Normalizing Flows on the Product Space of SO(3) Manifolds for Probabilistic Human Pose Modeling

Olaf Dünkel[12], Tim Salzmann[3], Florian Pfaff [4]

[1] Max Planck Institute for Informatics,   [2] Karlsruhe Institute of Technology,
[3] Technical University of Munich,   [4] University of Stuttgart

oduenkel@mpi-inf.mpg.de, tim.salzmann@tum.de, pfaff@ias.uni-stuttgart.de

Abstract

Normalizing flows have proven their efficacy for density estimation in Euclidean space, but their application to rotational representations, crucial in various domains such as robotics or human pose modeling, remains under-explored. Probabilistic models of the human pose can benefit from approaches that rigorously consider the rotational nature of human joints. For this purpose, we introduce HuProSO3, a normalizing flow model that operates on a high-dimensional product space of SO(3) manifolds, modeling the joint distribution for human joints with three degrees of freedom. HuProSO3's advantage over state-of-the-art approaches is demonstrated through its superior modeling accuracy in three different applications and its capability to evaluate the exact likelihood. This work not only addresses the technical challenge of learning densities on SO(3) manifolds, but it also has broader implications for domains where the probabilistic regression of correlated 3D rotations is of importance. Code will be available at https://github.com/odunkel/HuProSO.

1. Introduction

Modeling uncertainties in high-dimensional Euclidean product spaces is a well-explored research problem [5–7, 12, 13, 24]. However, these methods often fall short in representing problems with an inherent rotational nature. This limitation is evident in scenarios like correlated object rotations in a scene or the orientation of animals in swarm behaviors, where the problems are modeled by a product space of SO(3) manifolds. A particularly relevant example is human pose modeling, a problem of multiple correlated joint rotations. Accurately modeling human poses as densities on the Cartesian product of such joint rotations holds significant value for various fields, including computer vision and robotics.
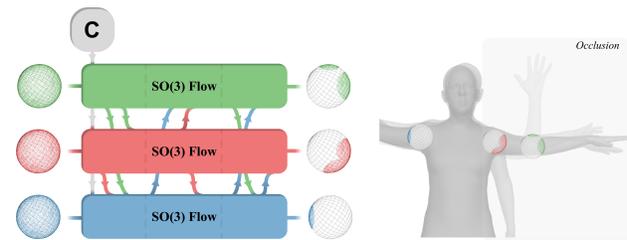
A probabilistic model defined on SO(3) manifolds rep-

Figure 1. Application of the normalizing flow for occcluded joints. Left: A normalizing flow defined on SO(3) manifolds enables the learning of expressive human pose distributions, incorporating a context vector **c** for conditioning. Right: Renderings of probable poses given condition **c**, which is the observation of the human with the left arm occluded. The right arm's pose is estimated with high certainty, while the left arm demonstrates varied but realistic poses due to the occlusion.

resenting joint rotation distributions is relevant for human pose and motion estimation. This model acts as a reliable prior in scenarios with unclear, noisy, incomplete, or absent observations [3, 14, 15, 29, 32], with an example being depicted in Fig. 1. Additionally, such a model is crucial in human-robot interaction for implementing risk-aware planning and control, necessitating the representation of human movements as normalized probabilistic densities [21].

Normalizing flows are recognized for their capability to learn normalized densities. Yet, their application has predominantly been in learning joint densities within Euclidean spaces, not fully addressing the properties of human pose, which is significantly influenced by the rotational nature of the human joints. Although some previous approaches in probabilistic human pose modeling have employed normalizing flows to learn distributions parameterized by rotational measures [14, 36], they fall short in learning normalized densities on the SO(3). This is due to their Euclidean space-based approaches, which lack a continuous bijective mapping to SO(3).

To overcome these shortcomings, we introduce a normalizing flow defined on the product space of SO(3) man-

ifolds, accurately capturing the density of human joint rotations. By defining flow layers that explicitly operate on SO(3), the learned density is normalized on the considered space of rotations with three degrees of freedom. The joint PDF on multiple SO(3) manifolds expresses the statistical dependencies between human joints accordingly. To achieve this, the SO(3) manifolds are explicitly linked using a nonlinear autoregressive conditioning not restricted to the common Gaussian setting.

We demonstrate that respecting the manifold structure of SO(3) in the normalizing flow design improves the performance for multiple applications that involve a probabilistic model of human pose, outperforming state-of-the-art approaches in most configurations.

In summary, our work makes three key contributions:

- We present a normalizing flow model for learning normalized densities on the high-dimensional product space of SO(3) manifolds. This is enabled by conceptualizing a flow layer that specifically operates on SO(3) and lifting the model to a product space via an expressive nonlinear autoregressive conditioning approach.
- Introducing HuProSO3, a probabilistic **Hu**man pose model on the **Pro**duct Space of **SO(3)** manifolds, we elucidate the application of our presented normalizing flow model in various applications where a probabilistic model of humans is relevant.
- We showcase HuProSO3's effectiveness and adaptability through applications like probabilistic inverse kinematics and 2D to 3D pose uplifting, outperforming several state-of-the-art methods in tasks involving probabilistic human pose models.

2. Related work

2.1. Normalizing Flows on Rotational Manifolds

Normalizing flows model complex distributions through a series of sequential flow layers. They learn a normalized density, represented by the change of variables formula:

$$p(x) = \pi \left(T^{-1}(x) \right) \left| J_{T^{-1}(x)} \right|, \qquad (1)$$

where $J_{T^{-1}(\cdot)}$ is the Jacobi-matrix of the inverse of the diffeomorphic transformation $T : \mathbb{R}^d \to \mathbb{R}^d$ and its inverse $T^{-1}(\cdot)$, where a diffeomorphic transformation is invertible and both, T and its inverse T^{-1}, are differentiable.

