Towards Stable Human Pose Estimation via Cross-View Fusion and Foot Stabilization

Li'an Zhuo*, Jian Cao*, Qi Wang, Bang Zhang, Liefeng Bo
Alibaba Group
{lianzhuo.zla, tanfeng.cj, wilson.wq, zhangbang.zb, liefeng.bo}@alibaba-inc.com

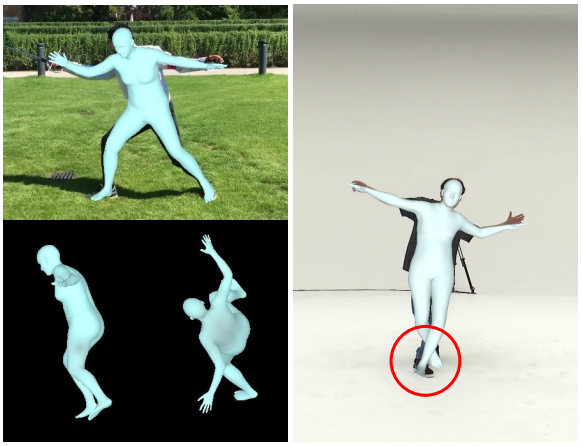
Abstract

Towards stable human pose estimation from monocular images, there remain two main dilemmas. On the one hand, the different perspectives, i.e., front view, side view, and top view, appear the inconsistent performances due to the depth ambiguity. On the other hand, foot posture plays a significant role in complicated human pose estimation, i.e.*, dance and sports, and foot-ground interaction, but unfortunately, it is omitted in most general approaches and datasets. In this paper, we first propose the Cross-View Fusion (CVF) module to catch up with better 3D intermediate representation and alleviate the view inconsistency based on the vision transformer encoder. Then the optimization-based method is introduced to reconstruct the foot pose and foot-ground contact for the general multi-view datasets including AIST++ and Human3.6M. Besides, the reversible kinematic topology strategy is innovated to utilize the contact information into the full-body with foot pose regressor. Extensive experiments on the popular benchmarks demonstrate that our method outperforms the state-of-the-art approaches by achieving* 40.1*mm PA-MPJPE on the 3DPW test set and* 43.8*mm on the AIST++ test set.*

1. Introduction

Estimating 3D poses from a monocular RGB camera is significant in computer vision and artificial intelligence, as it is fundamental in many applications, *e.g*. robotics, action recognition, animation, human-object interaction, *etc*. Benefiting from the dense representation of SMPL models [18], SMPL-based methods [9–12] have recently dominated the 3D pose estimation and achieved state-of-the-art results. Although these methods have considerably decreased the reconstruction error, they still suffer from two main challenges in pose stability. Thus, in this paper, we focus on SMPL-based 3D pose estimation and present a method for reducing the instability in estimation.

* indicates the equal contributions.



(a) Inconsistent performance from different perspectives. (b) Inaccurate foot posture and foot-ground interaction.

Figure 1. Two main challenges towards stable human pose estimation.

The first challenge is the inconsistency performance of poses from different perspectives. An example is shown in Figure 1a that the front view projection of the 3D poses predicted by the model can be well aligned with the picture, but from its side view, the human poses are oblique. The difficulty mainly stems from the fact that estimating 3D human poses requires a model to extract good 3D intermediate representation from monocular images, which is difficult due to the lack of depth input. The second challenge is the stability of the foot posture. As shown in Figure 1b, the estimated foot posture is inaccurate and does not match the foot-ground contact. The main reason is that the contact between the foot and the ground and the posture of foot joints *i.e*. heels, foot toes, ankles, *etc*. are omitted in most work.

In the literature, most SMPL-based methods directly extract the holistic features from the image and then feed them to the subsequent regression networks to calculate the SMPL parameters [9–12]. These holistic methods do not explicitly model the pose-related 3D features. In addition, it is also challenging to directly predict the SMPL param-

