_N \sim \mathbb{P}_z = \mathcal{N}(0, \mathcal{I}_d) \subset \mathbb{R}^d$ (*Normal*)
- GAN-U: $\mathbf{z}_U \sim \mathbb{P}_z = \mathcal{U}_{[-1,1]^d} \subset \mathbb{R}^d$ (*Uniform*)
- GAN-S: $\mathbf{z}_S \sim \mathbb{P}_z = \mathcal{S} \subset \mathbb{R}^d$ (*Spherical*)

where the spherical vectors are sampled by drawing vectors \mathbf{z}_N from a normal distribution and computing $\mathbf{z}_S := \frac{\mathbf{z}_N}{\|\mathbf{z}_N\|_2}$.

The unbounded nature of the Gaussian distribution $\mathcal{N}(0, \mathcal{I})$ prevents sampling from rare modes and may make the resulting prior suffer from the same drawbacks as GMMs and VAEs when used in regression tasks. While the Uniform distribution does not have such a limitation, it imposes artificial bounds $[-1, 1]^d$ that do not have a clear meaning in the output pose space. Intuitively, because one can smoothly move from one pose to another, we would rather expect a latent pose space to be continuous, without strict boundaries as the uniform space. The desirable properties of the latent space, such as continuity and boundedness are all inherent to the Spherical distribution. Our experiments show that, in practice, it does indeed tend to perform better than the others.

3.2. Training

We define the generator \mathcal{G} in our GAN architecture to have the same structure as the decoder of the VAE in VPoser [22]. As for our discriminator \mathcal{D} , we base our structure on that of the HMR approach [11], using $K + 1$ discriminators, one for each joint angle and one for the whole set of pose parameters.

As can be seen in Fig. 1 (A), we draw samples from the latent space \mathbb{P}_z and train the generator to map them to the SMPL pose space. The discriminators are trained to distinguish the SMPL pose vectors Θ , obtained by a real pose dataset, from the ones produced by the generator $\hat{\Theta}$. The training loss function, aiming to balance the two opposing goals of the generator and discriminator, can thus be expressed as

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{\Theta} [\log \mathcal{D}(\Theta)] + \mathbb{E}_{z \sim \mathbb{P}_z} [\log(1 - \mathcal{D}(\mathcal{G}(z)))] . \quad (1)$$

When training image-generating GANs, the usual practice is to take the ratio of training steps for \mathcal{G} and \mathcal{D} to be 10:1 because the former requires more updates to produce realistic “fake” samples. In our setting the competing models have similar capacities. Hence, using that ratio yields a severe mode collapse. Thus, we update the discriminator weights 10 times for every update of the generator.

We train the model using the common splits of the AMASS dataset [19], following the procedure for VPoser in SMPL-X [22]. As AMASS provides SMPL-H parameters [26], the body pose is a bit different from the original SMPL [2, 17] one; it only contains $K = 21$ joint angles, with 2 angles from SMPL having been moved to SMPL-H “hands” articulations.

3.3. Using the Generator as a Universal Prior

Being trained once, our generative model can be used in many applications. We introduce some of them here and present the results in the following section.

Interpolation in Latent Space. One way to gauge the quality of a latent representation is to check how smooth the interpolation from one latent vector to another is. Ideally, the transition should vary equally in each step from the source to the target samples, rather than most of the transformation occurring in only a few steps. To check this, we randomly select N samples $\{\mathcal{B}_1^t, \dots, \mathcal{B}_N^t\}$ from the test set and optimize (similar to the paragraph below) for every body \mathcal{B}_i^t the corresponding latent vector \mathbf{z}_i^r that yields the closest mesh $\mathcal{B}_i^t = \phi(G(\mathbf{z}_i^r), \beta)$. For each pair of such latent vectors, we construct an interpolation sequence $\{\mathbf{z}_0^r, \mathbf{z}_1^r, \dots, \mathbf{z}_T^r\}$ and $\{\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_T\}$ using either linear interpolation

$$\mathbf{z}_t^r = \left(1 - \frac{t}{T}\right) \mathbf{z}_0^r + \frac{t}{T} \mathbf{z}_T^r \quad (2)$$

for GAN-N and GAN-U, or spherical interpolation

$$\mathbf{z}_t^r = \frac{\sin\left(\left(1 - \frac{t}{T}\right)\theta\right)}{\sin\theta} \mathbf{z}_0^r + \frac{\sin\left(\frac{t}{T}\theta\right)}{\sin\theta} \mathbf{z}_T^r \quad (3)$$

for GAN-S, with $\mathcal{B}_t^r = \phi(G(\mathbf{z}_t^r), \beta)$, and θ representing the angular distance between two points \mathbf{z}_0^r and \mathbf{z}_T^r on a sphere.