Several methods adapt normalizing flows, typically applied in Euclidean spaces, for rotational manifolds, including SO(3). Rezende et al. [27] successfully apply the Möbius transformation, circular splines, and a non-compact projection for learning densities on such manifolds. Falorsi et al. [8] develop a normalizing flow in Euclidean space that is then mapped to SO(3), but this approach involves non-unique mappings from the Lie algebra to the Lie group. Contrarily, Liu et al. [18] propose a novel Möbius coupling layer and quaternion affine transformation, facilitating the learning of normalizing flows directly on SO(3).

Flows for Product Spaces of Rotational Manifolds. Papamakarios et al. [23] present a strategy for an autoregressive model that allows learning densities on high-dimensional Euclidean spaces. Building on this, Stimper et al. [33] introduced a PyTorch package for defining normalizing flows on Cartesian products of 1-spheres using two-dimensional embeddings of angular quantities. However, this method does not extend to general n-spheres or SO(3) manifolds. Additionally, Glüsenkamp [11] facilitated learning on rotational manifolds, including tori and 2-spheres, using an autoregressive conditioning approach. Yet, his framework is limited by a fixed conditioning sequence and lacks support for flows on SO(3) manifolds.

2.2. Learning Human Pose Distributions

In the domain of human pose estimation, models traditionally use joint rotations as parameters [1, 16] to reflect the skeleton's rotational characteristics. Pioneering methods for learning the unconditional human pose distribution include a GMM [2] and a VAE [25], both proving effective as pose priors. Later advancements utilized normalizing flows in Euclidean space [36] using the 6D representation [38].

Subsequent works [4, 34] that learn a human pose prior have highlighted the inadequacy of a Gaussian assumption for the joint rotation parameterization due to their unbounded nature. So, they propose different ways to account for this issue in the design of their generative models for the human pose. Davydov et al. [4] show that a spherical noise distribution in the latent space of GAN-based approach results in a distribution with more realistic human poses and a smoother latent space. Tiwari et al. [34] model a manifold on the product space of SO(3) that represents the set of feasible human poses.

Probabilistic techniques have been effectively employed to infer human pose distributions from image inputs [14, 30–32]. Sengupta et al. [30] utilize the Matrix-Fisher distribution for learning the rotational distributions of individual joints on SO(3). Meanwhile, Kolotouros et al. [14] model joint rotations using a 6D representation and employ normalizing flows to learn densities, incorporating an additional loss term to account for the orthonormality of the columns in the 6D representation. Sengupta et al. [32] present a novel approach to model human pose distributions on the product space of SO(3). This method factorizes the joint PDF by using autoregressive conditioning along the human kinematic tree. It respects the manifold structure of joint rotations by learning a normalizing flow on the Lie algebra and subsequently mapping it to the Lie group. Voleti et al. [35] builds upon [22] to model the pose and applies it, e.g., to inverse kinematics from sparse 3D key points.
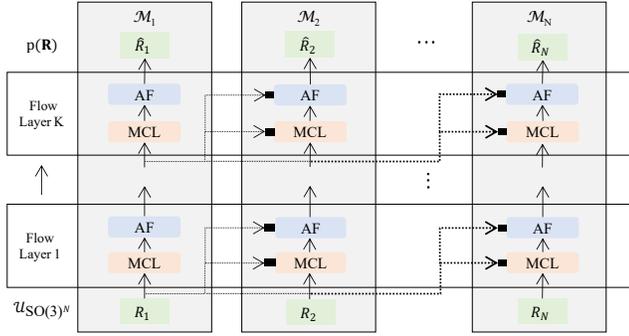
Figure 2. Overview of the components of the normalizing flow: The flow is defined on a product space of N manifolds $\mathcal{M}_i = SO(3)$. It includes K flow layers that transform samples R_i from a uniform distribution on SO(3) to samples of the learned distribution $\hat{R}_j$. The flow is composed of a Möbius coupling layer (MCL) and a quaternion affine transformation (AF). Vertical arrows indicate the flow of SO(3) samples through layers, while dotted arrows represent autoregressive conditioning using an MLP (black boxes).

3. Method

In this section, we describe our method for deriving and implementing the following desiderata: First, as outlined in Sec. 3.2, each normalizing flow layer is designed to respect and operate within the SO(3) manifold. Second, we ensemble these individual flow layers to model a density on the product space of SO(3) manifolds. Finally, we present the probabilistic human pose model HuProSO3, which comprises and instantiates the aforementioned methodology.

3.1. Problem Statement

Given the special orthogonal group SO(3) that is defined as the set of rotation matrices R

$$\{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^T\mathbf{R} = I, \quad \det \mathbf{R} = 1\}, \tag{2}$$

we aim to learn a PDF $p(\mathbf{R}|\mathbf{c})$, where the random variable $\mathbf{R} = [\mathbf{R_1}, ..., \mathbf{R_N}]$ is defined on a product space $\mathbf{P} = \prod_i^N \mathcal{M}_i$ of N SO(3) manifolds $\mathcal{M}_i = SO(3)$ and the context vector $\mathbf{c} \in \mathbb{R}^L$ conditions the PDF.

3.2. Normalizing Flows on a Product Space of SO(3) Manifolds

The structure of our model, as illustrated in Fig. 2, involves multiple flow layers that convert a uniform density on SO(3) into the target density for each manifold $\mathcal{M}_i = SO(3)$. Samples R_i are drawn from a uniform distribution on SO(3). Then, the samples are mapped to the target distribution $\hat{R}_i$ through the sequential application of diffeomorphic flow layers. The normalizing flow is trained by maximizing the likelihood of the target sample that is propagated through the flow layers (NLL Loss).

The manifolds are connected in an autoregressive manner to allow for learning a joint PDF. The flow is defined on SO(3) by leveraging a Möbius coupling layer and a quaternion affine transformation [18] as invertible flow layers on SO(3). The joint likelihood of the product space $\mathbf{P}$ is learned using randomly masked nonlinear autoregressive conditioning. We present these mechanisms in more detail in the following.