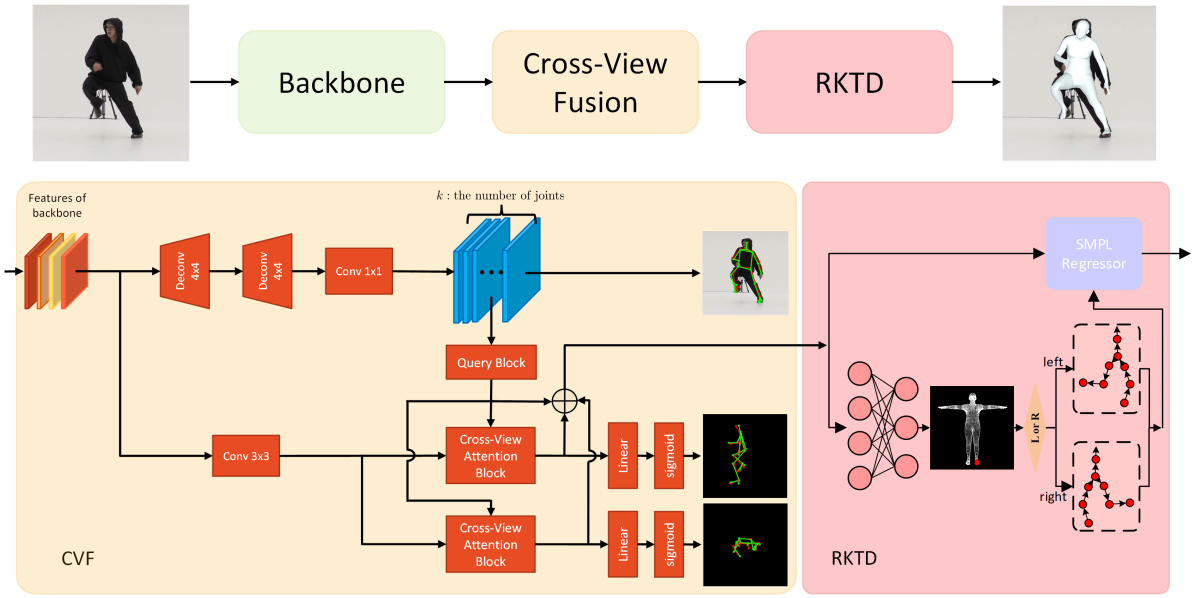


Figure 2. The top-down framework for 3D human pose and shape estimation, which consists of three parts, including the vision transformer encoder, the cross-view attention representation, and the reversible kinematic topology decoder.

eters from the holistic features due to the highly nonlinear mapping [22]. Our first contribution is that we propose an intermediate representation architecture called the Cross-View Fusion module (CVF). It learns a fused 3D intermediate representation by supervision over three views: the front, the side, and the top. Specifically, our method consists of three branches. Each branch learns 2D poses and features in its corresponding view. Predicting the 2D poses in side-view and top-view from input images is challenging, so we design an attention-based architecture that leverages prior information from the front-view branch to facilitate the training of side-view and bird-view branches. Thanks to the better 3D intermediate representation, our method alleviates the view inconsistency and outperforms other SMPL-based methods on 3D pose estimation.

Understanding the foot-ground contact and learning the inherent dynamic dependencies among joints is the key to solving the challenge of foot stability. However, most datasets lack the annotation information of foot-ground contact. For this reason, we propose a method based on multi-view optimization to add foot-ground annotations to some public datasets, *i.e.*, Human3.6M [7] and AIST++ [16]. Different from the previous optimization-based methods (*e.g.*, SMPLify [1]), our method utilizes multi-view images, which can deal with the severe joints occlusion, and thus obtain better foot joint annotations and foot-ground contact annotations. To the best of our knowledge, our work is the first to perform unified foot-ground contact annotations on multiple existing large-scale 3D pose datasets. We believe these additional annotations will further improve the human pose estimation task in the future.

Inspired by [32], we further propose a Reversible Kinematic Topology Decoder (RKTD) that can dynamically adjust the predicted order of individual lower limb joints according to the state of foot-ground contact.

Our method achieves state-of-the-art performance on multiple 3D human pose estimation benchmarks. On the 3DPW [31] dataset, it achieves 2.7mm improvement compared to the best art D&D [14]. Although our method is trained on single-frame images, it does achieve better results than existing video-based methods, such as MAED [33] and D&D [14]. We annotated foot joint and foot-ground contact on Human3.6M and AIST++ and then trained our method on them. Our method reduced MPJPE by 2mm and 3mm on Human3.6m and AIST++, respectively.

In summary, we make the following four contributions:

- We design a 3D intermediate feature representation module called Cross-View Fusion to extract the features of the key points in the front, side, and bird's eye views. By doing this, our method achieves more consistent performances in different perspectives than other state-of-the-art methods.
- We design an optimization-based scheme to reconstruct the foot poses and annotate foot-ground contacts for the commonly-used multi-view datasets, including AIST++ and Human3.6M. These new annotations can be used to improve pose stability during the foot-ground interaction in future work.
- We propose a Reversible Kinematic Topology Decoder(RKTD) that utilizes the foot-ground contact in-

formation to dynamically adapt the prediction order of the joints on the leg limb chain. This strategy improves the accuracy of pose estimation when there is a foot touchdown.

- We conduct extensive experiments on the commonly-used benchmarks, including 3DPW, Human3.6M, and AIST++. Compared to other existing methods, our method achieves state-of-the-art performance quantitatively. The qualitative comparison shows that our method estimates more stable poses, *i.e*., the performances are more consistent under different views with more accurate foot-ground contacts.

2. Related Work

2.1. 3D Pose and Shape Estimation

The existing 3D human pose and shape estimation methods widely adopt the parametric 3D mesh models, such as SMPL [18]. HMR [9] is the first SMPL-based method that uses a CNN-based backbone to extract image features and then regress the pose and shape coefficients of SMPL. Subsequently, some works follow the HMR's framework and carry forward this framework, such as SPIN and VIBE *etc*. [2, 10–12, 28, 32]. Nikos *et al*. [12] proposes a deep network for 3D human pose and shape estimation through a tight collaboration between a regression-based and an iterative optimization-based approach. VIBE [10] is a video-based method aiming at estimating SMPL parameters for each video frame by a temporal generation network, which is trained together with a motion discriminator. Kocabas *et al*. [11] propose a part attention regressor, which adds a 2D part prediction branch and designs a part attention module to fuse part features and image features, to increase robustness to occlusion. Recently, Li *et al*. propose a video-based method to estimate the dynamic camera, which achieves state-of-the-art results [14]. Recently, ViT [3] has achieved great success in computer vision tasks, and Xu*et al*. propose ViTPose [34], which uses vision transformer as the backbone and achieves state-of-the-art results on 2D pose estimation tasks. Therefore, we use ViTPose as our backbone network to build our proposed method (Section 3.3). The above methods commonly ignore the importance of the overall stability of pose estimation.

