Decanus to Legatus: Synthetic Training for 2D-3D Human Pose Lifting

Yue Zhu(✉) and David Picard

LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France
{yue.zhu,david.picard}@enpc.fr
https://github.com/Zhuyue0324/Decanus-to-Legatus

Abstract. 3D human pose estimation is a challenging task because of the difficulty to acquire ground-truth data outside of controlled environments. A number of further issues have been hindering progress in building a universal and robust model for this task, including domain gaps between different datasets, unseen actions between train and test datasets, various hardware settings and high cost of annotation, etc. In this paper, we propose an algorithm to generate infinite 3D synthetic human poses (Legatus) from a 3D pose distribution based on 10 initial handcrafted 3D poses (Decanus) during the training of a 2D to 3D human pose lifter neural network. Our results show that we can achieve 3D pose estimation performance comparable to methods using real data from specialized datasets but in a zero-shot setup, showing the generalization potential of our framework.

Keywords: 3D Human pose · Synthetic training · Zero-shot

1 Introduction

3D Human pose estimation from single images [1] is a challenging and yet very important topic in computer vision because of its numerous applications from pedestrian movement prediction to sports analysis. Given an RGB image, the system predicts the 3D positions of the key body joints of human(s) in the image. Recent works on deep learning methods have shown very promising results on this topic [6,21,26,48–50]. Current existing discriminative 3D human pose estimation methods, in which the neural network directly outputs the positions, can be put into two categories: One stage methods which directly estimate the 3D poses inside the world or camera space [29,34], or two stage methods which first estimate 2D human poses in the camera space, then lift 2D estimated skeletons to 3D [18].

However, all these approaches require massive amount of supervision data to train the neural network. Contrarily to 2D annotations, obtaining the 3D annotations for training and evaluating these methods is usually limited to controlled

This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011012640 made by GENCI, and was supported and funded by Ergonova.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-26316-3_16.

L. Wang et al. (Eds.): ACCV 2022, LNCS 13844, pp. 257–274, 2023.
https://doi.org/10.1007/978-3-031-26316-3_16

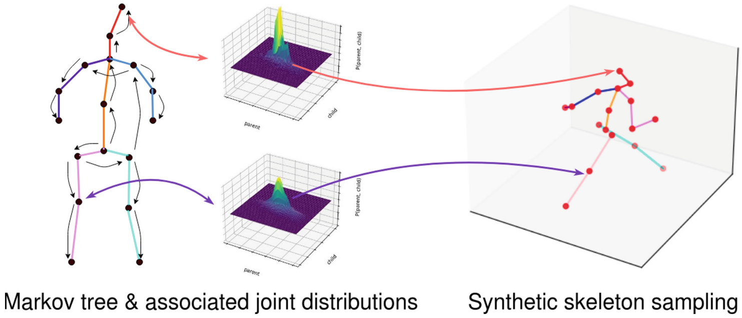
P(parent, child)
parent
child
P(parent, child)
parent
child
Markov tree & associated joint distributions
Synthetic skeleton sampling

Fig. 1. The main idea of our synthetic generation method: use a hierarchic probabilistic tree and its per joint distribution to generate realistic synthetic 3D human poses.

environments for technical reasons (Motion capture systems, camera calibration, etc.). This brings a weakness in generalization to in-the-wild images, where there can be more unseen scenarios with different kinds of human appearances, backgrounds and camera parameters.

In comparison, obtaining 2D annotations is much easier, and there are much more diverse existing 2D datasets in the wild [3,22,51]. This makes 2D to 3D pose lifting very appealing since they can benefit from the more diverse 2D data at least for their 2D detection part. Since the lifting part does not require the input image but only the 2D keypoints, we infer that it can be trained without any real ground-truth 3D information. Training 3D lifting without using explicit 3D ground-truth has previously been realized by using multiple views and cross-view consistency to ensure correct 3D reconstructions [45]. However, multiple views can be cumbersome to acquire and are also limited to controlled environments.