Möbius Coupling Layers. The core principle of normalizing flows involves applying diffeomorphic transformations in each layer, which ensures that the original density remains normalized. However, defining these transformations for the SO(3) manifold, unlike in Euclidean space, presents challenges. We must ensure that both the forward transformation and its inverse operate on SO(3) and do not map to points outside this manifold. To this end, we apply the Möbius transformation, which expresses parameterized distributions on spheres.

A rotation matrix $R \in SO(3)$ can be described using any two of its three orthonormal three-dimensional unit column vectors $(\mathbf{x}_1|\mathbf{x}_2|\mathbf{x}_3)$ [38]. Thus, each column represents a three-dimensional unit vector defined on the unit sphere $\mathbf{x}_i \in \mathcal{S}^2$, $i \in \{1, 2, 3\}$, parameterized in Euclidean coordinates $\mathbf{x}_i \in \mathbb{R}^3$.

The Möbius transformation of a point $\mathbf{y} \in \mathcal{S}^\mathcal{D}$, given a parameter $\omega \in \mathbb{R}^{D+1}$, is defined as

$$h_\omega(\mathbf{y}) = h(\omega, \mathbf{y}) = \frac{1 - ||\omega||^2}{||\mathbf{y} - \omega||}. \tag{3}$$

It projects a point $\mathbf{y}$ to a new location on the same sphere $\mathcal{S}^D$. The Möbius transformation is leveraged to design the Möbius coupling layer [18]—an invertible flow layer on SO(3) that transforms a valid rotation matrix $R = (\mathbf{x}_1|\mathbf{x}_2|\mathbf{x}_3)$ into a different valid rotation matrix $R' = (\mathbf{x}'_1|\mathbf{x}'_2|\mathbf{x}'_3)$. The Möbius coupling layer ensures orthogonality in the transformed rotation matrix as follows: The first column $\mathbf{x_1}' = \mathbf{x}_1$ remains unchanged, providing the basis for adapting the second column vector. This second column, modeled as a point $\mathbf{x}_2 \in \mathcal{S}^2$ is transformed into a new point $\mathbf{x}'_2 = h(\omega_{\mathbf{x_1}}, \mathbf{x_2}) \in \mathcal{S}^2$, using the Möbius transformation. The transformation parameter

$$\omega_{\mathbf{x_1}} = \xi(\mathbf{x_1}) = g(\mathbf{x_1}) - \mathbf{x_1}(\mathbf{x_1} \cdot g(\mathbf{x_1})) \tag{4}$$

is derived from $\mathbf{x_1}$. Here, $g(\cdot) \in \mathbb{R}^3$ represents an MLP output, projected onto a plane perpendicular to $\mathbf{x}'_1$. Consequently, the resulting column vector $\mathbf{x}'_2$ is orthogonal to $\mathbf{x}'_1$. The final step in constructing the rotation matrix involves computing the third column, $\mathbf{x}'_3$, using the cross product $\mathbf{x}'_3 = \mathbf{x}'_1 \times \mathbf{x}'_2$. This ensures the completion of the rotation matrix R' with a vector orthogonal to the first two columns.

In a rotation matrix, the first column specifies two out of three rotational degrees of freedom, while the second column determines the remaining degree. Since the first column $\mathbf{x}_1$ remains unchanged in a single Möbius coupling layer, each layer only modifies the rotation along one degree of freedom. To address this, we employ permutations

in the conditioning sequence of consecutive flow layers. For example, we use the Möbius transformation to derive a new column $\mathbf{x}_1' = f(\omega_{\mathbf{x_2}}, \mathbf{x_1})$, where $\omega_{\mathbf{x_2}}$ is calculated from $\mathbf{x_2}$. By applying such permutations changes for several flow layers all rotational degrees of freedom can be modeled. These permutations, being invertible and volume-preserving transformations with an absolute Jacobian of 1 [24], can be integrated into normalizing flows.

While a single Möbius coupling layer effectively warps distributions on SO(3) nonlinearly, it only transforms one rotational degree of freedom via the second column of a rotation matrix. However, this strategy does not directly allow the modification of multiple rotational degrees simultaneously by moving or scaling distributions on SO(3), as [18] illustrates. To account for this, we also apply an affine quaternion transformation [18] in our normalizing flow.

Quaternion Affine Transformation. The quaternion affine transformation allows learning an affine transformation on the 3-sphere, a double-cover of SO(3). It, thus, realizes a global rotation and a shifting operation on SO(3). The quaternion affine transformation

$$f_{\mathrm{q}}(\mathbf{q}) = W\mathbf{q} \cdot (||W\mathbf{q}||)^{-1} \tag{5}$$

is applied to a quaternion $\mathbf{q} = \mathrm{m}_{\mathbf{R} \to \mathbf{q}}(\mathbf{R})$, where $\mathbf{q}$ is computed from the considered rotation matrix $\mathbf{R}$ and the invertible 4×4 matrix W is constructed using *svd* decomposition. The parameters of the matrices of the *svd* decomposition are learned during training. As shown in Fig. 2, this transformation follows each Möbius coupling layer. Maintaining the quaternion's real part positive is required for the transformation's invertibility. This constraint does not hinder expressiveness, as $\mathbf{q}$ and $-\mathbf{q}$ denote the same rotation.

Joint PDF on a Product Space. While the described Möbius and affine quaternion transformations effectively operate on a single SO(3) manifold, they do not suffice to construct a normalizing flow for the product space $\mathbf{P} = \prod_i^N SO(3)$. We address this by introducing an autoregressive masking strategy, designed to learn a joint PDF on such a product space. Masked autoregressive flows [24] learn PDFs defined on n-dimensional Euclidean spaces $\mathbf{z} \in \mathbb{R}^n$. They implement autoregressive conditioning by applying consecutive MADE [10] blocks for linking different dimensions with the Gaussian conditionals parameterized by

$$z_i = u_i \exp f_{\alpha_i}(\mathbf{z}_{1:i-1}) + f_{\mu_i}(\mathbf{z}_{1:i-1}), \tag{6}$$

with the scalar functions $f.(\cdot)$ that output the mean and log standard deviation of the conditional i given all previous dimensions. The term $u_i \sim \mathcal{N}(0, 1)$ represents noise.