We discuss spherical interpolation (SLERP) in more detail, including the proof of Eq. 3 for high dimensions, in the supplementary material. Ideally, the samples \mathcal{B}_{t-1} and \mathcal{B}_{t+1} should be roughly equidistant from \mathcal{B}_t , indicating smooth transitions. The per-vertex mesh distance is computed as follows:

$$d(\mathcal{B}^r, \mathcal{B}^t) = \frac{1}{N_{verts}} \sum_{v=1}^{N_{verts}} \|\mathcal{B}_v^r - \mathcal{B}_v^t\|_2 , \quad (4)$$

where v sums over the vertices. The sampling process is depicted in Fig. 1 (B).

Optimization from Keypoints. Given the 2D joint targets \mathcal{Y} obtained from a monocular observation and assuming neutral SMPL shape parameters $\beta = 0$, our goal is to find the SMPL pose parameters $\hat{\Theta}$ that produce the target \mathcal{Y} using the SMPL model ϕ , which translates from SMPL space $(\hat{\Theta}, \beta)$ to the space of body meshes \mathcal{B} . Fig. 1 (C) describes the idea. The recovered mesh can be projected to 2D joints using camera parameters, i.e., $\Pi(\phi(\hat{\Theta}, \beta)) = \mathcal{Y}$, where Π is the camera projection function. To find the optimal SMPL parameters, one can minimize $L(\Pi(\phi(\hat{\Theta}, \beta)), \mathcal{Y})$, where L is a loss function such as the L_2 distance between the 2D mesh joints and the corresponding target joints.

To better constrain the pose output by SMPL, we make use of our pose prior. That is, instead of directly optimizing $\hat{\Theta}$, we optimize a vector \mathbf{z} in the GAN’s latent space and obtain the corresponding $\hat{\Theta}$ by feeding \mathbf{z} to the generator \mathcal{G} . Altogether, we therefore solve the optimization problem

$$\min_{\mathbf{z}} \|\Pi(\phi(\mathcal{G}(\mathbf{z}), \beta)) - \mathcal{Y}\|_2^2 . \quad (5)$$

Image-to-Mesh Regression. Our GAN models can also be used as drop-in priors to improve existing pretrained image-to-mesh algorithms [10, 11, 13]. To demonstrate this, we start from the model of [10], whose architecture is a *Resnet50* model based on the one of [11]. It is pretrained on pseudo ground-truth COCO [16] dataset obtained by [10]. We then inject our model into it as shown in Fig. 1 (D). More specifically, we introduce an additional MLP \mathcal{F} that maps intermediate features of *Resnet50* to a latent vector \mathbf{z} of the pre-trained SMPL prior, which then can be mapped by $\mathcal{G}(\mathbf{z})$ into the pose vector Θ of SMPL. One can then decode pose parameters Θ into a human mesh \mathcal{B} using the SMPL model ϕ . In turn, \mathcal{F} can be used in conjunction with the pre-trained SMPL prior \mathcal{G} and the SMPL decoder ϕ to reconstruct a complete body mesh, which can then be compared to the ground-truth targets. We used this process to train only the \mathcal{F} in an end-to-end setup and obtain the corresponding body mesh \mathcal{B} .

4. Experiments

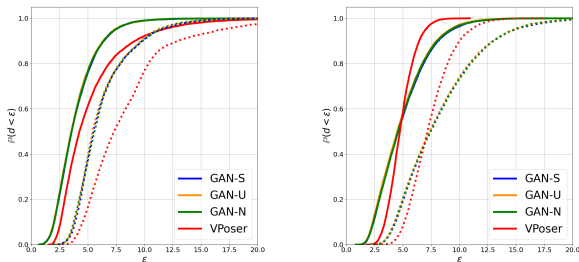
We now compare the three versions of our approach to sampling the latent vectors, GAN-N, GAN-U, and GAN-S,

Table 1. Statistics for Recall experiment on *Train/Test* splits of AMASS dataset [19]. The values are the mean, variance and medians of distances between real samples and closest neighbors among generated samples for every model (in *mm*). GAN models demonstrate indistinguishable behaviour, while VPoser [22] provides consistently larger discrepancy with real set.