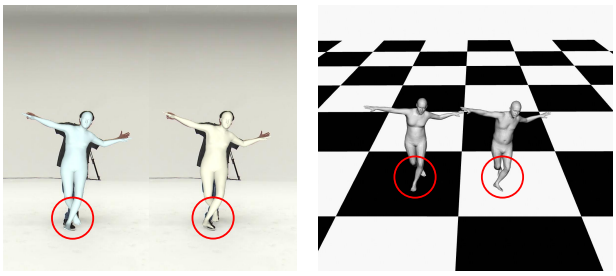
2.2. 3D Intermediate Feature Representation

Learning a good 3D feature representation from the monocular images is non-trivial and helpful for alleviating view inconsistency. In early SMPL-based methods [9, 10, 12], a CNN backbone, such as ResNet, is used to extract the holistic feature from an image. Wan *et al*. [33] propose a spatial-temporal attention encoder to learn the spatial and temporal features simultaneously. These methods do not explicitly learn the 3D feature representation and thus suffer from the nonlinear mapping between the input 2D image and the 3D parameters of SMPL. Using 2D information as an intermediate supervision is a common approach to alleviate the gap between image and 3D coefficients of SMPL [21, 23, 35]. Pavlakos *et al*. [23] extract intermediate 2D keypoints and silhouette features to alleviate this problem. Omran *et al*. [21, 35] use semantic body part segmentation as an intermediate representation to reduce the interference of environment and clothing information in images and improve 3D inference. However, the intermediate features they extracted, whether 2D keypoints or segmentation, are still 2D representations. For building better representations in the network, Sun *et al*. [29] propose to simultaneously infer the center positions in the front view and the center depths in the "bird's eye view" (BEV) of multiple people, which alleviates the ambiguity in monocular depth. Jin *et al*. [8] decouple the problem into 2D pose regression and depth regression. They design a 2D Pose-guided Depth Query Module to enhance the depth prediction with 2D pose features in a model-free framework. Li *et al*. [15] propose HybrIK which predicts the 3D coordinates and partial coefficients of SMPL (*i.e*., shape parameters and twist angle of joints) and feeds them into an inverse kinematic module to solve the pose parameters. Different from supervising 3D coordinates at an interval stage, our proposed CVF module predicts 2D coordinates for three views, whose features are extracted by three branches respectively. This simple decomposition strategy (Section 3.4) allows information exchange among features of each view, which finally improves learning view-specific features (Section 4.3).

2.3. Foot-Ground Contact

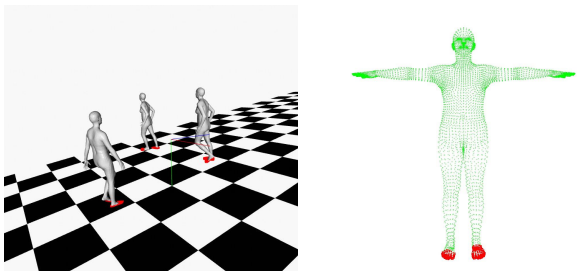
The foot-ground contact labels are an important reference in the Inverse Kinematics solution for eliminating foot sliding when driving avatar [5, 26, 27]. The contact information between the human and the ground can also be used as prior knowledge to help improve the accuracy of human posture estimation. For instance, Ugrinovic *et al*. [30] assume that all people are standing on the same ground plane, and uses this prior to eliminate ambiguities in the body scale and the relative camera-body translation. HuMoR [24] predicts foot-ground contacts to constrain pose estimation at test time. PhysCap [25] exploits foot-ground contact signals and introduces a real-time physics-based pose optimizer that considers environmental constraints, gravity, and biophysical plausibility of human poses. Mourot *et al*. [20] propose a method for foot contact detection and ground reaction force estimation. Zou *et al*. [36] propose an end-to-end method to combine foot-ground contact estimation with pose estimation by directly exploiting a zero-velocity constraint on the foot joints.

The Kinematic Topology Decoder (KTD) is introduced in the previous article [33], which takes into account the



(a) Left: original annotations, right: revised results. (b) Left: original annotations, right: revised results.

Figure 3. Qualitative results of the foot pose annotator.



(a) The predicted ground. (b) The contact annotations.

Figure 4. Qualitative results of the contact annotator.

kinematics tree of the body when estimating the pose of the body joints. In this paper, we extend KTD to include the foot part and propose a reversible version of KTD called Reversible Kinematic Topology Decoder (Section 3.5). Our method can dynamically adjust the prediction order of the lower limb joints according to the foot-ground contact.

3. Method

In this Section, we first briefly review the general parametric human body model SMPL used in our method. Then we introduce the pseudo-GT annotators of the foot poses and foot-ground contacts for the open-source multi-view datasets by the multi-view optimization. Finally, we represent our top-down method for 3D human pose and shape estimation, which consists of three parts, including the vision transformer encoder, the cross-view fusion module, and the reversible kinematic topology decoder (Figure 2).

3.1. SMPL Model

The parametric human body model SMPL takes the body poses $\theta \in \mathbb{R}^{24\times3}$ and the human shape $\beta \in \mathbb{R}^{10}$ as input, and outputs the 3D human body mesh with the vertices $v \in \mathbb{R}^{6890*3}$. The body poses θ consists of the global rotation of the root joint, *i.e.*, pelvis, and the 23 local rotations of the corresponding joints relative to their parents along the kinematic tree. The k 3D joints can be calculated by the linear combination of the vertices as $j_{3d} = Mv$, where $M \in \mathbb{R}^{k\times6890}$.

