

Efficient 2D to Full 3D Human Pose Uplifting including Joint Rotations

Katja Ludwig*, Yuliia Oksymets*, Robin Schön, Daniel Kienzle, & Rainer Lienhart
Chair for Machine Learning & Computer Vision, University of Augsburg, Germany

{firstname.lastname}@uni-a.de

Abstract

In sports analytics, accurately capturing both the 3D locations and rotations of body joints is essential for understanding an athlete’s biomechanics. While Human Mesh Recovery (HMR) models can estimate joint rotations, they often exhibit lower accuracy in joint localization compared to 3D Human Pose Estimation (HPE) models. Recent work [18] addressed this limitation by combining a 3D HPE model with inverse kinematics (IK) to estimate both joint locations and rotations. However, IK is computationally expensive. To overcome this, **we propose a novel 2D-to-3D uplifting model that directly estimates 3D human poses, including joint rotations, in a single forward pass.** We investigate multiple rotation representations, loss functions, and training strategies — both with and without access to ground truth rotations. Our models achieve state-of-the-art accuracy in rotation estimation, are 150 times faster than the IK-based approach, and surpass HMR models in joint localization precision.

1. Introduction

Classical monocular 3D Human Pose Estimation (HPE) methods have shown impressive results in recent years. They estimate a 3D human pose consisting of a set of 3D keypoints and a skeleton from either a single image or a video. Most promising methods are 2D to 3D uplifting methods, meaning that they first estimate 2D keypoints in each frame of a video and then lift them to 3D via an upsampling model operating on 2D pose sequences. In sports, a significant limitation of estimated 3D poses from such models is that they do not capture the rotation of body parts. However, rotations are crucial for sports analytics, as they are essential for understanding an athlete’s biomechanics and calculating the forces and torques acting on the body.

In contrast, Human Mesh Recovery (HMR) models estimate 3D human meshes. Most of these models rely on a parametric representation of the human body, such as SMPL-X [21], which explicitly separates body shape and

*Equal contribution

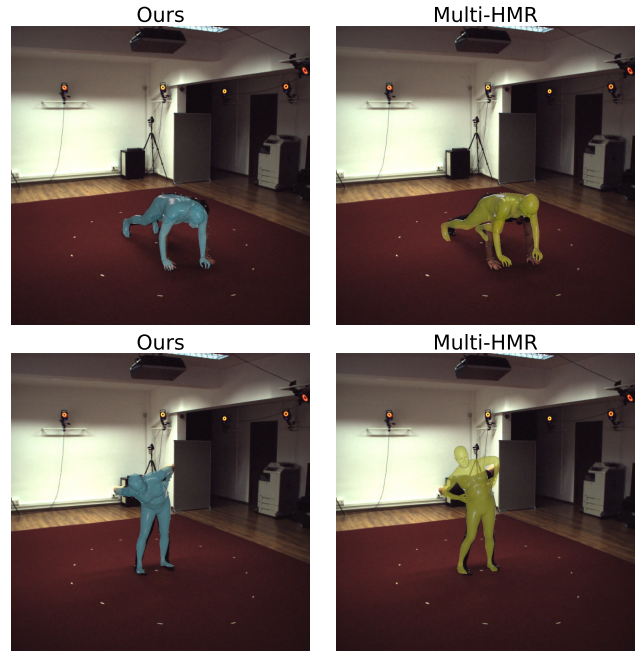


Figure 1. Two examples comparing the results of our model (blue, left column) compared to those of the SOTA HMR model Multi-HMR [2] (yellow, right column). We display meshes created with the estimated rotations and the ground truth body shape.

pose. The body shape is modeled as a low-dimensional embedding, while the body pose is defined by a set of 3D joint rotations. A mesh in a template pose is generated by applying the shape parameters to the parametric model, from which joints are regressed. The estimated joint rotations are subsequently applied to the regressed joints and the corresponding body parts to create the final posed mesh. As a result, HMR models based on parametric body models like SMPL-X can recover full 3D poses, including joint rotations. However, they have a key limitation: They are not able to leverage temporal information from long video sequences. Especially in the field of sports with high-speed movements and extreme poses, this leads to worse detection accuracy regarding keypoint locations compared to 3D HPE models, as investigated by Ludwig et al. [18]. Therefore, they propose to combine a 3D HPE model with a body

shape estimation model and inverse kinematics (IK) to recover a human mesh with more accurate joint locations. They need to apply IK to the 3D poses estimated by the 3D HPE model to obtain the joint rotations, since they are not included in the output of 3D HPE models. IK is a complex and computationally expensive process, since it is an optimization-based approach required to run for each frame.

In this paper, we propose a different approach, which estimates the full 3D pose, including rotations. We extend a recent 2D to 3D uplifting model such that it can estimate a 3D pose including the rotations. Apart from the precision of the estimated joints, we further evaluate the precision of the estimated rotations and show that our model outperforms other SOTA models regarding the accuracy of the estimated rotations. For sports analysts, these rotations are crucial for accurately analyzing an athlete’s biomechanics. Furthermore, our proposed model outperforms Ludwig et al. [18] in speed, as it eliminates the need for an additional IK step. Our contributions can be summarized as follows:¹

- We introduce a novel 2D to 3D uplifting model capable of **estimating 3D human poses, including joint rotations**. We explore multiple rotation representation variants and compare models trained with and without direct supervision on joint rotations.
- A comprehensive evaluation of both joint position and rotation accuracy demonstrates that **our model achieves superior rotation estimation performance** compared to existing SOTA approaches.
- Our proposed models offer a substantial **improvement in computational efficiency** over the method of Ludwig et al. [18] by eliminating the need for an additional IK step, while maintaining comparable joint position accuracy and enhancing rotation estimation.