In order to tackle this problem, we propose an algorithm which generates infinite synthetic 3D human skeletons on the fly during the training of the lifter from just a few initial handcrafted poses. This generator provides enough data to train a lifter to invert 2D projections of these generated skeletons back to 3D, and can also be used to generate multiple views for cross-view consistency. We introduce a Markov chain with a tree structure (Markov tree) type of model, following a hierarchical parent-child joint order which allows us to generate skeletons with a distribution that we evolve through time so as to increase the complexity of the generated poses (see Fig. 1). We evaluate our approach on the two benchmark datasets Human3.6M and MPI-INF-3DHP and achieve zero-shot results that are competitive with that of weakly supervised methods. To summarize, our contributions are:

- A 3D human pose generation algorithm following a probabilistic hierarchical architecture and a set of distributions, which uses zero real 3D pose data.
- A Markov tree model of distributions that evolve through time, allowing generation of unseen human poses.
- A semi-automatic way to handcraft few 3D poses to seed initial distribution.
- Zero-shot results that are competitive with methods using real data.

2 Related Work

Monocular 3D Human Pose Estimation. In recent years, monocular 3D human pose estimation has been widely explored in the community. The models can be mainly categorized into generative models [2,4,7,24,33,39,47] which fit 3D parametric models to the image, and discriminative models which directly learn 3D positions from image [1,38]. Generative models try to fit the shape of the entire body and as such are great for augmented reality or animation purpose [35]. However, they tend to be less precise than discriminative models. On the other hand, a difficulty that the discriminative models have is that depth information is hard to infer from a single image when it is not explicitly modeled, and thus additional bias must be learned using 3D supervision [25,26], multiview spatial consistency [13,45,48] or temporal consistency [1,9,23]. Discriminative models can also be categorized into one stage models which predict directly 3D poses from images [14,25,29,34] and two stage methods which first learn a 2D pose estimator, then lift the obtained 2D poses to 3D [18,28,45,48,49,52]. Lifting 2D pose to 3D is somewhat of an ill-posed problem because of depth ambiguity ambiguity. But the larger quantity and diversity of 2D datasets [3,22,51], as well as the already achieved much better performance in 2D human pose estimation provide a strong argument for focusing on lifting 2D human poses to 3D.

Weak Supervision Methods. Since obtaining precise 3D annotations of human poses are hard due to technical reasons and are mostly limited to controlled environments, many research proposals tackled this problem by designing weak supervision methods to avoid using 3D annotations. For example, Iqbal et al. [18] apply a rigid-aligned multiview consistency 3D loss between multiple 3D poses estimated from different 2D views of the same 3D sample. Mitra et al. [30] learn 3D pose in a canonical form and ensure same predicted poses from different views. Fang et al. [13] propose a virtual mirror so that the estimated 3D poses, after being symmetrically projected into the other side of the mirror, should also look correctly, thus simulating another way of 'multiview' consistency. Finally, Wandt et al. [45] learn lifted 3D poses in a canonical form as well as a camera position so that every 3D pose lifted from a different view of a same 3D sample should still have 2D reprojection consistencies. For us, in addition to 3D supervision obtained from our synthetical generation, we also use multiview consistency to improve our training performance.

Synthetic Human Pose Training. Since the early days of the Kinect, synthetic training has been a popular option for estimating 3D human body pose [40]. The most common strategy is to perform data augmentation in order to increase the size and diversity of real datasets [16]. Others like Sminchisescu *et al.* [43] render synthetically generated poses on natural indoor and outdoor image backgrounds. Okada *et al.* [32] generate synthetic human poses in a subspace constructed by PCA using the walking sequences extracted from the CMU Mocap dataset [19]. Du *et al.* [12] create a synthetic height-map dataset to train a dual-stream convolutional network for 2D joints localization. Ghezelghieh *et al.* [15] utilize

3D graphic software and the CMU Mocap dataset to synthesize humans with different 3D poses and viewpoints. Pumarola *et al.* [36] created 3DPeople, a large-scale synthetic dataset of photo-realistic images with a large variety of subjects, activities and human outfits. Both [11] and [25] use pressure maps as input to estimate 3D human pose with synthetic data. In this paper, we are only interested in generating realistic 3D poses as a set of keypoints so as to train a 2D to 3D lifting neural network. As such, we do not need to render visually realistic humans with meshes, textures and colors for this much simpler task.