The expression in Eq. (6) links individual dimensions. However, this approach does not account for manifolds that cannot be expressed by Cartesian products in Euclidean space, e.g., the 2-sphere or SO(3). Using standard Gaussian conditionals falls short in capturing the dependencies of non-Euclidean manifolds.

To address these challenges, we first apply autoregressive conditioning based on random variables defined on SO(3). For this purpose, the joint density is decomposed autoregressively:

$$p(\mathbf{R}) = \prod_i^N p(\mathbf{R}_i | \mathbf{R}_{1:i-1}), \tag{7}$$

where each conditional is defined on $\mathbf{R}_i \in$ SO(3) and conditioned on preceding SO(3) manifolds $\mathbf{R}_{1:i-1} \in \prod_1^{i-1} SO(3)$. Such a decomposition is generally restricted to a fixed order of conditioning, which does not capture arbitrary dependencies between different SO(3) manifolds. To mitigate this problem, we randomly vary the order of conditioning in each flow layer, similar to [23], which results in more expressive flows [24]. We achieve this by permuting the SO(3) rotation sequences between autoregressive layers, leveraging the invertibility of permutations.

Instead of using a linear coupling with the conditionals parameterized as Gaussians for Euclidean space as proposed in [23] and expressed in Eq. (6), we compute the parameters of the considered manifold $\mathcal{M}_i$ autoregressively. This involves a nonlinear map $g_{\mathrm{c}}\left(\mathbf{x}_{12}^{(1:i-1)}\right)$ with parameters learned during training to condition the current manifold's distribution on the preceding ones. We parameterize the SO(3) rotations of previous dimensions $\mathbf{R}_{1:i-1}$ in the continuous 6D representation [38] with $\mathbf{x}_{12}^{(i:i-1)}$. This conditioning is then integrated into both the affine transformation and the Möbius coupling layer.

The affine transformation is conditioned on the previous dimensions by using a neural network to compute the matrix $\mathbf{W}_i = g_{\mathrm{c-W}}^{(i)}\left(\mathbf{x}_{12}^{(1:i-1)}\right)$. We realize the autoregressive conditioning in the Möbius coupling layer by computing the parameters of the Möbius transformation based on the first rotation matrix' column of the considered dimension $\mathbf{x}_1^{(i)} \in \mathbb{R}^3$ and the previous dimensions $\mathbf{x}_{12}^{(1:i-1)} \in \mathbb{R}^{3 \times 2 \times (i-1)}$ in 6D representation $g_{\mathrm{c-M}}\left(\left[\mathbf{x}_1^{(i)}, \mathbf{x}_{12}^{(1:i-1)}\right]\right)$.

Conditioning the Joint PDF. The joint PDF can be conditioned on additional information $\mathbf{c}$ by augmenting the set of conditions of each term in the chain rule of probability with the condition $\mathbf{c}$:

$$p(\mathbf{R}|\mathbf{c}) = \prod_i p(\mathbf{R}_i | \mathbf{R}_{1:i-1}, \mathbf{c}). \tag{8}$$

3.3. Learning Human Pose Distributions

We leverage our normalizing flow to learn a probabilistic model of the human pose. We use the SMPL [2] model to parameterize the human pose with 21 joints. After removing static joints in the AMASS database, our model describes a density $p(\mathbf{R}|\mathbf{c})$ on a product space of $N = 19$ SO(3) manifolds. We call this normalizing flow HuProSO3,

a probabilistic **Hu**man pose model on the **Pro**duct Space of **SO(3)** manifolds.

We deploy HuProSO3 in several applications, the first being to learn an unconditional human pose prior. In this case, we do not condition, i.e., $\mathbf{c} = \emptyset$. This pose prior acts as a generative model for sampling realistic human poses and can also be used for evaluating the probability of a specific pose that is parameterized by joint rotations $p\left(\mathbf{R} = \{R_{\text{joint},1}, ..., R_{\text{joint},19}\}\right)$. Second, we inject a condition $\mathbf{c}_{\text{feat}}$ into the model to learn a density that is specific to a given context. To handle varying contexts and to remove computations in each flow layer, we add an additional MLP $\mathbf{c} = g(\mathbf{c}_{\text{feat}})$ that computes the relevant features that are injected into the normalizing flow, as outlined in Eq. (8). HuProSO3 generally supports arbitrary conditions. We present conditioning with 2D and 3D key points, where $\mathbf{c}_{\text{feat}} \in \mathbb{R}^{J \times k}$ with $J = 21$ joints and $k \in \{2,3\}$ for conditioning with 2D and 3D key points.

To demonstrate HuProSO3's ability to capture human pose distributions, we adapt the masking strategy from [3]. Parts of the context features $\mathbf{c}'_{\text{feat}} = \mathbf{m} \cdot \mathbf{c}_{\text{feat}}$ are removed by applying the mask $\mathbf{m} = [m_1, m_2, ..., m_J]$ with $m_i \in \{0,1\}$ indicating whether a joint position is accessible. To allow conditioning with an arbitrary number of joints during evaluation, we seek a different masking strategy to [3] and we mask each joint with varying probabilities $p_m \in [0,1]$ during training. Therefore, the model has learned to handle varying numbers of occluded joints.

4. Experiments

While HuProSO3's design ensures that all flow transformations are on SO(3) and capture nonlinear dependencies between the SO(3) manifolds, we now also demonstrate experimentally that it has sufficient expressiveness to accurately model human pose distributions. We, therefore, evaluate HuProSO3 on different applications that involve a probabilistic model of the human pose. First, we learn an unconditional human pose prior that showcases HuProSO3's capability of learning the intricate distribution human poses.