	Train set		Test set	
	$\mu \pm \sigma$ (\downarrow)	median (\downarrow)	$\mu \pm \sigma$	median
GAN-S (<i>Ours</i>)	4.0 \pm 1.9	5.5	6.3 \pm 2.6	5.5
GAN-U (<i>Ours</i>)	3.9\pm1.9	5.4	6.2\pm2.5	5.4
GAN-N (<i>Ours</i>)	4.0 \pm 1.9	3.6	6.2\pm2.5	5.6
VPoser [22]	5.2 \pm 3.2	4.3	6.3 \pm 4.0	7.3

with the VPoser VAE-based approach of SMPL-X [22] and the NF model of [28].

4.1. Dataset Coverage



(a) Recall. “Do *real* samples live in latent spaces?” (b) Precision. “How close are *fake* samples to *real*?”

Figure 2. Empirical estimation of data coverage of generative models for both *Recall* (a) and *Precision* (b). Experiments with data from the *Train* set are drawn with solid lines, and from the *Test* set with dashed lines. Higher means better in all charts.

An ideal latent representation should cover the whole space of realistic human poses and nothing else. In other words, it should have good *Recall* and *Precision*. By recall, we mean that all samples in the training set should be well approximated by poses our model generates. By precision, we mean that these generated poses should never deviate too far from the training set. While recall indicates how well the generated samples cover the dataset distribution, precision indicates how realistic the generated samples are. We define these metrics as follows.

Recall. To evaluate recall, we use our pose generator to produce SMPL poses and take the shape parameters to be a zero vector, which yields a neutral body shape. Hence, for all models we produce a body mesh $\mathcal{B} = \phi(\mathcal{G}(\mathbf{z}), \beta)$ given a sampled latent vector \mathbf{z} and a fixed β . We first generate $6M$ samples from the pose generator of each model. Then, given a ground-truth body \mathcal{B}^t from either the training or test set, we select the generated body $\mathcal{B}^r = \phi(\mathcal{G}(\mathbf{z}^r), \beta)$ with minimum vertex-to-vertex distance Eq. 4.

We then repeat this operation for $10k$ bodies randomly sampled from either the training or test set. We report the mean, variance and median of the resulting distances in Table 1. In Fig. 2a, we plot the cumulative distribution $\mathbb{P}(d < \epsilon)$ given the values $d(\mathcal{B}^r, \mathcal{B}^t)$ for each training sample. Note that all versions of our approach deliver consistently higher values than the VPoser [22], indicating that our models better cover the entire distribution.

In Fig. 3, we show the t-SNE projection [18] of the resulting SMPL Θ^r pose vectors superposed on the Θ^t vectors that were used to generate the training examples. All GANs cover the space spanned by the training examples more completely than VPoser, which is consistent with the previous result. In other words, our learned prior can represent more diverse poses than the other ones.

We provide more *Recall* experiments for various sampling strategies in the supplementary material.

Precision. Our approach to computing precision mirrors the one we used for recall. We randomly generate $10k$ latent points \mathbf{z}^r from every model, and for each sample $\mathcal{B}^r = \phi(G(\mathbf{z}), \beta)$, with a fixed β , look in the training or test datasets for the nearest neighbor in terms of the distance given in Eq 4. If the latent representation only produces poses similar to those seen in training, this distance should be consistently small.

As shown in Fig. 2b, GAN models tend to produce meshes that are further away from the training distribution than the VAE model. This could be interpreted as a failure to produce realistic poses. However, these unseen samples correspond to plausible bodies. They are nothing but the result of semantic interpolation that GANs implicitly learn from the data. In Fig. 4, we show the worst 10 samples based on the distance metric of Eq. 4 and their nearest neighbors from the training set. Note that all of these samples look realistic even though they are far from the closest neighbor in the dataset. This indicates that our generators are able to produce novel samples that were not observed in the training set, however, this is more observed in GAN-S and GAN-U compared to GAN-N, as GAN-N generates samples closer to its mean, hence deviating less to more diverse poses.

4.2. Interpolation in Latent Space

To evaluate our model on the first application described in Section 3.3 and in Fig. 1 (B), we randomly select $N = 128$ samples $\{\mathcal{B}_1^t, \dots, \mathcal{B}_N^t\}$ from the test set, and, for each pair, we construct the corresponding interpolation sequence $\{\mathbf{z}_0^r, \mathbf{z}_1^r, \dots, \mathbf{z}_T^r\}$ and $\{\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_T\}$. We use the mean per-vertex position error (Eq. 4) between body meshes \mathcal{B}_i and \mathcal{B}_j and compute pairwise distances between their body meshes by $\Delta(\mathcal{B}_i, \mathcal{B}_j)$, which we represent by Δ_{ij} .