2. Related Work

2D to 3D Pose Uplifting. To improve 3D joint localization, recent works have leveraged context from neighboring frames in videos. Pavllo et al. [22] propose an uplifting model based on a temporal convolutional network (TCN), which processes long input sequences and models local context by convolving neighboring frames. To model spatial and temporal correlations simultaneously across joints, subsequent works [3, 14] utilize graph convolutional networks. Recently, Transformer-based architectures have become popular for capturing spatio-temporal correlations. PoseFormer [28] stacks a temporal Transformer to learn global dependencies between frames and a spatial Transformer to capture local joint correlations. Li et al. [17] leverage a strided Transformer to efficiently process long input sequences. We select the Uplift and Upsample (UU)

model [8] as the backbone architecture for our models because it is a very efficient SOTA 3D HPE model. It combines spatial, temporal, and strided Transformers. Ludwig et al. [18] combined UU with IK to estimate joint rotations, but their approach is computationally expensive.

Human Mesh Recovery. HMR has been an active area of research in the last years. The parametric SMPL-X [21] body model disentangles the parameter set for pose and shape and has established a stable foundation for HMR. Cai et al. [4] design a Vision Transformer based generalist HMR foundation model using 4.5M training examples from diverse data sources. The challenges caused by smaller features, such as hands and facial expressions, have lead to numerous works that use multi-crop pipelines [6, 9, 20]. Recent approaches extend HMR to multi-person settings, where two-stage pipelines with a human detector and a single-person mesh estimation model dominate [5, 12]. Sun et al. [24] propose a single-shot network with an imaginary Bird’s-Eye-View to efficiently reason about depth in a multi-person setting. Qiu et al. [23] leverage Transformers to capture spatio-temporal context among instances in an end-to-end manner. For comparison with our proposed approaches, we select the SOTA model Multi-HMR [2], which is a single-shot, multi-person Transformer-based network building upon the works of Sun et al. and Qiu et al. [23, 24].

Learning with Rotations. Many computer vision tasks, such as pose estimation from images [7, 27] and point clouds [11] as well as structure from motion, perform regression on rotations [25]. Evaluating the distance between two 3D rotations is often an essential task. Many works use axis-angle vectors or quaternions to represent 3D rotations. In early work, Huynh et al. [15] argue that quaternions are the most efficient representations both spatially and computationally and propose several distance metrics for different representations. Levinson et al. [16] demonstrate that symmetric orthogonalization of rotation matrices via SVD achieves SOTA performance. Zhou et al. [29] discourage using 3D or 4D representations, as they introduce discontinuities in the optimization process.

3. Method

Base Model. All our model variants are based on the Uplift and Upsample (UU) architecture proposed by Einfalt et al. [8]. We briefly recap its architecture, which is visualized in Figure 2. As an input, the UU model takes a sequence of 2D poses $\mathcal{P}^{2D} = p_{t-s_{in} \cdot m}^{2D}, \dots, p_{t+s_{in} \cdot m}^{2D}$ around a central frame p_t^{2D} at time t . Special for the UU model is that this sequence has a stride, hence the poses are not of subsequent frames, but are spaced apart by a fixed number of frames, which is the reason for its efficiency. At first, a spatial Transformer T_{Sp} is applied to each 2D pose separately to enhance the pose-internal representation. This results in a sequence y of enhanced feature tokens per pose. If the output stride

¹The code is available at https://github.com/kaulquappe23/full_3d_hpe_uplifting

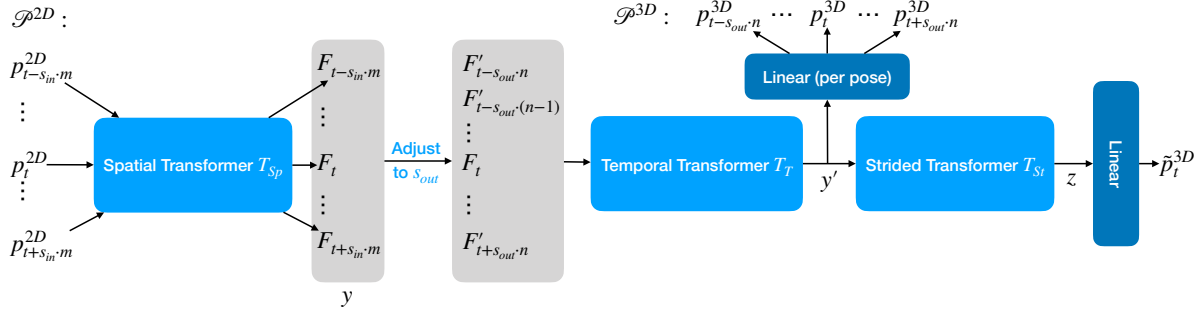


Figure 2. General architecture of the UU model. A pose sequence is fed through an intra-pose operating spatial Transformer T_{Sp} followed by an inter-pose operating temporal Transformer T_T . A linear layer outputs 3D pose estimates for each pose in its input sequence, while a strided Transformer T_{St} with a final linear layer reduces the sequence length to output a single 3D pose estimate for the central frame at position t in the input sequence.