Second, we condition the distribution demonstrating HuProSO3's capabilities of injecting observations or other conditions for learning task-specific distributions and correctly adapting the respective uncertainties arising from different sources of conditioning. For this task, we also evaluate cases when only partial information is given. This includes scenarios with partial information, such as occluded joints, where HuProSO3 effectively reasons about uncertainties due to missing information.

4.1. Unconditional Pose Prior

To demonstrate HuProSO3's effectiveness in an unconditional setting, we train a human pose prior using the

Table 1. Summary of precision and recall statistics for AMASS. The reported values represent the cumulative geodesic distances for all joint rotations between samples from the dataset and the evaluated pose prior.

	Test (mean [median])		Train (mean [median])	
	Recall	Precision	Recall	Precision
GAN-S [4]	3.76 [3.34]	4.51 [4.23]	3.57 [3.34]	4.38 [4.13]
6D NF	3.66 [3.16]	4.50 [4.00]	3.55 [3.32]	4.42 [4.10]
Ours	**3.44 [2.95]**	**4.24 [3.71]**	**2.93 [2.64]**	**3.90 [3.59]**

AMASS database [19] with its standard dataset split. We then compare its ability to generate realistic human poses against various other human pose priors.

Metrics. In general, the likelihood is the best metric for evaluating how well a model has learned a distribution. However, we are the first to provide normalized joint densities of the human pose distribution on SO(3) manifolds. Therefore, our evaluation relies on comparing generated samples from HuProSO3 with those from previous approaches. We follow an experimental pipeline similar to [4] to evaluate whether human poses sampled from the considered pose prior deviate from samples from the dataset distribution (precision) and whether the pose prior has captured the variety of poses in the dataset (recall). For recall, we compare 1k dataset samples against 100k poses sampled from our model, calculating the minimum error (nearest neighbor). Precision is evaluated by comparing 100k dataset samples against 1k model-generated samples, again computing the minimum error for these model samples.

Since we aim to evaluate how well the model captures the distribution $p(\mathbf{R})$ on the product space of SO(3), we evaluate precision and recall based on the sum of all $J = 21$ geodesic distances $d_{\text{geo}}(R_{i,k}, R_{j,k})$ of the rotations $R_{i,k}$ and $R_{j,k}$ that correspond to the k-th SMPL joint of pose i and j, respectively, where the geodesic distance is computed by

$$d_{\text{geo}}(R_{i,k}, R_{j,k}) = \arccos \frac{\text{tr}\left(R_{i,k}^T R_{j,k}\right) - 1}{2}. \quad (9)$$

Baselines. In our evaluation, HuProSO3 is compared with the GAN-S human pose prior [4] and a 6D normalizing flow similar to models in [36, 37]. We train GAN-S from scratch using the standard parameter configuration provided by [4]. To compare against a 6D normalizing flow, we trained an autoregressive [23] neural spline flow [7] using the normflows library [33]. We provide additional results for Pose-NDF [34] as a pose prior in see Tab. 8 (Appendix D.1).

Results. In Tab. 1, we present the mean and median of precision and recall for both, training and test datasets. The results illustrate that HuProSO3 has captured the training dataset's distribution best. The means of precision and recall metric, both, are lower depicting that the model generates poses that correspond to realistic poses from the dataset

Table 2. MGEO [rad] and MPJPE [mm] results for the IK task on the AMASS test split. Performance comparisons are made for optimization-based (<u>best</u>) and probabilistic methods (**best**). Probabilistic methods are assessed using a single random sample and the average of 10 random samples.

Method	MPJPE (1 | 10)	MGEO (1 | 10)
VPoser [25]	26.6 | -	<u>0.230</u> | -
GAN-S [4]	12.1 | -	0.224 | -
Pose-NDF [34]	<u>0.2</u> | -	0.289 | -
HF-AC [32]	33.2 | 21.2	0.308 | 0.251
Ours	8.2 | **5.5**	0.163 | **0.121**

but it also captures the variety of the seen poses during training. We observe similar results for the median metric. Moreover, while having learned the training distribution more accurately, our model does not overfit and it outperforms other methods on the precision and recall metric for the test dataset as well. Precision and recall curves, log likelihood comparisons for training and test datasets, and qualitative joint correlation illustrations are provided in the appendix. Additionally, rendered poses in Fig. 12 confirm HuProSO3's ability to generate realistic and diverse human poses (see Appendix C.3).

4.2. Inverse Kinematics

In this experiment, we condition the learned density on the 3D position of the $J = 21$ SMPL joints to solve the human pose inverse kinematic (IK) problem: The goal is to learn the PDF $p(\mathbf{R}|\mathbf{c})$ that is defined on SO(3) and conditioned on the joint positions in 3D space $\mathbf{c} \in \mathbb{R}^{J \times 3}$.

Metrics. To assess the models, we use the Mean Geodesic Distance (MGEO) across joint rotations for rotational error and the Mean Per Joint Position Error (MPJPE) to evaluate the accuracy of learned densities in relation to error propagation in forward kinematics. Effective probabilistic models are expected to produce samples consistent with the given conditioning, with accuracy increasing upon evaluating more samples. Therefore, we test probabilistic methods using both a single random sample and an average of 10 random samples.

Baselines. We compare our approach against both, optimization-based and learning-based baselines by evaluating the MGEO and MPJPE metrics. We consider the following learned pose priors: We use the IK solver provided by [25] focusing solely on pose without shape optimization. Furthermore, we evaluate PoseNDF [34] utilizing the pretrained model and the existing pipeline. As implemented in the published code by [34], we use the optimization objective

$$\mathcal{L}_{\text{IK}} = \sum_{k=1}^{N_{\text{test}}} \sum_{i=1}^{J} ||\mathbf{x}_{\text{gt},i,k} - \hat{\mathbf{x}}_{i,k}||_2, \quad (10)$$

which compares the ground truth joint position i $\mathbf{x}_{\text{gt},i,k}$ with the predicted position $\hat{\mathbf{x}}_{i,k} = FK(\mathbf{R}_{i,k})$ after applying for-

Table 3. MPJPE | MGEO for different types of occlusions: Left leg (L), left arm and hand (A+H), and right shoulder and upper arm (S+UA).