3.2. Foot Pose and Contact Reconstruction

Foot Poses. Foot poses play a significant role in human pose estimation and foot-ground contact prediction, especially in complicated situations (*e.g.*, dance and sports). However, foot poses are not annotated in most popular datasets, as shown in Figure 1b. Consequently, the general evaluation has not considered the foot keypoints. To address the above problem, we introduce the SMPLify-based [1] framework to recover the foot poses by multi-view optimization.

We first extract the 2D keypoints for each view separately using the 2D pose estimator. Then the original SMPL parameters are optimized under the SMPLify-based framework with multi-view 2D reprojection constraints. As shown in Figure 3, we make use of the close relationship between the foot and the lower leg by jointly optimizing them in the recovery procedure, which benefits a more natural and compatible full-body mesh reconstruction and a better foot pose estimation. The complete optimization is formulated as follows:

$$\underset{\theta_{\text{lower legs}},\theta_{\text{feet}}}{\arg\min}\ \mathcal{L}_{\text{proj}}(\theta,\beta,T,j_{2D}) + \mathcal{L}_{\text{smooth}}(\theta), \tag{1}$$

where θ and β are SMPL parameters that denote the joints' rotation and human shape separately. T is the camera extrinsic matrix. j_{2D} represents the 2D keypoints obtained by the 2D pose estimator. $\mathcal{L}_{\text{proj}}$ is the projection loss function that penalizes the misalignment between the re-projected 3D joints from SMPL and the annotated 2D keypoints.

$$\mathcal{L}_{\text{proj}}(\theta,\beta,T,j_D) = \frac{1}{n}\sum_{i\in\Omega}^{n}(\pi(j_{3D_i},T) - j_{2D_i})^2, \tag{2}$$

where Ω denotes the set of all keypoints, n is the total number of keypoints, j_{3D} represent the 3D keypoints obtained by SMPL model with θ and β as input, and π represents the projection transformation.

$\mathcal{L}_{\text{smooth}}$ is the sequentially-smooth loss function as:

$$\mathcal{L}_{\text{smooth}}(\theta) = \theta_{[1:t-1]} - \frac{1}{3}(\theta_{[0:t-2]} + \theta_{[1:t-1]} + \theta_{[2:t]}), \tag{3}$$

where $\theta_{[a:b]}$ denotes that the temporal poses within ath frame to bth frame, t is the total length of the frames.

Foot-ground Contacts. we propose a method to further obtain the foot-ground contact annotations for the popular indoor datasets having flat ground. Given a sequence of SMPL meshes from a video as input, we first set an initial plane below all the meshes (*e.g.*, $y = -10$), and then compute the closest vertex to this plane in each mesh. A more accurate plane is then estimated by the least square method using these closest vertices. The above procedure

is repeated several times to estimate the object plane function as shown in Figure 4. Consequently, the foot-ground contact annotations are obtained by:

$$GC(v) = \begin{cases} \text{True} & \text{if } D(v, plane) < \delta, \\ \text{False} & \text{otherwise}, \end{cases} \tag{4}$$

where $v \in \mathbb{R}^{6890\times 3}$ is the human body vertices from SMPL. D represents the Euclidean distance from the point to the plane. The vertices with vertex-plane distance less than the height threshold δ are labeled as in contact, *i.e.*, 0.025m in our method.

3.3. Vision Transformer Encoder

In order to extract more powerful features, we use ViT [3] as our backbone. Following the ViTPose [34], the input 2D cropped image $X \in \mathbb{R}^{H\times W\times 3}$ is firstly embedded into a sequence of flattened 2D patches via the patch embedding layer $F \in \mathbb{R}^{\frac{H}{d}\times\frac{W}{d}\times C}$, where H, W and C represent the height, width and the channel dimension, and d denotes the downsampling ratio of the patch embedding layer. After that, the embedded patches are processed by several transformer layers, each consisting of a multi-head self-attention (MHSA) layer and a feed-forward network (FFN),

$$\begin{aligned} F'_{i+1} &= F_i + \text{MHSA}(\text{LN}(F_i)) \\ F_{i+1} &= F'_{i+1} + \text{FFN}(\text{LN}(F'_{i+1})), \end{aligned} \tag{5}$$

where LN denotes the layer normalization. F_i represents the output of the ith transformer layer and $F_0 = \text{PatchEmbed}(X)$ are the output features of the patch embedding layer. The output features of the vision transformer encoder are denoted as $F_{out} \in \mathbb{R}^{\frac{H}{d}\times\frac{W}{d}\times C}$.