s_{out} is lower than the input stride s_{in} , which is used to efficiently generate a denser output, these tokens are padded with special upsampling tokens for every missing frame in the sequence. Then, they are fed through a temporal Transformer T_T , which operates across the pose tokens. A linear layer is applied to every pose token from the output sequence y' , resulting in an auxiliary output sequence of 3D poses $\mathcal{P}^{3D} = p_{t-s_{out} \cdot n}^{3D}, p_{t-s_{out} \cdot (n-1)}^{3D}, \dots, p_{t+s_{out} \cdot n}^{3D}$ with the output stride s_{out} . Last, y' is fed through a strided Transformer T_{St} , which gradually reduces the sequence length and outputs a single enhanced 3D pose $\tilde{p}_t^{3D}$ for the central frame of the input sequence. The final output is $\tilde{p}_t^{3D}$. A root-relative mean per-joint position error (MPJPE) is used as the loss function L_{joint} for both $\tilde{p}_t^{3D}$ and the auxiliary output sequence $p_{t-s_{out} \cdot n}^{3D}, \dots, p_{t+s_{out} \cdot n}^{3D}$. UU is pre-trained on the large motion capture dataset AMASS [19] and fine-tuned on the target dataset. We adapt this model to estimate the root-relative 3D joint locations and rotations.

Rotation Definition. In this work, we use the same rotations as the SMPL-X body model [21]. It consists of 22 joints with a root joint at the pelvis. Each rotation is defined relative to its parent joint. The rotation of the root joint itself is defined relative to the global coordinate system, hence it defines the global rotation of the body. We further add 2 joints per hand pose to our set of rotations to capture the position of the hands in more detail. We do not want to estimate the rotations for all fingers, since sports analysts are mainly interested in the body pose. Hence, our main set of rotations consists of 26 joints. However, the set of rotations (called body pose in SMPL-X) can be defined differently depending on the application. For some of our methods, arbitrary rotation definitions are possible, but we also experiment with the SMPL-X body model as an intermediate layer, which only allows SMPL-X compatible rotations.

3.1. Rotation Representations and Losses

3D rotations can be represented in various ways. This paper examines three representations: rotation matrices, quater-

nions, and axis-angle forms. Additionally, we investigate different loss functions for learning rotations. We apply them to both the central output $\tilde{p}_t^{3D}$ and the output sequence $\mathcal{P}^{3D}$, where the mean over all sequence elements is used.

Rotation Matrices. Mathematically, 3D rotations in Euclidean space are represented as rotation matrices R in the special orthogonal group $R \in SO(3) \subset \mathbb{R}^{3 \times 3}$. All these matrices are orthogonal and have a determinant of 1. Using rotation matrices as the network output can not be applied directly, since a neural network can not be constrained to directly output valid rotation matrices. We solve this by projecting the network output to the closest valid rotation matrix regarding the Frobenius norm with a Singular Value Decomposition [16].

Axis-Angle Form. The axis-angle representation of a 3D rotation is given by a rotation axis $\omega \in \mathbb{R}^3$ and an angle $\alpha \in \mathbb{R}$, whereby the rotation axis ω is a unit vector. Axis-angle is more compact than rotation matrices, but it faces the problem of double cover [29]. This means that it has two representations for the same rotation, which leads to discontinuities in the representation space: It is possible that the shortest distance between two elements in $SO(3)$ corresponds to a much larger distance in the axis-angle representation space which can hinder gradient-based optimization. Additionally, axis-angle representations can suffer from singularities when the rotation angle approaches zero.

Quaternions. Quaternions extend the concept of complex numbers to higher dimensions and can be defined through four real values as $q = (w, x, y, z) \in \mathbb{R}^4$. The rotation axis is defined by the vector $x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, whereby $\mathbf{i} = (1, 0, 0)$, $\mathbf{j} = (0, 1, 0)$, and $\mathbf{k} = (0, 0, 1)$ are unit vectors. To define the rotation around this axis by an angle α , the quaternion q is defined as

$$q = \cos(\alpha/2) + \sin(\alpha/2)(x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) \quad (1)$$

Despite being a robust, efficient, and numerically stable way to handle rotations in 3D space, quaternions also double cover $SO(3)$ [15]. The quaternions q and $-q$ repre-

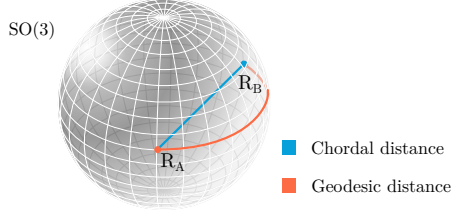


Figure 3. Visualization of the Chordal and Geodesic distances on the $SO(3)$ sphere for two rotation matrices R_A and R_B . [1]

sent the same rotation in $SO(3)$. As mentioned before, this leads to discontinuities in the representation space and can impede gradient-based optimization.

Mean Squared Error (MSE). MSE is a common loss function, which we also use in this work. We apply it to the rotations in the used rotation representation space (rotation matrix, axis-angle or quaternions). For rotation matrices, the MSE loss can be interpreted as the squared Chordal distance [13], which is visualized in Figure 3.