Method	L	A+H	S+UA
Pose-NDF [34]	28.3 | 0.341	38.3 | 0.360	<u>2.1</u> | 0.333
GAN-S [4]	<u>16.8</u> | <u>0.240</u>	<u>29.6</u> | <u>0.260</u>	18.8 | <u>0.241</u>
HF-AC [32] (N=10)	101.3 | 0.289	81.4 | 0.301	72.2 | 0.270
Ours (N=10)	**13.9** | **0.165**	**30.2** | **0.201**	**12.7** | **0.162**

ward kinematics to the optimized rotation and all N_{test} samples of the test dataset. We utilize the optimization process from [34], omitting the temporal smoothness term.

Similarly, we use the trained GAN-S model and we optimize its latent code $\mathbf{z}$ to minimize the loss in Eq. (10) with $\hat{\mathbf{x}}_i = \mathcal{G}(\mathbf{z})$ using the LBFGS optimizer [17].

In addition, we use the implementation of HuMani-Flow [32], which implements the normalizing flow for learning the ancestor-conditioned density on a single SO(3) manifold. Different from its original design, which conditions on visual features from a CNN encoder, we adapt it to condition on 3D keypoint positions and train the normalizing flow from scratch. We refer to it with HF-AC.

Results. The low errors, as depicted in Tab. 2, show that HuProSO3 is capable of respecting conditions accurately. Our probabilistic method has mostly comparable or better results than other pose priors optimized in latent space [4, 25]. Pose-NDF performs best on the MPJPE metric because it does not penalize a pose as long as it is realistic. For this reason, the optimization-based approach allows to match the given 3D key points nearly perfectly. However, the worse performance on the MGEO metric demonstrates that Pose-NDF models the manifold of plausible poses. Therefore, it cannot infer the most likely joint rotations for the given 3D key points, but it infers a rotation that belongs to the set of feasible joint rotations.

We evaluate the exact log-likelihood of HuProSo3 and HF-AC approach in Tab. 7 in Appendix D.1.

4.3. Inverse Kinematics with Partial Observation

We evaluate the capabilities of HuProSO3 to solve IK in the case of partially given 3D key points, e.g., when some joints are occluded. The rotational distribution of these joints can be inferred using a probabilistic human pose model. A joint PDF is essential for this task because it captures all statistical dependencies, which contrasts selecting a fixed sequence of conditioning, e.g., along the kinematic tree. We provide a dataset analysis in Appendix A. Following the methodology of [26, 34], we evaluate three examples of occlusions: the left leg (L), the left arm including the hand (A + H), and the right shoulder with the upper arm (S + UA).

We utilize the same HuProSO3 model across all experiments, conditioned on 3D keypoint positions and trained
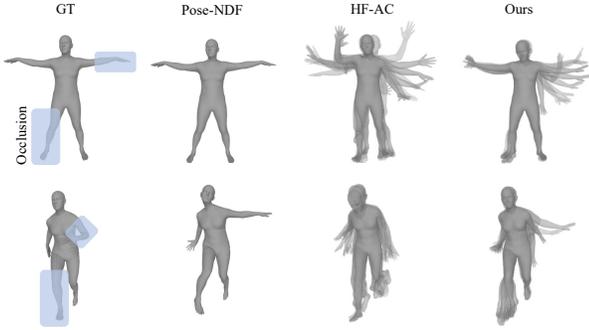
Figure 3. Qualitative results for inverse kinematics with partial occlusion (left arm and right leg). For the normalizing flow based models (HF AC and HuProSO3), we visualize 10 samples, where less likely poses are more transparent.

with the randomized masking strategy outlined in Sec. 3.3 to account for occluded joints. This approach enables the model to handle various occlusion scenarios effectively.

Metrics and Baselines. Similar to the IK setting elaborated above, we report MGEO and MPJPE and we compare with Pose-NDF and GAN-S, as detailed in Sec. 4.2. Additionally, we compare HuProSO3 to previously reported results based on the per-vertex error as in [26] and [34] (see Tab. 11 in Appendix D.4).

To account for the occluded joints, we mask the occluded joints in the optimization objective similar to the conditioning for HuProSO3 in Sec. 3.3 and we augment the optimization objective Eq. (10)

$$\mathcal{L}_{\text{IK}} = \sum_{k}^{N_{\text{test}}} \sum_{i}^{J} m_i \, ||\mathbf{x}_{\text{gt},i,k} - \hat{\mathbf{x}}_{i,k}| \, |_2, \qquad (11)$$

with the mask $\mathbf{m} = [m_1, ..., m_J]$, $m_i \in \{0, 1\}$, that cancels the contributions of the occluded joints to the loss term. The joint rotations are initialized randomly but close to 0, as stated in [34].

Results. We present the MGEO and MPJPE evaluation results in Tab. 3. Optimization-based methods Pose-NDF [34] and VPoser [25] are initialized with rotations close to the rest pose. For this reason, they perform significantly better for occluded legs, where the ground truth involves mostly straight legs. However, these methods yield higher errors when optimizing the more variable arm joints. In contrast, HuProSO3 demonstrates more accurate estimates across various types of occlusions. The results show HuProSO3 is excelling for all occlusion types on the MGEO metric. It better captures the distribution on the joint rotation space. We present the per-vertex errors in Tab. 11 in Appendix D.4.