Human Pose Prior. Since the human body is highly constrained, it can be leveraged as an inductive bias in pose estimation. Bregler*et al.* [8] use kinematic-chain human pose model that follow the skeletal structure, extended by Sigal *et al.* [42] with interpenetration constraints. Chow*et al.* [10] introduced Chow-Liu tree, the maximum spanning tree of all-pairwise-mutual-information tree to model pairs of joints that exhibit a high flow of information. Lehrmann*et al.* [20] use a Chow-Liu tree that maximize an entropy function depending on nearest neighbor distances and learn local conditional distributions from data based on this tree structure. Sidenblahn*et al.* [41] use cylinders and spheres to model human body. Akhter *et al.* [2] learn joint-angle limits prior under local coordinate systems of 3 human body parts as torso, head,and upper-legs. We use a variant of kinematic model because the 3D limb lengths are fixed no matter the view, which can facilitate the generation process of synthetic skeleton.

Cross Dataset Generalization. Due to the diversity of human appearances and view points, cross-dataset generalization has recently been the center of attention of several works. Wang *et al.* [46] learn to predict camera views so as to auto-adjust to different datasets. Li *et al.* [21] and Gong *et al.* [16] perform data augmentation to cover the possible unseen poses in test dataset. Rapczyński *et al.* [37] discuss several methods including normalisation, viewpoint estimation, etc., for improving cross-dataset generalization. In our method, since we use purely synthetic data, we are always in a cross-dataset generalization setup.

3 Proposed Method

The goal of our method is to create a simple synthetic human pose generation model allowing us to train on pure synthetic data without any real 3D human pose data information during the whole training procedure.

3.1 Synthetic Human Pose Generation Model

Local Spherical Coordinate System. Without loss of generalization, we use Human3.6M skeleton layout shown in Fig. 2 (a) throughout the paper. To simplify human pose generation, we set the pelvis joint (joint 0) as root joint and the origin of the global Cartesian coordinate system from which a tree structure is applied to generate joints one by one. We suppose that the position of one joint

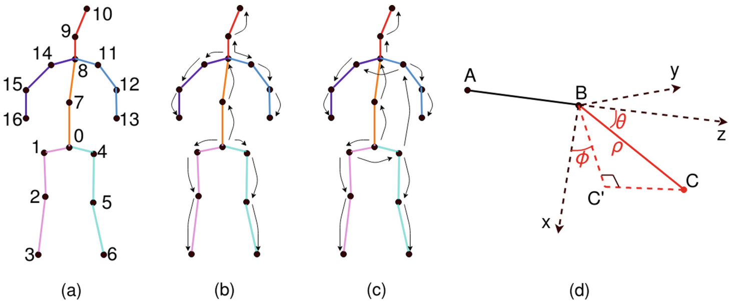

Fig. 2. (a) The 17-joint model of Human3.6M that we use (b) The **parent-child joint relation** graph. With parent joint's coordinate as origin of local spherical coordinate system, it generates child joint's position. (c) The **parent-child ρ, θ , ϕ relation** graph. With parent joint's ρ, θ , ϕ information, it samples child joint's ρ, θ , ϕ. (d) An example of how child joint is generated with sampled ρ, θ , ϕ from relationship in (c) under the local spherical coordinate system with it's parent joint in (b) as origin.

depends on the position of the joint which is directly connected to it but closer (in geodesic meaning) to the root joint. We call this kinematic chain **parent-child joint relations**, as shown in Fig. 2 (b). With this relationship, we propose to generate the child joint in a local spherical coordinate system (ρ, θ, ϕ) centered on its parent joint (see Fig. 2 (d)). The ρ, θ , ϕ values are sampled with respect to a conditional distribution $P(x_{child}|x_{parent})$. This produces a Markov chain indexed by a tree structure which we denote as a Markov Tree.

Our motivation to use a local spherical coordinate system for joint generation is that each human body branch has a fixed length ρ no matter the movement. Also, since the supination and the pronation of the branches are not encoded in skeleton representation, the new joint position can be parameterized with polar angle θ and azimuthal angle ϕ. Furthermore, by using an axis system depending on 'grandparent-parent' branch instead of global coordinate system, the possible angle interval of θ and ϕ achieved by human is more limited than in a global coordinate system. Finally, our local spherical coordinate system is entirely bijective with global coordinate system.