Geodesic Loss. The Geodesic loss is calculated only for rotation matrices R . In case of another selected representation, we convert it to the corresponding rotation matrix first. Essentially, the Geodesic distance represents the minimal angular difference between rotations. It is defined as:

$$L_{geo}(R, \tilde{R}) = \frac{1}{K} \sum_{k=1}^K \arccos \left(\frac{\text{trace}(R_k \tilde{R}_k^T) - 1}{2} \right), \quad (2)$$

where K is the number of joint rotations, and $\tilde{R}$ and R are the predicted and ground truth rotation matrices, respectively [29]. In the case of complete alignment, L_{geo} is zero. Figure 3 provides a visual interpretation of the Geodesic distance and highlights its difference from the Chordal distance. Visually, the Geodesic distance is the shortest path between two points on the surface of the $SO(3)$ sphere.

3.2. Fully Supervised Rotation Estimation

At first, we introduce model variants that are trained with direct supervision on joint rotations. This is only possible if ground truth rotations are available. We explore possibilities without access to the ground truth in the next section.

3.2.1. Naive Approach

In our first approach, we naively extend the UU model to estimate rotations by adding a second head branch for rotation estimation analogous to the joint location estimation. A linear layer is applied to the output of the temporal Transformer T_T to obtain the rotation estimates for the full sequence $\mathcal{R}$. The output of the strided Transformer T_{St} is fed through a linear layer to obtain a refined estimate for the rotations for the central frame $\hat{\theta}_t$. The loss function L_{angle} is applied in addition to L_{joint} to the output sequence $\mathcal{R}$

and the output for the central frame $\hat{\theta}_t$. This approach is visualized in Figure 4. We combine this approach with all three rotation representations and both loss functions in our experiments.

3.2.2. Rotation Estimation with a SMPL-X layer

In the naive approach, joint rotation and location estimations are completely separate. Since they are actually highly correlated, we propose to unify both estimations by using the SMPL-X body model as an intermediate layer. To use it, we further need body shape parameters β . Since it is not the focus of this paper, we do not estimate the body shape. During training, we use the ground truth body shape. For evaluation, we further use the A2B methods presented by Ludwig et al. [18] to obtain a consistent body shape. Note that our methods combined with such estimated β parameters form an HMR method which leverages 2D pose sequences and outperforms the SOTA model Multi-HMR [2] in our experiments (see Section 4.6).

The SMPL-X approach consists of a single head branch for the central frame. Since our goal is improved runtime, we refrain from applying a SMPL-X layer to the output sequence since it slows down training and inference. Hence, the model for the output sequence is identical to the naive approach, while the output for the central frame is generated by passing the output z of the strided Transformer T_{St} to a linear layer to estimate $\hat{\theta}_t$. Next, $\hat{\theta}_t$ is used as an input to the SMPL-X layer to regress the joint locations. The loss functions L_{angle} and L_{joint} are applied to the respective outputs. This approach is visualized in Figure 5.

3.3. Inverse Kinematics (IK)

One option to obtain joint rotations θ and body shape β from 3D joint locations only are Inverse Kinematics (IK). Pavlakos et al. [21] provide an IK algorithm tailored to the SMPL-X body model. However, since IK involves solving an optimization problem for each frame independently, it is computationally intensive. Ludwig et al. [18] utilized IK

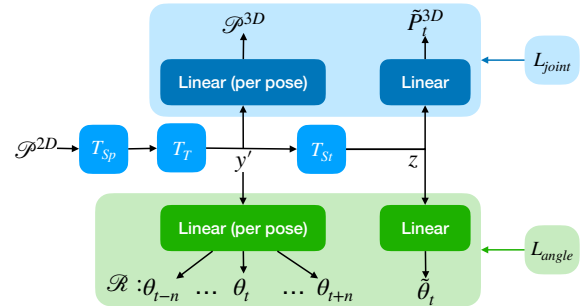


Figure 4. Naive approach to estimate rotations. A second head branch (green) is added to the UU model (blue) to estimate a rotation sequence $\mathcal{R}$ and a refined rotation estimate for the central frame $\hat{\theta}_t$.

on outputs from the original UU model to extract joint rotations. We use their model as a baseline and compare our methods to this approach in our experiments.

Pseudo Label Approach. Additionally, IK can be applied to ground truth joint positions to generate pseudo ground truth joint rotations for datasets lacking SMPL-X annotations. This enables the training of fully supervised models using these pseudo labels.

3.4. Weakly Supervised Rotation Estimation

We further explore alternative models for datasets without access to ground truth rotations. Since the methods still require 3D joint location annotations, we refer to these approaches as weakly supervised.

3.4.1. Rotation Estimation with a SMPL-X layer

This approach is very similar to the supervised SMPL-X approach presented in Section 3.2.2. The main difference is that L_{angle} can not be calculated since the necessary ground truth is not available. Hence, the loss function is only L_{joint} and the central joint rotations are supervised indirectly through the SMPL-X layer. However, there is no supervision of the rotation output sequence $\mathcal{R}$.

Experiments show that despite achieving good joint localization accuracy, the joint rotations of this model deviate significantly from the expected output. Since the rotations of the SMPL-X body model are not limited to the same range as the human body, the model learns to predict unrealistic rotations, although the joint locations are fairly accurate. We visualize this problem in Figure 7. Therefore, we will not include this approach in the experiments.