Pose-NDF performs exceptionally well in cases of right shoulder and upper arm occlusions, likely due to the accurate optimization of arm joint rotations based on the given
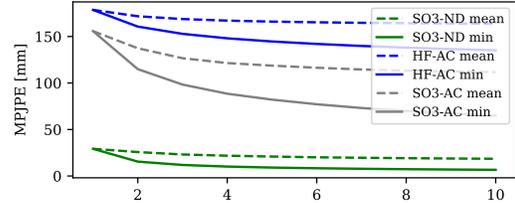

Figure 4. Minimum MPJPE and MPJPE of the mean pose for HuProSO3, ancestor-conditioned SO(3), and HF-AC for randomly occluded joints ($p_m = 0.3$) with varying numbers of samples.

Table 4. Evaluation results for the 5-point evaluation task in MPJPE [mm] and MGEO [rad]. We consider 1 sample and the average over 10 samples for HuProSO3.

Method	Pose-NDF [34]	GAN-S [4]	Ours (1 | 10)
MPJPE	<u>37.8</u>	46.5	36.4 | **27.3**
MGEO	0.405	<u>0.284</u>	0.290 | **0.208**

hand position. However, the MGEO metric is comparably high. As elucidated for the IK task, Pose-NDF does not represent a probabilistic model of the human pose but models the manifold of feasible poses. For this reason, the most likely joint rotation is not inferred, but only one possible rotation is computed. Therefore, large errors arise in particular for the rotations of the leaf joints (see, e.g., the right hand in the second row of Fig. 3).

GAN-S shows a lower MPJPE for occlusions of the left arm and hand, with performance comparable to HuProSO3 for other occlusions. This indicates the effectiveness of the latent code optimization in GAN-S, especially on the MPJPE metric. However, our model captures the joint rotation distribution more accurately, as the better MGEO metric depicts. In addition, our model predicts without the need for optimization on the joint positions, without being specifically trained for that particular type of occlusion, and it predicts in a probabilistic manner. To demonstrate how a varying number of randomly occluded joints affects performance, we refer to Fig. 14 in Appendix D.5.

For the scenario of occluded joints, HuProSO3 outperforms HF-AC on rotation- and position-based metrics. We reason that this is, firstly, due to HuProSO3 modeling the complete joint density while HuManiFlow [32] learns only the ancestor-conditioned distribution of individual joints. If we train our model with a fixed ancestor-conditioning, we still reach a better accuracy than HF-AC for randomly occluded joints (Fig. 4). See more details in Appendix D.5. We assume that designing the normalizing flow directly on SO(3) contributes to the better performance.

Fig. 3 illustrates qualitative results for IK with occluded joints. HuProSO3 accounts for the ambiguous nature of the task: While visible joints only have a low diversity, occluded joint result in diverse but likely predictions. We
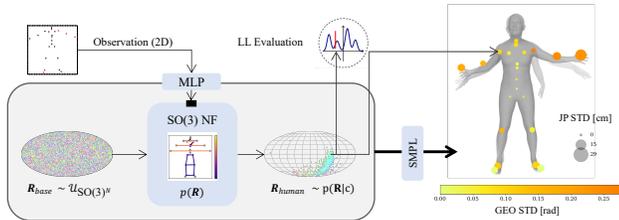
Figure 5. Overview of an example application of HuProSO3: Samples from $p(\mathbf{R}|\mathbf{c})$ are generated by propagating samples from a uniform distribution on the product space of SO(3) through the normalizing flow. Partially given 2D key points serve as conditioning, where occluded joints are depicted in red. The prior $p(\mathbf{R})$ captures the statistical dependencies between different joints. The color of the blobs depict the standard deviation of the joint rotations computed for 20 samples. The blob size relates to the standard deviation of the joint position (JP) after applying forward kinematics.

compare the diversity of the predictions to the model extracted from [32] as proposed by [32] by computing the averaged Euclidean deviation from the sample mean for all generated samples. We evaluate for 10k random samples from the AMASS test datasets. Here, our model has a sample diversity of 11 mm and 109 mm for visible and occluded joints, respectively, while the ancestor-conditioned model by [32] results in a diversity of 39 mm and 98 mm for visible and occluded joints, respectively.

5-Point Evaluation. We also assess HuProSO3 on the 5-point evaluation benchmark [22, 35], where the model must infer all joint rotations based solely on the leaf joints. Despite not being explicitly trained for this specific condition, HuProSO3 successfully infers the pose distribution from the 3D keypoint positions of the leaf joints (see Tab. 4).

4.4. 2D to 3D Uplifting via SO(3) Parameterization

In this experiment, we condition the learned density with 2D keypoint positions $\mathbf{x}_{2D} \in \mathbb{R}^{J \times 2}$ of the $J = 21$ SMPL joints and learn $p(\mathbf{R}|\mathbf{c})$, where $\mathbf{c} = g_{\text{feat}}(\mathbf{x}_{2D})$ is computed by an MLP. Then, the 3D pose is obtained by applying forward kinematics. We illustrate this setting in Fig. 5.

Metrics and Baselines. For the 2D to 3D uplifting application, we use MGEO and MPJPE as evaluation metrics, similar to the IK task. Our model is compared with Pose-NDF, GAN-S, and HF-AC. Unlike the previous experiments focusing on 3D keypoint positions (Eq. (10)), here we optimize for 2D keypoint positions.

Results. Evaluation results for the 2D to 3D uplifting task are detailed in Tab. 5, focusing on models using rotation representations for human pose. Unlike the IK task, inferring human poses from 2D data involves significant ambiguity. The results show that the considered optimization-based methods are performing worse since the optimization objective is less expressive and the results depend on the ini-

Table 5. MPJPE [mm] and MGEO [rad] results for the 2D to 3D uplifting task on the AMASS test dataset. For probabilistic models, results are shown for both a single sample and the average of 10 samples.

Method	MPJPE (1 | 10)	MGEO (1 | 10)
Pose-NDF [34]	133.1 | -	0.522 | -
GAN-S [4]	<u>61.5</u> | -	<u>0.277</u> | -
HF-AC [32]	74.4 / 56.0	0.346 / 0.278
Ours	**32.8 | 23.4**	**0.209 | 0.153**

tialization. Our model HuProSO3 outperforms other methods, affirming its effectiveness in probabilistically capturing ambiguous input conditions.