The minimal transformation Δ_{ij} between every consec-

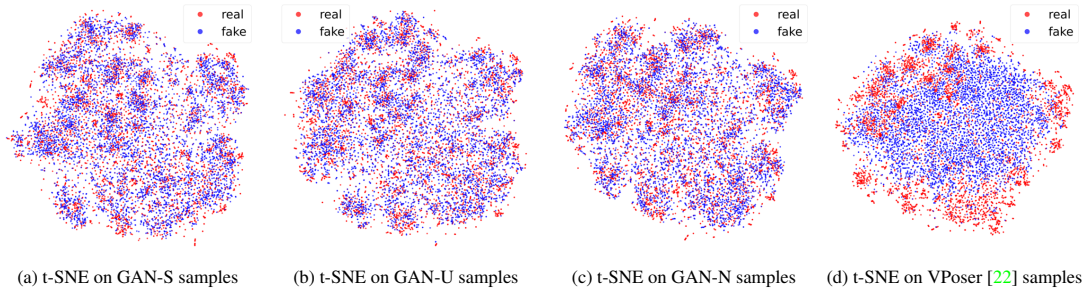


Figure 3. t-SNE projections of “real” samples from the training set and of “fakes” generated by GAN-S (a), GAN-U (b), GAN-N (c) and VPoser [22] (d) models.



Figure 4. “Worst” samples according to the precision metric Δ in Figure 2b. For each GAN model we show 10 samples with the largest distance to the first nearest neighbor (INN) in the training set, ordered from the worst sample on the right. Generated samples themselves are absolutely plausible human bodies, despite being away from training samples. Note that in GAN-S and GAN-U these samples are further away from INN compared to GAN-N.

itive bodies is equal to $\Delta_{0T} = \frac{\Delta(\mathcal{B}_0^i, \mathcal{B}_T^i)}{T}$. For different initial pairs, this value can be drastically different, as the corresponding bodies might be very close to or very far from each other. Hence, we normalize the transformation Δ_{ij}

of every sequence by the expected average transformation Δ_{0T} , yielding $\bar{\Delta}_{ij}$, which should be 1 in the minimal case. Note, however, that such an ideal case can typically only be achieved by going through physically-impossible poses, for

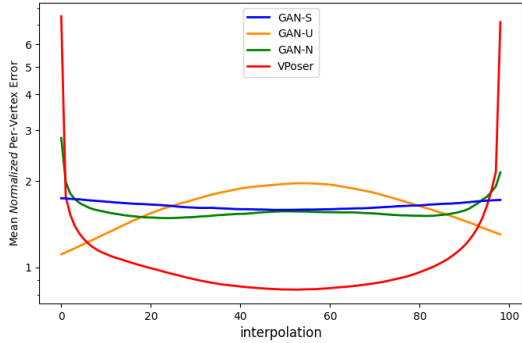


Figure 5. Average normalized mesh transformation of consecutive generated samples in interpolations between source and target examples. x axis depicts the number of iterations and the y axis shows the normalized mesh deviation in log-scale. The ideal transformation should be represented as a flat line. VPoser applies most of transformations at the beginning or at the end of the interpolations. GAN-N reduced this affect, while GAN-S obtains the smoothest transitions.

Table 2. Comparison of interpolation smoothness between different models. max -to- min ratios of consecutive deformations for every pair of interpolation is measured according to Eq. 6. Mean, median, and standard deviation of R_{ij} over a set of interpolations is then reported for each model.

	$\mu \pm \sigma$ (\downarrow)	median (\downarrow)
GAN-S (<i>Ours</i>)	4.3 ± 3.0	3.5
GAN-U (<i>Ours</i>)	7.1 ± 7.7	5.1
GAN-N (<i>Ours</i>)	$5.7 \times 10^2 \pm 5.3 \times 10^2$	8.6
VPoser [22]	$1.8 \times 10^5 \pm 6.9 \times 10^5$	9.1

instance by shrinking the arms to go from a body with arms up to one with arms down. Hence, actual transformations will typically obtain values higher than 1, but a good latent space should nonetheless yield values as constant as possible throughout the entire interpolation steps, indicating a smooth gradual transition.