3.4.2. Rotation Estimation with a Human Body Prior

To address this issue, we include a building block that enforces realistic rotations during training. We choose VPoser, which is a human body prior that has learned plausible poses from the large AMASS [19] dataset. VPoser is an autoencoder that encodes joint rotations θ into a lower-

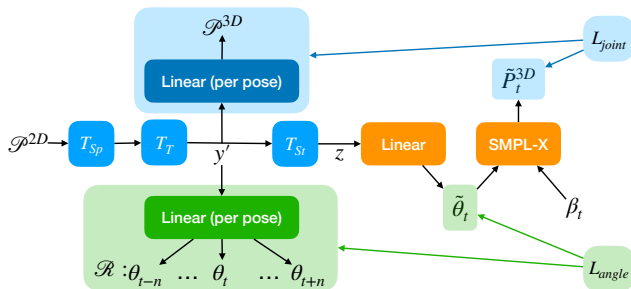


Figure 5. Unified central frame joint rotation and location estimation. The estimated central rotations $\tilde{\theta}_t$ are fed through a SMPL-X layer (orange) to estimate the joint locations. The body shape β_t is either the ground truth or obtained with the methods from [18].

dimensional latent space. The original SMPL-X θ parameters for the body pose (without the hands) in an axis-angle format are described by a 63-dimensional vector (3 values for each of the 22 joints apart from the root joint). In contrast, the VPoser latent space has 32 dimensions, and it is normally distributed. This means that the closer a VPoser latent vector is to zero, the more probable is the pose.

We incorporate VPoser in our model. The output of the strided Transformer T_{St} is fed through a linear layer to estimate the rotations v_t in the latent space of VPoser. VPoser is used to decode v_t to the SMPL-X body pose θ_t . Next, the SMPL-X body model is used as before to regress the joints. The loss function L_{joint} is applied to the regressed joint locations and the output pose sequence $\mathcal{P}^{3D}$. This approach is visualized in Figure 6.

4. Experiments

In this section, we present the experimental results of our proposed model variants. We evaluate **joint location performance** as well as **joint rotation performance**. We compare our model variants to the approach involving IK by Ludwig et al. [18] and the SOTA HMR model Multi-HMR [2]. Moreover, we evaluate the computational efficiency. For all experiments, we initialize the UU model with the pretrained weights from the AMASS dataset as in [8].

4.1. Evaluation Metrics

We evaluate the joint rotation performance with the Mean Per Joint Angular Error (MPJAE) [26]. Let the relative rotation of joint k be defined by the rotation matrix $R_k \in \mathbb{R}^{3 \times 3}$. For the entire set of K joints, it is represented as $R = (R_1, \dots, R_K)^T \in \mathbb{R}^{K \times 3 \times 3}$. The MPJAE measures the geodesic distance between the estimated joint rotations $\tilde{R}$ and the ground truth rotations R .

For each joint k , we define $R'_k = \tilde{R}_k R_k^T$. The matrix R'_k equals the identity matrix $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ if the estimate and ground truth perfectly match. Otherwise, R'_k is the rotation required to align the predicted orientation with the ground truth. To find the angle of this rotation φ , we derive it from the trace of the matrix ($\text{trace}(R'_k) = 1 + 2 \cos \varphi$), using the arc-cosine. This metric reports the error in radians, but we convert it to degrees for easier interpretation in our evaluation tables.

$$\text{MPJAE}(R, \tilde{R}) = \frac{1}{K} \sum_{k=1}^K \arccos \left(\frac{\text{tr}(R'_k) - 1}{2} \right) \quad (3)$$

Moreover, the joint location performance is evaluated using the very common root-relative Mean Per Joint Position Error (MPJPE).

4.2. Dataset

We evaluate our model variants on the **fit3D dataset** [10], as it is the only publicly available sports dataset with SMPL-

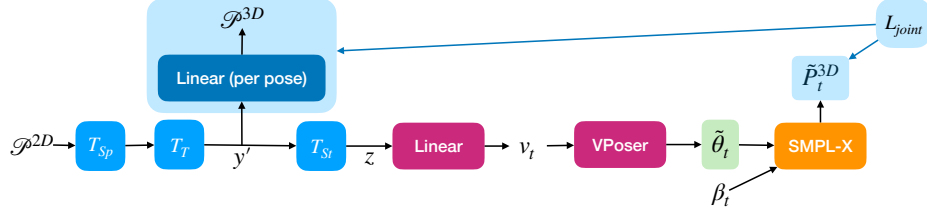


Figure 6. Weakly supervised rotation estimation with the VPoser body prior. The rotation estimation is done in the latent space of VPoser (purple). The estimated rotations are decoded to the joint rotations θ_t and used to regress the joint locations. Rotations are only estimated for the central frame and indirectly supervised via L_{joint} .

X annotations. The fit3D dataset comprises videos of human subjects performing various fitness exercises, specifically designed for studying repetitive human motion in a fitness context. The dataset includes recordings of eleven individuals captured in a controlled studio environment from multiple viewpoints, with only one person visible per video. The recording setup utilized four synchronized RGB cameras along with a VICON motion capture system consisting of twelve motion cameras. Additionally, each subject was 3D scanned, which is important for gathering the body shape coupled with the body pose. The recorded exercises target different muscle groups, covering a total of 47 exercises performed by either certified fitness instructors or trainees with varying skill levels.