Hierarchic Probabilistic Skeleton Sampling Model. Generating a human pose in our local spherical coordinate system is equivalent to generating a set of (ρ,θ,ϕ). We thus propose to sample these values from a distribution that approximate that of real human poses. To retain plausible poses, we limit the range of (ρ,θ,ϕ) for each joint based on what is on average biologically achievable.

Since body joints follow a tree-like structure, it is unlikely that sampling each joint independently of the others leads to realistic poses. Instead, we propose to model the distribution of the joints by a Markov chain index by a tree following the skeleton, where probability of sampling a tuple (ρ,θ,ϕ) for a joint depends on the values sampled for its parent. More formally, denoting a child joint c and its parent $p(c)$ following the tree structure, we have:

$$(\rho_c, \theta_c, \phi_c) \sim P((\rho, \theta, \phi)|(\rho_{p(c)}, \theta_{p(c)}, \phi_{p(c)})) \quad (1)$$

Please note that the tree structure used for accounting the dependencies between joints as shown on Fig. 2 (c) is slightly different than the kinematic one. We found in practice that it is better to condition the position of one shoulder on the position of the same side hip, and to condition symmetrical shoulder/hip on their already generated counterpart rather than on their common parent. Intuitively, this seems to better encode global consistency.

To facilitate modeling distribution $P((\rho, \theta, \phi)|(\rho_{p(c)}, \theta_{p(c)}, \phi_{p(c)}))$, we make further assumption that all 3 components only depend on their parent counterparts. More formally:

$$\rho_c \sim P(\rho|\rho_{p(c)}),\ \theta_c \sim P(\theta|\theta_{p(c)}),\ \phi_c \sim P(\phi|\phi_{p(c)}) \quad (2)$$

This allows us to model each distribution with a simple non-parametric model consisting of a simple 2D histogram representing the probability of sampling, *e.g.*, ρ_c knowing the value of $\rho_{p(c)}$. In practice, we use 50 bins histograms for each value, totalling to $3 \times 16 = 48$ 2D histograms of size 50×50. When there is no ambiguity, we use the same notation $P(\cdot|\cdot)$ for the histogram and the probability.

3.2 Pseudo-realistic 3D Human Pose Sampling

The next step is to estimate a distribution that can approximate the real 3D pose distribution, and from which our model can sample, so that the generated poses look like real human actions. Under the constraint of zero-shot 3D real data, we choose to make breakthrough by looking at limited amount of 2D real poses and 'manually' lift them into 3D to make our distribution. However, it is impossible for us to tell the exact depths of keypoints from an image with our eye, and it is also a huge amount of work to do if we check a lot of images one by one. Instead, we choose a 3-step procedure to get our handcrafted 3D pose:

High-Variance 2D Poses. We randomly sample 1000 sets of 10 2D-human poses from the target dataset (*e.g.*, Human3.6M). We then compute the total variance for each set and pick the sets with largest variance as our candidates. This ensure our initial pose set has high diversity.

Semi-automatic 2D to 3D Seed Pose Lifting. Next, we use a semi-automatic way to lift samples in each seed set to 3D. The idea is as follows: from an image for which we already know the 2D distances between connected joints, and if we can estimate the 3D length of each branch who connects the joints as well as the proportion λ_{prop} between the 2D length in the image (in pixel) and the 3D length (in centimeter), we can estimate the relative depth between connected joints using Pythagorean theorems under the assumption that the camera produces an almost orthogonal projection. The ambiguity about the sign of these depths, which decide if one joint is in front of or in the back of its parent joint, can easily be manually annotated.