We illustrate the behavior of different models in Fig. 5, where we average the consecutive interpolation distances $\bar{\Delta}_{ij}$ across all pairs. The closer the curve is to being horizontal, the smoother is the interpolation. The VPoser [22] curve indicates that interpolation with this model is subject to jumping from B_0^i to B_T^i in very few steps, either at the beginning or the end of the sequence, and the remaining steps are spent performing small pose adjustments. This effect can also be seen in GAN-N but to a lesser degree. In contrast, GAN-S and GAN-U both produce smooth interpolations, with a slight advantage to the spherical distribution. More plots with interpolation distances for particular pairs can be found in the supplementary material.

To measure the smoothness of an interpolation, we compute the ratio between the maximal and minimal transitions in a sequence Δ_{ij} , i.e.,

$$R_{ij} = \frac{\max(\Delta_{ij})}{\min(\Delta_{ij})}. \quad (6)$$

Table 3. Reporting 3D pose error (in mm) for the bodies recovered through optimization from 2D joint targets.

	P-MPJPE (\downarrow)
GAN-S (<i>Ours</i>)	84.3
GAN-U (<i>Ours</i>)	90.7
GAN-N (<i>Ours</i>)	85.4
NF [28]	89.6
VPoser [22]	90.1
GMM [2]	92.3

Table 4. Reporting 3D pose error (in mm) for the bodies recovered through regression from input images.

	MPJPE (\downarrow)	P-MPJPE (\downarrow)
HMR _(COCO EFT) [10]	75.91	68.02
GAN-S (<i>Ours</i>)	63.24	56.32
GAN-U (<i>Ours</i>)	64.28	57.30
GAN-N (<i>Ours</i>)	67.19	59.92
VPoser [22]	69.18	61.71

We report the mean, variance and median of the ratios R_{ij} in Table 2. These results confirm our previous conclusions: GAN-S yields the smoothest interpolations, closely followed by GAN-U. In Fig. 6 we show interpolation between two pairs of samples for different models.

We provide more ablation on interpolation experiments in the supplementary material.

4.3. Mesh Optimization from 2D Joints

We now turn to the optimization application discussed in Section 3.3 and in Fig. 1 (C). Given the 2D joint locations, the optimal latent vector can be found using an iterative optimization algorithms. In our experiments, we use L-BFGS-B for all models. For GAN-S and GAN-U, we renormalize the estimated z given their input bound at each optimization step. Projecting the 3D body joints to the observed 2D joints on the image via the camera project Π requires access to the camera parameters and the body orientations. We obtain them in the same way as in [2].

In Table 3, we report the 3D pose errors (after rigid alignment) obtained by recovering the SMPL parameters $\hat{\Theta}$ from 2D joints for the H3.6M dataset [8], following *Protocol 2*. Note that GAN-S again yields the best results for this application, this time closely followed by GAN-N. By contrast, GAN-U yields a higher error, indicating its input bounds makes it less suitable for the optimization-based tasks. To compare against the the NF model [28], we trained a Real-NVP version of it ourselves because we did not have access to the code or the weights.

4.4. Image to Mesh Regression

We train the approach to body regression from an image introduced in Section 3.3 and in Fig. 1 (D) for the three versions of our approach and for VPoser [22]. We report accuracy results on test data in Table 4 in terms of 3D pose error of the recovered bodies after Procrustes alignment (P-