Fit3D provides official training and test splits. Since the ground truth data is needed for our evaluation to obtain the body shape, and it is only available for the training subset, we do not use the provided test subset in our work. Instead, we divide the original training subset as follows: 6 subjects (s03, s04, s05, s07, s08, s10) are used for training, and 1 subject each for validation (s09) and test (s11). Since sports analysts focus mainly the body and not the hands/face, we select a subset of joints that we use for our trainings and evaluations. We choose the 22 main body joints and 2 joints on each hand (thumb and pinky). VPoser is only trained on the main body pose, therefore we evaluate both on the 22 main body joints and our 26 joints.

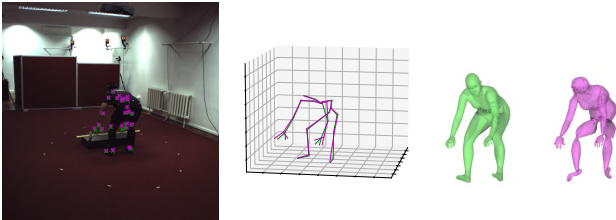


Figure 7. Example prediction for rotation estimation with a SMPL-X layer in weakly supervised manner. The ground truth is displayed in green, the prediction in pink. Joint locations are shown in the first two images, the resulting meshes afterwards. The predicted rotations are completely twisted, resulting in an impossible mesh.

4.3. Within-Batch Augmentation

We use horizontal flipping to augment paired image and 3D pose data in our trainings. We use it in the form of within-batch augmentation (WBA), since it has shown promising results for 2D to 3D uplifting methods, including UU [8]. WBA means that half of each batch is made up of the original pose sequence and the other half of horizontally flipped pose sequences.

4.4. Fully Supervised Training Results

Uplift Upsample. For comparison, we provide the results of the original UU model. Hence, it is only trained on the 3D joint locations and can not estimate joint rotations. Results are provided in Table 1, model UU.

Naive Approach. We combine our naive model as explained in Section 3.2.1 with all rotation representations and all loss functions. Results are presented in Table 1. We mark all models based on the naive approach with a leading *N*- and name them according to their rotation representa-

Model	L_{angle}	WBA	MPJPE ↓	MPJAE ↓
UU-1	-	-	35.51	-
UU-2	-	✓	34.68	-
N-AA-1	MSE	-	62.52	10.29
N-AA-2	Geodesic	-	<u>48.35</u>	9.39
N-AA-3	MSE	✓	86.18	9.38
N-AA-4	Geodesic	✓	49.69	<u>9.28</u>
N-Q-1	MSE	-	<u>78.51</u>	<u>10.60</u>
N-Q-2	Geodesic	-	563.84	85.35
N-Q-3	MSE	✓	95.42	11.17
N-Q-4	Geodesic	✓	531.82	88.40
N-RM-1	MSE	-	45.66	9.33
N-RM-2	Geodesic	-	39.40	8.82
N-RM-3	MSE	✓	45.05	9.21
N-RM-4	Geodesic	✓	41.11	8.84

Table 1. Results for fully supervised training with UU and the naive method (denoted with *N*-) with all three rotation representations (axis-angle *AA*, quaternions *Q*, and rotation matrices *RM*) and both different loss functions. MPJPE results are given in mm, MPJAE results in degrees. The best results for each model are underlined, the best overall results are marked in bold.

Model	L_{angle}	WBA	MPJPE	MPJAE
S-AA-1	MSE	-	62.48	9.14
S-AA-2	Geodesic	-	54.03	9.72
S-AA-3	MSE	✓	36.69	9.21
S-AA-4	Geodesic	✓	41.31	9.23
S-RM-1	MSE	-	42.90	9.42
S-RM-2	Geodesic	-	41.90	9.44
S-RM-3	MSE	✓	42.69	9.22
S-RM-4	Geodesic	✓	<u>40.21</u>	<u>9.19</u>

Table 2. Results for fully supervised training with the SMPL-X layer method (denoted with *S*-) for axis-angle (*AA*) and rotation matrix (*RM*) rotation representations and both loss functions. MPJPE results are given in mm, MPJAE results in degrees. The best results for each model are underlined, the best overall results for model variants with rotation estimation are marked in bold.

tion *AA* for axis-angle, *Q* for quaternions, and *RM* for rotation matrices. We observe very different results for the different representations. Quaternions perform badly, while axis-angle and rotation matrices show promising results, especially with geodesic loss, while quaternions perform better with MSE loss. Interestingly, WBA fails to improve the naive model’s results, unlike observed with the UU model. The best results are achieved with rotation matrices and geodesic loss.

SMPL-X Layer Approach. Next, we evaluate the approach including an SMPL-X layer (see Section 3.2.2). We combine it with both loss functions and rotation matrices and axis-angle rotation representations. We leave out quaternions since they perform so badly in the naive approach and since they are not used in the SMPL-X model itself as representations. Results are presented in Table 2 and marked with a leading *S*-. The results differ from the naive approach. Axis-angle now outperform rotation matrices. The best MPJAE result is achieved with MSE and without WBA (S-AA-1), but the MPJPE of this experiment is relatively bad, over 28 mm higher than the original UU model. However, with WBA (S-AA-3), the MPJAE rises a little, but the MPJPE is reduced to 36.69 mm, which is only 2 mm higher than the original UU model and nearly 3 mm lower than the best naive model. Rotation matrices achieve similar MPJAE scores, but the best MPJPE score is over 3 mm higher than the best axis-angle model. Qualitative results of the overall best supervised model S-AA-3 are shown in Figure 1.