To estimate the 3D length, we define a set of fixed value representing branch lengths ($||c-p(c)||_2, \forall c$ except the root joint) of the human body based on biological data. Since we later calculate under a proportionality assumption between 3D and 2D, we only need it to roughly represent the proportionality between different human bone length. We also manually annotate $sign_c$ for each keypoint c, denoting if it is relatively further or closer to the camera compared to its parent joint $p(c)$. Finally the 2D-3D size proportion λ_{prop} is calculated under the assumption that the 3 joints around the head (head top, nose and neck) form a triangle of known ratio which is independent of rotation and view, visually shown in Fig. 3. This is reasonable since there are no largely moving articulated part in this triplet. We choose $AB = 1$ the unit length and we suppose the proportion between AB, BC and CA is fixed ($BC = \alpha AB$,$AC = \beta AB$). Noting $d_B = B'B - A'A$ and $d_C = C'C - A'A$, for the 2D skeleton we know $A'B'$,$B'C'$ and $A'C'$, then we have 3 unknown variables d_B, d_C, and $\lambda_{prop} = \frac{A'B'(pixels)}{A'B'(meters)}$ and 3 equations:

$$d_B^2 = AB^2 - (\frac{A'B'}{\lambda_{prop}})^2, \quad d_C^2 = (\beta AB)^2 - (\frac{A'C'}{\lambda_{prop}})^2,$$
$$(d_B - d_C)^2 = (\alpha AB)^2 - (\frac{B'C'}{\lambda_{prop}})^2 \tag{3}$$

Then we can solve λ_{prop}. In practice, we set $\alpha = 1$ and $\beta = 5/3$.

After obtaining these depths, we apply Pythagorean theorem to get the final depth value of all joints with the kinematic order. Examples of semi-automatic lifted 3D poses are shown on Fig. 4. Since there are only a few keypoints to label as *in front of* or *behind* their parent joint, the labeling process is very easy and takes about 3 min per image only.

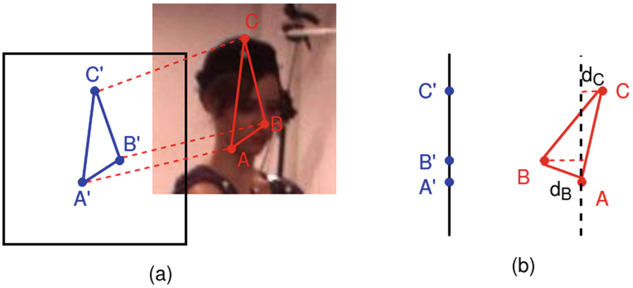

Fig. 3. **(a)** 3D poses (red A,B and C, unit in centimeters) of 3 joints of the head projected onto 2D camera plan (blue A',B' and C', unit in pixels). **(b)** same but right side view after 90^o rotation. (Color figure online)

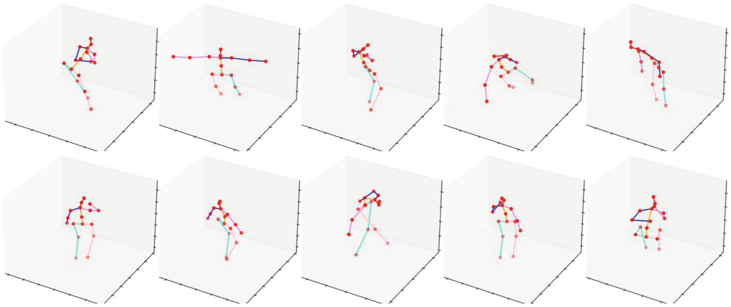

Fig. 4. A example of a set of 10 semi-automatic lifted 3D poses. This set of seeds is also the one which produce our best score on Human3.6M dataset. These 10 lifted samples have a 79.42mm MPJPE error compare to the groundtruth.

Distribution Diffusion. We then transform 3D poses into the local spherical coordinate system and used each seed set as initial distribution to populate the histograms. Since the sampling of a new skeleton follows the Markov tree structure and different limbs have a weak correlation between them in our model, it is possible to sample skeletons that look like combinations of the original 10 samples within the seed set.

However, these initial samplings are by no mean complete, and we run the risk of overfitting the lifter network to these poses only. To alleviate this problem, we introduce a diffusion process among each 2D histogram such that the probability of adjacent parameters is raised over time. More formally:

$$P(x_c|x_{p(c)})_{t+1} = P(x_c|x_{p(c)})_t + \alpha_{x_c} \Delta P(x_c|x_{p(c)})_t,\ x \in \{\rho, \theta, \phi\} \tag{4}$$

where Δ is the Laplacien operator and α_{x_c} is the diffusion coefficient. This idea is derived from the heat diffusion equation in thermodynamics, in which bins with a higher probability diffuse to their neighbours (Laplacian operator), making the generation process more and more likely to generate samples out of initial bin.