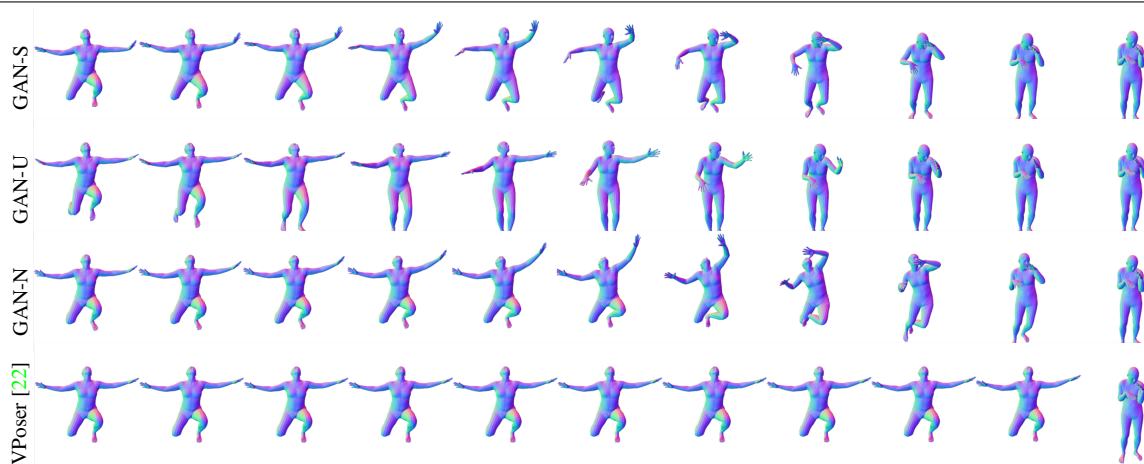


Figure 6. Examples of interpolations for different generative models. All GAN models provide smooth (yet semantically very different) interpolations, while VPoser [22] sticks at one pose for most of the path and “jumps” into the ending pose. More examples can be found in the supplementary material.

MPJPE), according to *Protocol 2* of [8]. Once again GAN-S performs the best, with GAN-U and GAN-N outperforming VPoser. Our models deliver better accuracy than [10], even though its accuracy is reported for pre-trained models.

4.5. Limitations

Using our GAN priors, the diversity of their learned distributions is limited by their training sets, which might not be diverse enough for downstream tasks. This is, however, similar to any other model that learns a distribution such as VPoser or HMR, which is also limited by its training distribution.

5. Conclusion

In this paper we proposed a simple yet effective prior for SMPL model to bound it to realistic human poses. We show

that the learned prior can cover the diversity of the training distribution, while also being capable of generating novel unseen samples. Further, we demonstrate the advantage of learning such a prior in generation, optimization, and regression based frameworks, where the learned prior can be trained once and for all, then used in any downstream task without requiring to balance different losses. Our results show that using a spherical distribution for the learned prior leads to smoother transition in the generated samples from the latent space, while also yielding more accurate results for optimization- and regression-based tasks, indicating this prior is better suited for learning human poses.

References

- [1] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in

- the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019. 2
- [2] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *European Conference on Computer Vision*, 2016. 1, 2, 4, 7
- [3] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. LIMP: Learning Latent Shape Representations with Metric Preservation Priors. *European Conference on Computer Vision*, 2020. 2
- [4] Boyang Deng, J. P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA: Neural Articulated Shape Approximation. In *European Conference on Computer Vision*, 2020. 2
- [5] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Košecká, and Ziyang Wu. Hierarchical Kinematic Human Mesh Recovery. In *European Conference on Computer Vision*, pages 768–784. Springer, 2020. 1, 2, 3
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3
- [7] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10884–10894, 2019. 2
- [8] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 7, 8
- [9] W. Jiang, N. Kolotouros, G. Pavlakos, X. Zhou, and K. Daniilidis. Coherent Reconstruction of Multiple Humans from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3
- [10] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020. 3, 4, 7, 8
- [11] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-To-End Recovery of Human Shape and Pose. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 4
- [12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 3
- [13] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. In *Conference on Computer Vision and Pattern Recognition*, 2019. 3, 4
- [14] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M.J. Black, and P.V. Gehler. Unite the People: Closing the Loop Between 3D and 2D Human Representations. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2
- [15] Ren Li, Meng Zheng, Srikrishna Karanam, Terrence Chen, and Ziyang Wu. Everybody Is Unique: Towards Unbiased Human Mesh Recovery. *arXiv Preprint*, 2021. 1
- [16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, pages 740–755, 2014. 4
- [17] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M.J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM SIGGRAPH Asia*, 34(6), 2015. 1, 2, 4
- [18] L.J.P.v.d. Maaten and G.E. Hinton. Visualizing High Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 2008. 5
- [19] 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*, pages 5442–5451, Oct. 2019. 4, 5
- [20] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. *arXiv preprint arXiv:2103.17269*, 2021. 3
- [21] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation. In *International Conference on 3D Vision*, 2018. 1
- [22] 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 *Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019. 1, 2, 3, 4, 5, 6, 7, 8
- [23] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 803–812, 2019. 2, 3
- [24] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2
- [25] Kathleen M Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, and Scott Fleming. Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary. Technical report, Sytronics Inc Dayton Oh, 2002. 2
- [26] J. Romero, D. Tzionas, and M. Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics*, 36(6):245, 2017. 2, 4
- [27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 3
- [28] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *European Conference on Computer Vision (ECCV)*, pages 465–481, 2020. 3, 5, 7