4.5. Weakly Supervised Training Results

Pseudo Label Approach. We run IK on the ground truth 3D joint locations to obtain pseudo labels for the joint rotations. This way, we do not use the ground truth joint rotations but can use the fully supervised training routines (see Section 3.3, pseudo label approach). We use the IK routine provided by Pavlakos et al. [21], which estimates only the

Model	WBA	MPJPE	MPJAE-22
PS-AA-1	-	38.05	16.20
PS-AA-3	✓	37.30	<u>16.18</u>
V-1	-	40.40	16.11
V-2	✓	<u>38.76</u>	15.90

Table 3. Results for models with joint location annotations only (MPJPE in mm, MPJAE in degrees). The SMPL-X layer method with pseudo labels is marked with leading *PS-AA*-, the VPoser approaches with *V*-. Best results for each model are underlined, the best overall results are bold.

main body pose and not the hand pose. Therefore, we also evaluate only on the 22 joints of the main body pose. We show the result of this experiment with the same two settings as the best supervised models (S-AA-1 and S-AA-3) in Table 3, model *PS-AA-1/3*.

VPoser Approach. We use VPoser as a human body prior to prevent the weakly supervised model from estimating unrealistic poses (see Section 3.4.2). Results of this experiment are shown in Table 3, with models marked by a leading *V*-. VPoser is trained only on the main body pose, so we evaluate on the 22 main body joints. Results for the 26-joint set for the best models are presented in Table 4.

The experiments show that the MPJAE is generally much higher than with direct supervision, which is something to be expected. The best weakly supervised model regarding the MPJAE is the VPoser model V-2. Compared to the best supervised model, the MPJPE scores for the weakly supervised models are only slightly higher and even better than the best naive model, most likely because joint locations are supervised with ground truth labels. The MPJPE results are best for the pseudo label approach PS-AA-3. Since V-2 achieves a little better MPJAE and PS-AA-3 a little better MPJPE, there is not a single best model in the weakly supervised case. However, the best results are achieved with WBA in both cases.

4.6. Comparison with other SOTA Methods and Runtime Evaluation

After evaluating our own model variants and identifying the best versions, we compare them with other SOTA methods. On the one hand, we choose Multi-HMR [2], which is a SOTA model for HMR and showed the best performance on fit3D [18]. It operates image-wise and estimates SMPL-X human meshes. In our evaluations, we keep the estimated rotations and combine them with the ground truth body shape to achieve a fair evaluation. On the other hand, we choose the pipeline involving UU and IK (called UU-IK) proposed by Ludwig et al. [18]. At first, we also use the ground truth body shape for this model. We evaluate with estimated body shapes in Section 4.7. Results are displayed in Table 4. We include the MPJAE on our full set

Model	GT rot.	MPJPE-26	MPJAE-26	MPJAE-22	MPJPE-37	Runtime [ms]
UU-2	-	34.68	-	-	34.3 [18]	7.11 ± 2.23
N-RM-2	✓	39.40	8.82	8.79	42.38	9.86 ± 2.78
S-AA-3	✓	36.69	9.21	9.24	41.14	11.09 ± 3.14
PS-AA-3	-	37.30	(18.09)	16.18	42.33	11.46 ± 3.50
V-2	-	38.76	(17.86)	15.90	44.43	16.08 ± 4.24
Multi-HMR [2]	(✓)	62.22	18.49	17.59	68.96	156.10 ± 10.09
UU-IK [18]	-	34.62	(18.30)	16.43	36.91	$1952.19 \pm 1091.52 /$ 4733.31 ± 1826.99

Table 4. Results of our best models and other SOTA models based on ground truth body shape (MPJPE in mm, MPJAE in degrees). The number of joints involved in the metric calculation is denoted after the respective metric name. The original UU model is displayed only for comparison and not included in highlighting best models since it is not capable of estimating rotations. We highlight best results (MPJPE, MPJAE and runtime) for models trained with and without ground truth rotations (GT rot.) available during training. Some models do not provide rotations for fingers. In these cases, we set them to 0 and put MPJAE-26 results in brackets. Since Multi-HMR is trained with rotation supervision, but not on the fit3D dataset, we put the checkmark in brackets. The runtime is measured in ms. For UU-IK, we provide the runtime for frames with (first value) and without (second value) pre-initialization.

of 26 joint rotations, also for the models which do not estimate the hand rotations (we leave them in the template pose in these cases). For comparison, we include the MPJPE on the same set of 37 keypoints as selected by Ludwig et al. [18] (our set contains 26 keypoints). The reader should keep in mind that these evaluations are biased towards their model, since scores are improving for keypoints included in the training target and nearly a third of them are not included in our losses. Moreover, we provide a runtime evaluation. We measure the time needed for a forward pass on a single image using a NVIDIA GeForce RTX 2080 Ti GPU. We provide the mean and standard deviation based on at least 5k forward passes. For UU-IK we measure the runtime separately for frames initialized with the pose from the previous frame and without pre-initialization (see [18]).