The main reason behind our diffusion process is that of curriculum learning [5]. At first, the diversity of sampled skeletons is low and the neural network is able to quickly learn how to lift these poses. At later stage, the diffusion process allows the sampling process to generate more diverse skeletons that are progressive extensions of the initial pose angles, avoiding overfitting the original poses. We show in Fig. 5 an example of evolution of the histogram and increase of generation variety through diffusion.

3.3 Training with Synthetic Data

The training setup of 2D-3D lifter network l_w is shown on Fig. 6 and consists of 3 main components: (1) Sampling a batch of skeletons at each step; (2) sampling different virtual cameras to project the generated skeletons into 2D; and finally

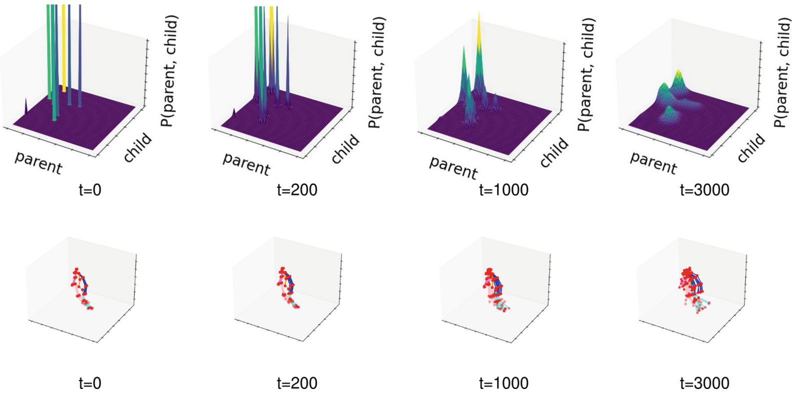

Fig. 5. First row is an example of the distribution histogram of a joint after 0, 200, 1000 and 3000 steps of diffusion. Second row shows an example of slightly increased generation variety when sampling from a single bin and generating 10 samples each time after 0, 200, 1000 and 3000 steps of diffusion.

Algorithm 1 Sampling algorithm

```
Require: True distribution $P_t$, empirical distribution $P_e$;
  $bins \leftarrow$ where $P_t > 0$ and $P_e \leq P_t$
  $b \sim \mathcal{U}(bins)$
  return Random sample from $b$
```

(3) the different losses used to optimize l_w. In practice, l_w is a simple 8-layer MLP with 1 in-layer, 3 basic residual blocks of width 1024, and 1 out-layer, adapted from [45].

When sampling a new batch of skeleton using our generator, we have to keep in mind that the distribution of the generator varies through time because of the diffusion process introduced in Eq. 4. To avoid over-sampling or under-sampling bins with low density, we propose to track the amount of skeletons that have been generated in each bin and adjust the sampling strategy accordingly. More formally, let us denote P_t the *true distribution* obtained by Eq. 4, and P_e the *empirical distribution* obtained by tracking the generation process. The corrected sampling algorithm is shown in Algorithm 1 and basically selects uniformly a plausible bin ($P_t > 0$) that has not been over-sampled ($P_e \leq P_t$). The whole generation process simply loops over all joints using the Markov tree and is shown on Algorithm 2.

At initialization, we sample 5000 real 2D poses, compute the proportion of nearest neighbour within each pose seed, and use it to initialize the histogram to give more importance to more frequent poses.