Fig. 5 qualitatively illustrates HuProSO3's applications for partially given 2D key points. The generated samples depict the arising uncertainties accordingly. More samples are visualized in Fig. 13 in Appendix C.4.

5. Conclusion

We addressed the gap in normalized density models for high-dimensional product spaces of rotational manifolds, crucial in human pose modeling, which depends on rotational quantities. Our solution, HuProSO3, is a normalizing flow that effectively learns densities on product spaces of SO(3) manifolds, capturing the rotational nature of human poses in a probabilistic way. Demonstrating its efficacy, HuProSO3 not only excels as an unconditional pose prior in generative applications but also adapts to applications that require conditioning on potentially incomplete information, as illustrated for the task of inverse kinematics and 2D to 3D uplifting via a rotational SO(3) parameterization.

Limitations and Future Work. While effective in practice, the autoregressive structure in HuProSO3 has limitations in capturing all dependencies in a high-dimensional space, as it heavily depends on the permutation operation of the conditioning sequence. Moreover, sampling from a high-dimensional autoregressive model is slow as it scales linearly with the number of manifolds. Another limitation is that our model currently only supports products of SO(3) manifolds. To better reflect the biomechanical structure of different joints, our approach could be extended to include other rotational manifolds. Additionally, the ability of our method to compute normalized densities opens up applications in human-robot collaboration. It also allows the integration as a prior or the injection of measurements in filtering and estimation problems, e.g., for human pose estimation based on key point measurements of the joints.

References

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[2] Ijaz Akhter and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2, 4

[3] Hai Ci, Mingdong Wu, Wentao Zhu, Xiaoxuan Ma, Hao Dong, Fangwei Zhong, and Yizhou Wang. GFPose: Learning 3d human pose prior with gradient fields. *arXiv preprint arXiv:2212.08641*, 2022. 1, 5

[4] Andrey Davydov, Anastasia Remizova, Victor Constantin, Sina Honari, Mathieu Salzmann, and Pascal Fua. Adversarial parametric pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 6, 7, 8, 3

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations (ICLR)*, 2017.

[7] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in neural information processing systems (NeurIPS)*, 2019. 1, 5

[8] Luca Falorsi, Pim de Haan, Tim R. Davidson, and Patrick Forré. Reparameterizing distributions on lie groups. In *Proceedings of the twenty-second international conference on artificial intelligence and statistics*, 2019. 2, 6

[9] Nicholas Fisher and A. Lee. Correlation Coefficients for Random Variables on a Unit Sphere or Hypersphere. *Biometrika*, 73, 1986. 1

[10] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked autoencoder for distribution estimation. In *Proceedings of the 32nd international conference on machine learning (ICML)*, 2015. 4

[11] Thorsten Glüsenkamp. Unifying supervised learning and VAEs – coverage, systematics and goodness-of-fit in normalizing-flow based neural network models for astroparticle reconstructions. *arXiv preprint arXiv:2008.05825*, 2023. 2

[12] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems (NeurIPS)*, 2018. 1

[13] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[14] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic Modeling for Human Mesh Recovery. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[15] Hsi-Jian Lee and Zen Chen. Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 1985. 1

[16] Andreas M. Lehrmann, Peter V. Gehler, and Sebastian Nowozin. A Non-parametric Bayesian Network Prior of Human Pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013. 2

[17] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1989. 6

[18] Yulin Liu, Haoran Liu, Yingda Yin, Yang Wang, Baoquan Chen, and He Wang. Delving into Discrete Normalizing Flows on SO(3) Manifold for Probabilistic Rotation Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 4

[19] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael Black. AMASS: Archive of Motion Capture As Surface Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 5, 1, 3

[20] Kieran A. Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-PDF: Non-Parametric Representation of Probability Distributions on the Rotation Manifold. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 3

[21] Haruki Nishimura, Boris Ivanovic, Adrien Gaidon, Marco Pavone, and Mac Schwager. Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction. *CoRR*, abs/2009.05702, 2020. 1

[22] Boris N. Oreshkin, Florent Bocquelet, Félix G. Harvey, Bay Raitt, and Dominic Laflamme. ProtoRes: Proto-residual network for pose authoring via learned inverse kinematics. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 8

[23] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in neural information processing systems (NeurIPS)*, 2017. 2, 4, 5

[24] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 2019. 1, 4

[25] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive Body Capture: 3D Hands, Face, and Body From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 7, 5

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6, 7, 5

[27] Danilo Jimenez Rezende, George Papamakarios, Sebastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. Normalizing flows on tori and

spheres. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 2

[28] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 2000. 1

[29] Tim Salzmann, Marco Pavone, and Markus Ryll. Motron: Multimodal Probabilistic Human Motion Forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[30] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Hierarchical kinematic probability distributions for 3D human shape and pose estimation from images in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[31] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Probabilistic 3D Human Shape and Pose Estimation from Multiple Unconstrained Images in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. HuManiFlow: Ancestor-Conditioned Normalising Flows on SO(3) Manifolds for Human Pose and Shape Distribution Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 6, 7, 8, 4, 5

[33] Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A PyTorch Package for Normalizing Flows. *Journal of Open Source Software*, 2023. 2, 5

[34] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. PoseNDF: Modeling Human Pose Manifolds with Neural Distance Fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 5, 6, 7, 8

[35] Vikram Voleti, Boris N. Oreshkin, Florent Bocquelet, Félix G. Harvey, Louis-Simon Ménard, and Christopher Pal. SMPL-IK: Learned Morphology-Aware Inverse Kinematics for AI Driven Artistic Workflows. *arXiv preprint arXiv:2208.08274*, 2022. 2, 8

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5

[37] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly Supervised 3D Human Pose and Shape Reconstruction with Normalizing Flows. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020. 5

[38] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 4