Our evaluations show that **our best fully supervised models outperform all other models regarding MPJAE scores**. Multi-HMR performs worst regarding MPJPE and MPJAE scores. UU-IK achieves the best MPJPE scores, but the MPJAE scores are worse compared to our models. Its significant problem is further the runtime. While having an average runtime of over 1900 ms even for frames with pre-initialization, our best models only need 9.86 ms and 11.48 ms (best fully and weakly supervised model, respectively). This is over **150 times faster**, which proves that we achieve our goal of providing a faster and comparably accurate 2D to 3D pose uplifting model including joint rotations.

4.7. Evaluation with Consistent Body Shape

Until now, we evaluated with the ground truth body shape. Lastly, we evaluate the best models with an estimated body shape. We use the A2B models suggested by Ludwig et al. [18], which estimate the body shape based on anthropometric measurements. This makes sense in sports, since professional athletes are measured for their analyses. Moreover,

Model	GT rot.	MPJPE-26	MPJPE-37
N-RM-2	✓	41.01	43.95
S-AA-3	✓	38.35	42.72
PS-AA-3	-	39.02	43.98
V-2	-	40.39	46.01
UU-IK [18]	-	35.71	38.41

Table 5. MPJPE results (for 26 and 37 keypoints) of our best models and other SOTA models based on A2B body shape [18] in mm. We only display the result of the best of the four A2B model variants. For MPJAE, see Table 4.

we use a single set of estimated body shape parameters to ensure a consistent body shape across all frames of a video, as suggested by [18]. Since we used the ground truth body shape so far, the evaluations might not seem completely realistic. However, the fit3D ground truth itself is not consistent as shown by [18], which makes a completely fair evaluation with consistent body shapes impossible. Results are shown in Table 5. Only MPJPE values are included since MPJAE is not affected by the body shape.

5. Conclusion

In this paper, we have proposed several novel model variants for **efficiently estimating a full 3D human pose, including joint rotations, based on a 2D pose sequence**. We explored models trained with and without ground truth rotations. We have shown that our models outperform the state-of-the-art in terms of rotation estimation accuracy. Furthermore, we have demonstrated that our models are computationally more efficient than the method of Ludwig et al. [18] by eliminating the need for an additional inverse kinematics step. Our models are particularly well-suited for sports analytics, as they provide accurate joint rotations, which are crucial for understanding an athlete’s biomechanics.

References

- [1] Olaya Alvarez-Tunon, Yury Brodskiy, and Erdal Kayacan. Loss it right: Euclidean and riemannian metrics in learning-based visual odometry. In *ISR Europe 2023; 56th International Symposium on Robotics*, pages 107–111. VDE, 2023. 4
- [2] Fabien Baradel, Matthieu Armando, Salma Galaoui, Romain Brégier, Philippe Weinzaepfel, Grégory Rogez, and Thomas Lucas. Multi-hmr: Multi-person whole-body human mesh recovery in a single shot. *arXiv preprint arXiv:2402.14654*, 2024. 1, 2, 4, 5, 7, 8
- [3] Yujun Cai, Lihao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2
- [4] Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Wang Yanzun, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, et al. Smpler-x: Scaling up expressive human pose and shape estimation. *Advances in Neural Information Processing Systems*, 36:11454–11468, 2023. 2
- [5] Hongsuk Choi, Gyeongsik Moon, JoonKyu Park, and Kyoung Mu Lee. Learning to estimate robust 3d human mesh from in-the-wild crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1475–1484, 2022. 2
- [6] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 20–40. Springer, 2020. 2
- [7] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018. 2
- [8] Moritz Einfalt, Katja Ludwig, and Rainer Lienhart. Uplift and upsample: Efficient 3d human pose estimation with up-lifting transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2903–2913, 2023. 2, 5, 6
- [9] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Collaborative regression of expressive bodies using moderation. In *2021 International Conference on 3D Vision (3DV)*, pages 792–804. IEEE, 2021. 2
- [10] Mihai Fieraru, Mihai Zanfir, Silviu-Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. Afit: Automatic 3d human-interpretable feedback models for fitness training. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5
- [11] Ge Gao, Mikko Lauri, Jianwei Zhang, and Simone Frin-trop. Occlusion resistant object rotation regression from point cloud segments. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2
- [12] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023. 2
- [13] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103:267–305, 2013. 4
- [14] 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*, pages 602–611, 2021. 2
- [15] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35:155–164, 2009. 2, 3
- [16] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33: 22554–22565, 2020. 2, 3
- [17] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 25:1282–1293, 2022. 2
- [18] Katja Ludwig, Julian Lorenz, Daniel Kienzle, Tuan Bui, and Rainer Lienhart. Leveraging anthropometric measurements to improve human mesh estimation and ensure consistent body shapes. *arXiv preprint arXiv:2409.17671*, 2024. 1, 2, 4, 5, 7, 8
- [19] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5442–5451, 2019. 3, 5
- [20] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3d hand pose estimation for whole-body 3d human mesh estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2308–2317, 2022. 2
- [21] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 1, 2, 3, 4, 7
- [22] Dario Pavlo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [23] Zhongwei Qiu, Qiansheng Yang, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Chang Xu, Dongmei Fu, and Jingdong Wang. Psvt: End-to-end multi-person 3d pose and shape estimation with progressive video transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21254–21263, 2023. 2

- [24] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13243–13252, 2022. [2](#)
- [25] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [26] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018. [5](#)
- [27] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. [2](#)
- [28] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, 2021. [2](#)
- [29] 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)*, pages 5745–5753, 2019. [2](#), [3](#), [4](#)