Regarding the projection of the batch into 2D, we propose to sample a set of batch-wise rotation matrices $R_{1,\dots,N}$, mostly rotating around the vertical axis, to simulate different viewpoints. Then, the rotated 3D skeletons are just simply: $X_{3D,i} = R_i X_{3D,0}, \quad i \in \{1,\dots,N\}$, with $X_{3D,0}$ being the original skeleton in

Algorithm 2 Pose generation algorithm

```
Require: True distribution $P_t$, empirical distribution $P_e$, Markov tree structure $T$,
  sampling algorithm S
  $X \leftarrow 0_{(J,3)}$                                        ▷ $3{=}\rho, \theta, \phi$
  for $i \in \rho, \theta, \phi$ do                                ▷ root joint
      $X[0,i] \leftarrow$ S( $P_t(X_0), P_e(X_0)$)
  end for
  for (p,c) in $T$ do                          ▷ parent-child relations in $T$
      for $i \in \rho, \theta, \phi$ do
          $X[c,i] \leftarrow$ S($P_t(X_{(c,i)}|X_{(p,i)}), P_e(X_{(c,i)}|X_{(p,i)})$)
          Update $P_e(X_{(c,i)}|X_{(p,i)})$
      end for
  end for
  return $X$ in Cartesian coordinates
```

global Cartesian coordinates. To simulate the cameras, we follow [45] and use a scaleless orthogonal projection:

$$X_{2D,i} = \frac{WX_{3D,i}}{\|WX_{3D,i}\|_F}, \quad W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \tag{5}$$

where W is the orthogonal projection matrix and $\|\cdot\|_F$ is the Frobenius norm. Normalizing by the Frobenius norm allows us to be independent of the global scale of $X_{2D,i}$ while retaining the relative scale of each bone with respect to each other. In practice, we found that uniformly sampling random rotation matrices at each batch renders the training much more difficult. Instead, we sample view with a small noise around the identity matrix and let the noise increase as the training goes on to generate more complex views at later stages.

Finally, to train the network, we leverage several losses. First, since we have the 3D ground-truth associated with each generated skeleton:

$$\mathcal{L}_{3D} = \frac{1}{N} \sum_{i=1..N} \left\| \frac{\hat{X}_{3D,i}}{\|\hat{X}_{3D,i}\|_F} - \frac{X_{3D,i}}{\|X_{3D,i}\|_F} \right\|_1, \tag{6}$$

with $\hat{X}_{3D,i} = l_w(X_{2D,i})$ being the output of the lifter l_w, and $\|\cdot\|_1$ the ℓ_1 norm. 3D skeletons are normalized before being compared because the input of the lifter is scaleless and as such it would make no sense to expect the lifter to recover the global scale of X_{3D}. Then, we use the multiple views generated thanks to R_i to enforce a multiview consistency loss. Calling $\hat{X}_{2D,i,j} = WR_jR_i^{-1}\hat{X}_{3D,i}$ the projection of the lifted skeleton from view i into view j, we optimize the cross-view projection error:

$$\mathcal{L}_{2D} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left\| \frac{\hat{X}_{2D,i,j}}{\|\hat{X}_{2D,i,j}\|_F} - \frac{X_{2D,j}}{\|X_{2D,j}\|_F} \right\|_1 \tag{7}$$

The global synthetic training loss we use is the following combination:

$$\mathcal{L} = \mathcal{L}_{2D} + \lambda_{3D}\mathcal{L}_{3D} \tag{8}$$

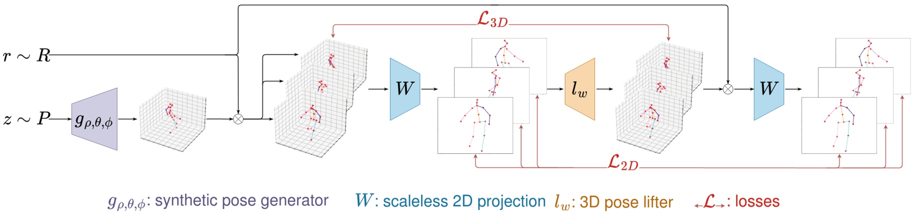

Fig. 6. Our whole training process with synthetic data. Our generator g generates a 3D human pose following given distributions P of ρ, θ and ϕ. It will be applied with multiple different random generated r to project into different camera view. Projector W will projects them into scaleless 2D coordinates and they are the network inputs. The output estimated 3D poses will be applied with scaleless 3D supervision loss $\mathcal{L}_{3D}$, and also cross-view scaleless 2D reprojection loss $\mathcal{L}_{2D}$, which rotate estimated 3D pose from one view to another with known r and apply 2D supervision after projection W.