3.4. Cross-View Fusion Module

We use ViT [3] as the backbone to extract features from the input image, and then, to alleviate the inconsistent performance among the front, side, and top views, we design a new 3D feature representation architecture called Cross-View Fusion (CVF). It learns the individual features for each of the three views and then forms the 3D feature by fusing the features from each view. As illustrated in Figure 2, the CVF module consists of three branches for three views and Cross-View Attention(CVA) blocks. The first branch is responsible for generating the keypoints from the front view. It takes the feature F_{out} as input and outputs the 2D keypoint heatmaps denoted as $J_{front} \in \mathbb{R}^{H\times W\times k}$. k is the number of joints. Without depth information, predicting the 2D poses in the other two perspectives from the input image is challenging. Thus, we propose to propagate the information from the front view to the side and top views to improve the performance of these two views, as illustrated in Figure 2. We use a query block to convert the 2D heatmaps from the front-view branch into front-view features F_{front}, and adopt two cross-view attention blocks to obtain the side-view features F_{side} and top-view features F_{top}. Then the features of the side view (or top view) can be obtained as follows:

$$F_{side} = F'_{out} + \text{MLP}(\text{softmax}(\frac{QK^T}{\sqrt{d_K}})V) \tag{6}$$

where $F'_{out} = \text{Conv}_{3\times 3}(F_{out})$, $Q = \text{MLP}(F_{front})$, $K = \text{MLP}(F'_{out})$ and $V = \text{MLP}(F'_{out})$. d_K denotes the dimension of K. MLP and softmax represent the MLP layer and softmax layer. The 2D poses in the side view are obtained from the feature F_{side} via several linear layers. The branch of the top view is the same as the above procedure. Finally, we add the three features F_{front}, F_{side}, and F_{top} to form a fused feature F_{fuse} for the regression of 3D poses.

3.5. Reversible Kinematic Topology Decoder

Following the previous works [10, 12], the decoder takes the features as input and estimates the SMPL parameters, *i.e.*, the human shape, the body joint poses, and the camera parameters to reconstruct the human mesh. Previous works ignore the inherent dependence among joints and treat them with the same importance. KTD [33] is proposed to iteratively generate the pose parameters from the root joint to others in hierarchical order according to the fixed kinematic tree. Different from the KTD, our Reversible Kinematic Topology Decoder(RKTD) provides a reversible kinematic tree based on the foot-ground contact prediction.

Firstly, we decode the camera parameters $\phi \in \mathbb{R}^3$ and the human shape $\beta \in \mathbb{R}^{10}$ of the SMPL parameters by the MLP layers directly. Then we introduce one branch to predict the body-scene contacts before the pose regressor. Similar to [6], the body-scene contacts $c \in [0, 1]^{6890\times 1}$ are formulated as per-vertex contact states on the human mesh, which consists of several MLP layers as shown in Figure 2. The contact prediction is supervised under the pseudo annotations of the contacts in Equation (4) with a BCELoss form. We further determine the ground-contact foot according to the foot vertices with a larger number of contact states, *i.e.*, the left foot contact states $c_l = \sum_i^{i\in v_l} c_i$ or the right foot contact states $c_r = \sum_i^{i\in v_r} c_i$. When a foot hits the ground, we sequentially estimate the poses of each joint from the ground-contact foot to the root according to the reverse order in the kinematic tree which is different from the root-to-leaf order in KTD, since the root is driven by a fixed grounded foot. The ith child joint pose is based on the input features F and the poses of the chain of ancestor's joints iteratively as

$$F_i = \begin{cases} \text{CONCAT}(F, \{\theta_k | k \in \text{ancestor}_l(i)\}) & \text{if } c_l > c_r > 0, \\ \text{CONCAT}(F, \{\theta_k | k \in \text{ancestor}_r(i)\}) & c_r > c_l > 0, \\ \text{CONCAT}(F, \{\theta_k | k \in \text{ancestor}_p(i)\}) & \text{otherwise}, \end{cases} \tag{7}$$

where $\theta_i = \text{MLP}(F_i)$. ancestor$_l$, ancestor$_l$ and ancestor$_p$ represent the ancestor joints in the left foot contact kinematic tree, the right foot contact kinematic tree, and the base kinematic tree from root to others used in the KTD, separately.

3.6. Loss Functions

The loss of the whole method is defined as

$$\begin{aligned}\mathcal{L} =&\lambda_{2D}\mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} + \lambda_{SMPL}\mathcal{L}_{SMPL}+ \\ &\lambda_{Contact}\mathcal{L}_{Contact},\end{aligned} \tag{8}$$

where λ_{2D}, λ_{3D}, λ_{SMPL} and $\lambda_{contact}$ are the coefficients of the loss of each part. The details are as follows:

$$\mathcal{L}_{2D} = \frac{1}{n}\sum_{i\in\Omega}^{n}(\pi(j_{3D_i}, T) - j_{2D_i}^{front})^2) + \mathcal{L}_{CVF}, \tag{9}$$

where Ω denotes the set of all 3D keypoints, n is the total number of 3D keypoints, j_{3D_i} represents the predicted position of the ith 3D keypoint in Ω . T means the weak perspective projection camera parameters, π represents the weak perspective projection transformation, $j_{2D_i}^{front}$ denotes the i-th 2D keypoint annotated in the front view (the input image). $\mathcal{L}_{CVF}$ the loss of the CVF intermediate representation module, defined as:

$$\mathcal{L}_{CVF} = \frac{1}{m}\sum_{i\in\omega}^{m}(j_{2D_i} - \hat{j}_{2D_i})^2 + \mathcal{L}_{jd}(j_{2D}, \hat{j}_{2D}, m), \tag{10}$$

$$\mathcal{L}_{jd}(j_{2D}, \hat{j}_{2D}, m) = \frac{1}{m}\sum_{i,j\in\omega}^{m}\Big|\,||j_{2D_i} - j_{2D_j}|| - ||\hat{j}_{2D_i} - \hat{j}_{2D_j}||\,\Big|, \tag{11}$$

where ω represents the set of all 2D keypoints in three views. m denotes the total number of 2D keypoints in the set ω. j_{2D_i} represents the position of ith 2D keypoint in the set ω. If there is not a hat symbol on j_{2D}, it means the value is predicted by the model, otherwise, it is the ground-truth. $\mathcal{L}_{jd}$ refers to the previous work [13], which can make the relative distances between predicted joints match those between the ground-truth joints. $\mathcal{L}_{jd}$ is beneficial to learn some high-dimensional semantics, such as the length of bones and the shape of the overall skeleton pose. In the front view, the coordinates of j_{2D} are in the image space, and the ground-truth comes from the 2D annotations of the input image. In the side and the top view, unlike the front view, the ground-truth comes from the results of the orthogonal projection of the 3D coordinates to the side view and the top view, and their coordinates are normalized, centered on the root and range from -1 to 1.

$$\mathcal{L}_{3D} = \frac{1}{n}\sum_{i\in\Omega}^{n}((j_{3D_i} - \hat{j}_{3D_i})^2 + \mathcal{L}_{jd}(j_{3D}, \hat{j}_{3D}, n), \tag{12}$$

$$\mathcal{L}_{SMPL} = (\beta - \hat{\beta})^2 + (\theta - \hat{\theta})^2 + ||\beta||^2 + ||\theta||^2, \tag{13}$$

where Ω, n, j_{3D_i} are the same as above. $\mathcal{L}_{jd}(j_{3D}, \hat{j}_{3D}, n)$ represents a 3D keypoint version of the $\mathcal{L}_{jd}$, similar to Equation (11). β and θ are the shape and pose parameter of SMPL model.

$$\mathcal{L}_{contact} = -(c\log(\hat{c}) + (1-c)\log(1-\hat{c}). \tag{14}$$

$\mathcal{L}_{contact}$ is about contact signals, where c represents the predicted body-scene contact state, $\hat{c}$ represents the corresponding ground-truth.

4. Experiments

4.1. Implementation Details

The following datasets are used: (1) **MS COCO [17]** provides in-the-wild 2D keypoints annotation. (2) **3DPW [31]** is a widely used in-the-wild dataset for the 3D human pose and shape estimation task. (3) **MPI-INF-3DHP [19]** is a multi-person dataset consisting of constrained indoor and complex outdoor scenes. (4) **Human3.6M [7]** is a commonly used indoor dataset for 3D pose estimation. Following previous works [9, 10], we downsample all videos from 50fps to 10fps. It originally annotated 17 keypoints, excluding the feet. We add 6 feet 2D keypoints annotations on each view and then fit those corresponding 3D foot poses and keypoints for training and evaluating the poses of the feet. (5) **AIST++ [16]** is an indoor dataset with various dance moves. Similarly, we downsample all videos from 50fps to 10fps. For training and evaluating the poses of the feet, we add 2D foot keypoints annotations and fit those corresponding 3D foot poses.

Training and evaluation: We first follow the original ViTPose [34] and train a 2D keypoints estimation model on MS COCO as our pre-trained model. We use mixed training sets to train our 3D human pose estimation model, including Human3.6M, MPI-INF-3DHP, 3DPW, and AIST++. We build a superset containing all keypoint definitions and the mappings between each other, just like the previous works, i.e., VIBE [10]. Since the annotators are proposed for multi-view indoor data, we do not reconstruct the foot poses and contacts on the 3DPW dataset. Our RKTD is only applied to Human3.6M and AIST++. We evaluate our model on the test sets of 3DPW, Human3.6M, and AIST++. The model is trained on eight NVIDIA A100 GPUs for 100 epochs. The mini-batch size is set to 256. The Adam optimizer is adopted with an initial learning rate of 5×10^{-5}, which is reduced by 0.3 times at the 50th, 70th and 90th epochs.

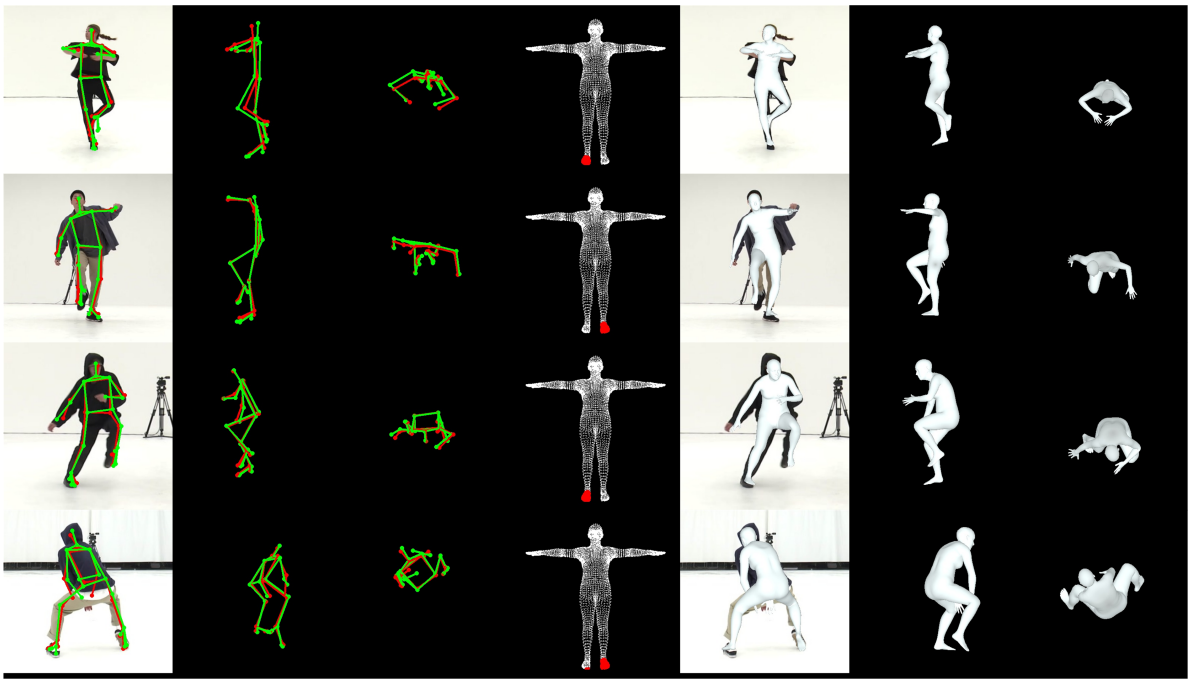


Figure 5. Qualitative results on AIST++. From left to right: tri-view 2D joints, tri-view 3D mesh (green for the ground truth and red for prediction).

4.2. Comparison with state-of-the-arts

We compare our model with the prior arts that focus on model improvement, including the video-based [2, 4, 10, 14, 33] and the image-based [9, 11, 12, 15, 28] methods, which exploit the temporal information or not separately. Since almost none of the previous work evaluates the pose accuracy of the foot, we first obtain the 14 LSP joints from the body mesh by using the same regressor and then compare their Mean Per Joint Position Error (MPJPE), Procrustes-Aligned MPJPE (PA-MPJPE) on those test datasets. the results are summarized in Table 1.

As is shown in Table 1, our model with ViT-Large as backbone achieves state-of-the-art performances on 3DPW test datasets, which outperforms D&D with a 2.6mm improvement on PA-MPJPE and do not introduce the additional temporal information. With the lower computation cost, our model with ViT-Base as backbone performs better than most of the prior arts, including Mesh Graphormer and MAED. As for the Human3.6M evaluation datasets, our model shows similar performance with other works when sharing the same implementation details with 3DPW. On AIST++ test datasets, our model outperforms the Trajectory Optimization physical-based method with a 22mm improvement on PA-MPJPE. From the side view comparisons on 3DPW in Figure 6, our method achieves more stable meshes than those by MAED. As shown in Figure 5, our method obtains the more exact global rotation and foot posture. More comparisons and qualitative results are shown in Supplementary Materials.

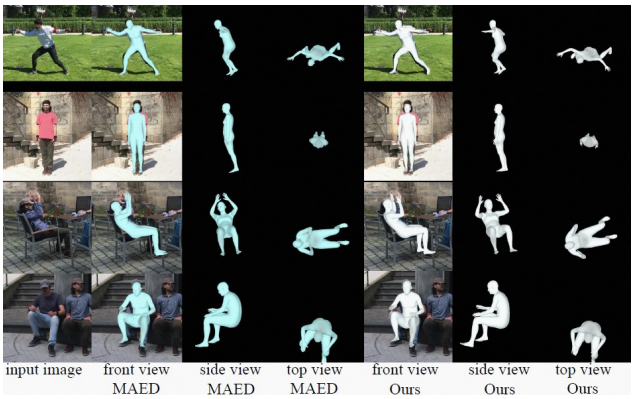


Figure 6. Qualitative comparison on 3DPW. From left to right: input images, tri-view results of MAED, tri-view results of our method.

4.3. Ablation Study

The effectiveness of different backbones. We test three networks, including ResNet50, ViT-Base and ViT-Large, as the feature extraction encoder for exploring the effectiveness of the transformer-based model. Following the training settings in Section 4.1, We use the feature extraction encoder and independent decoder same as [10]. As is illustrated in Tab. 2, ViT-Base outperforms ResNet50 with the

	Method	3DPW		Human3.6M		AIST++	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
Video-Based	TCMR [2]	86.5	52.7	-	-	-	-
	VIBE [10]	82.7	51.9	65.6	41.1	-	-
	MAED [33]	79.1	45.7	56.4	38.7	-	-
	Trajectory Optimization [4]	-	-	84.0	56.0	107.0	67.0
	D&D [14]	73.7	42.7	52.5	35.5	-	-
Image-Based	HMR [9]	130.0	81.3	-	56.8	-	-
	SPIN [12]	96.9	59.2	-	41.7	107.7	-
	HybrIK [15]	80.0	48.8	54.4	**34.5**	-	-
	ROMP [28]	76.7	47.3	-	-	-	-
	PARE [11]	79.1	46.4	-	-	-	-
	Ours-ResNet50	78.8	45.4	-	-	-	-
	Ours-Base	77.8	44.7	55.2	38.9	63.2	45.9
	Ours-Large	**70.8**	**40.1**	**52.4**	36.6	**60.1**	**43.8**

Table 1. Performance comparison between our method and state-of-the-art methods on 3DPW, Human3.6M and AIST++.

Backbone	MPJPE	PA-MPJPE
ResNet50	84.5	48.5
ViT-Base	82.1	47.8
ViT-Large	76.9	44.5

Table 2. Ablation study of the effectiveness on the backbone.

0.7mm improvement on PA-MPJPE, which denotes that the transformer-based model is also suitable for 3D human pose estimation task. When the model goes deeper and wider, ViT-Large achieves 44.5mm PA-MPJPE on the 3DPW test set, surpassing most previous works.

The effectiveness of CVF. Table 3 investigates the efectiveness of each part of Our CVF module. All of the approaches use Human3.6M, MPI-INF-3DHP, and 3DPW as training sets, and use 3DPW as a test set, with ViT-base as the backbone. Table 3 summarises the results of: (1) estimating 3D poses directly from backbone features without any intermediate representation, and estimating with the intermediate representations of (2) only the front view, (3) the front and the top views (4) the front, side and top views without the cross-view fusion, and (5) the three views with the cross-view fusion. This result shows that the whole CVF module achieves an overall improvement of 3.6mm on PA-MPJPE, with each part contributing partly.

The effectiveness of RKTD. We implement three generation types of kinematic tree used in the decoder, including independence which means that the joint does not rely on any other joints, KTD [33] and proposed RTKD. Following the same training implementation on Table 1, we test on the AIST++ evaluation set. As described in Table 4, RKTD outperforms KTD by 0.7mm improvement on whole-body PA-MPJPE, and 1.3mm improvement on foot PA-MPJPE especially. With the help of RKTD, the output meshes have more accurate global pose.

Approach	MPJPE	PA-MPJPE
(1) Baseline	83.1	48.7
(2) Front view	81.7	47.0
(3) Front and top views	79.4	46.1
(4) Three views w/o CVA	80.1	45.9
(5) Three views w/ CVA	**79.4**	**45.1**

Table 3. The effectiveness of each part of Our Cross-View Fusion module. 'Three views' indicates our proposed front, side, and top view representation without the cross-view fusion. 'CVA' indicates our Cross-View Attention blocks as shown in Figure 2.

Approach	MPJPE		PA-MPJPE	
	Whole	Foot	Whole	Foot
Indenpendence	53.6	56.2	37.5	35.2
KTD	52.3	55.9	37.1	34.9
RKTD	**51.5**	**54.6**	**36.4**	**33.6**

Table 4. Ablation study of the effectiveness on RKTD.

5. Conclusion

Although 3D human pose estimation has achieved improvement tremendously, there remain several challenges, *e.g.*, view inconsistency and unnatural foot-ground interaction, when facing more complex body posture in the real world. In this paper, we propose the Cross-View Fusion (CVF) module to construct a better 3D intermediate representation to alleviate the view inconsistency. Then we introduce the optimization-based method to revise the foot poses and foot-ground contacts for the general multi-view datasets. Additionally, the reversible kinematic topology strategy is innovated to utilize the contact information in the SMPL regressor. Extensive experiments on the popular benchmarks demonstrate that our method outperforms the state-of-the-art approaches by a significant margin.

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016. 2, 4

[2] Hongsuk Choi, Gyeongsik M, Chang JY, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, 2021. 3, 7, 8

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3, 5

[4] Erik Gärtner, Mykhaylo Andriluka, Hongyi Xu, and Cristian Sminchisescu. Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video. In *CVPR*, 2022. 7, 8

[5] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM TOG*, 36(4):1–13, 2017. 3

[6] Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and inferring dense full-body human-scene contact. In *CVPR*, 2022. 5

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 36(7):1325–1339, 2014. 2, 6

[8] Lei Jin, Chenyang Xu, Xiaojuan Wang, Yabo Xiao, Yandong Guo, Xuecheng Nie, and Jian Zhao. Single-stage is enough: Multi-person absolute 3d pose estimation. In *CVPR*, 2022. 3

[9] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 3, 6, 7, 8

[10] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1, 3, 5, 6, 7, 8

[11] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part attention regressor for 3D human body estimation. In *ICCV*, 2021. 1, 3, 7, 8

[12] Nikos Kolotouros, Pavlakos G, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 1, 3, 5, 7, 8

[13] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 6

[14] Jiefeng Li, Siyuan Bian, Chao Xu, Gang Liu, Gang Yu, and Cewu Lu. D &d: Learning human dynamics from dynamic camera. In *ECCV*, 2022. 2, 3, 7, 8

[15] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *CVPR*, 2021. 3, 7, 8

[16] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++, 2021. 2, 6

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, (6). 1, 3

[19] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017. 6

[20] Lucas Mourot, Ludovic Hoyet, François Le Clerc, and Pierre Hellier. Underpressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. *arXiv*, 2022. 3

[21] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, 2018. 3

[22] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017. 2

[23] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *CVPR*, 2018. 3

[24] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, 2021. 3

[25] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM TOG*, 39(6), 2020. 3

[26] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 3

[27] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM TOG*, 39(4):54–1, 2020. 3

[28] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, 2021. 3, 7, 8

[29] 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 *CVPR*, 2022. 3

[30] Nicolas Ugrinovic, Adria Ruiz, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Body size and depth disambiguation in multi-person reconstruction from single images. In *3DV*, 2021. 3

[31] 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 *ECCV*, 2018. 2, 6

[32] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *ICCV*, 2021. 2, 3

[33] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *ICCV*, 2021. 2, 3, 5, 7, 8

[34] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022. 3, 5, 6

[35] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *ECCV*, 2020. 3

[36] Yuliang Zou, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang. Reducing footskate in human motion reconstruction with ground contact constraints. In *WACV*, 2020. 3