4 Experiments

4.1 Datasets

We use two widely used dataset Human3.6M [17] and MPI-INF-3DHP [29] to quantitatively evaluate our method.

We only use our generated synthetic samples for training and evaluate on S9 and S11 of Human3.6M and TS1-TS6 on MPI-INF-3DHP with their common protocols. In order to compare the quality of our generated skeletons with real 2D data, We also use the COCO [22] and MPII [3] datasets to check the generalizability of our method with qualitative evaluation.

4.2 Evaluation Metrics

For the quantitative evaluation on both Human3.6M and MPI-INF-3DHP we use MPJPE, i.e. the mean euclidean distance between the reconstructed and ground-truth 3D pose coordinates after the root joint is aligned ($P1$ evaluation protocol of Human3.6M dataset). Since we train the network with a scaleless loss, we follow [45] and scale the output 3D pose's Forbenius norm into the ground-truth 3D pose's Forbenius norm in order to compute the MPJPE. We also report PCK, i.e. the percentage of keypoints with the distance between predicted 3D pose and ground-truth 3D pose is less or equal to half of the head's length.

4.3 Implementation Details

We use a batch-size of 32 and we train for 10 epochs on a single 16G GPU using Adam optimizer and a learning rate of 10^{-4}. We set the number of views $N = 4$ and the total number of synthetic 2D input samples for each epoch is the same as the number of H36M training samples to make a fair comparison. The distribution diffusion coefficient α_{x_c} is a joint-wise loss dependent value, set to

Table 1. Comparison of our results with the state-of-the-arts under the common protocol 1 on Human 3.6M and MPI-INF-3DHP. The value before and after ± symbol are mean and standard deviation values.

		Weak supervision			Synthetic training				**Ours**	
		[18]	[30]	[45]	[21]	[15]	[12]	[44]	10 sets	Best run
H36M	MPJPE↓	67.4	120.95	65.9	106.8	$\geq$ 78.13	126.47	111.6	95.4 ± 13.5	60.8
3DHP	MPJPE↓	109.3	-	104.0	-	-	-	-	148.4 ± 7.6	132.8
	PCK↑	79.5	-	77.0	-	-	-	-	57.7 ± 2.3	61.9

$10^{-5} \times 10^{|\delta\mathcal{L}|/(10\times N)}$ where $\delta\mathcal{L}$ is the joint-wise difference between loss of the last batch and the current batch, and the rotation R are sampled with a noise that increases in $\frac{1}{2\times\#batch}$ after each step, with $\#batch$ the number of elapsed batches in the current epoch. For the loss, $\lambda_{3D} = 0.1$ is set empirically. To account for the variation due to the selection of the 2D pose using total variance, we keep the 10 sets with highest variance and show averaged results. Our method trains on about 100k generated samples per hour on a V100 GPU, whereas inference time for lifting is negligible.

4.4 Comparison with the State-of-the Art

We compare our results with the state-of-the-art methods with synthetic supervision for training in Table 1. We present several weak supervision methods which also do not use real 3D annotations, and instead use other sort of real data supervision whereas we do not. We can see that our method outperforms these synthetic training methods and achieves the performance on par with weakly supervised methods on H36M, while never using a real example for training.

We show qualitative results on the COCO dataset on Fig. 7. Since the COCO layout is different from that of H36M, we use a linear interpolation of existing joints to localize the missing joints. We can see that our model still achieves good qualitative performances on zero shot lifting of human poses in the wild (first 2 rows). Failed predictions (last row) tend to bend the legs backward even when the human is standing still, which may be a bias of the generator.

5 Ablation Studies

5.1 Synthetic Poses Realism

We want to see how similar our synthetic skeletons are to real skeletons. Qualitatively we compare our distribution after diffusion with the distribution of the whole Human3.6M and MPI-INF-3DHP datasets, for some of the joints as shown in Fig. 8. We can see that, even though there are many poses in MPI-INF-3DHP have never appear in Human3.6M, the distributions of angles θ and ϕ of these two real datasets have very similar shapes, which means our local spherical coordinate system successfully models the invariance of the biological achievable

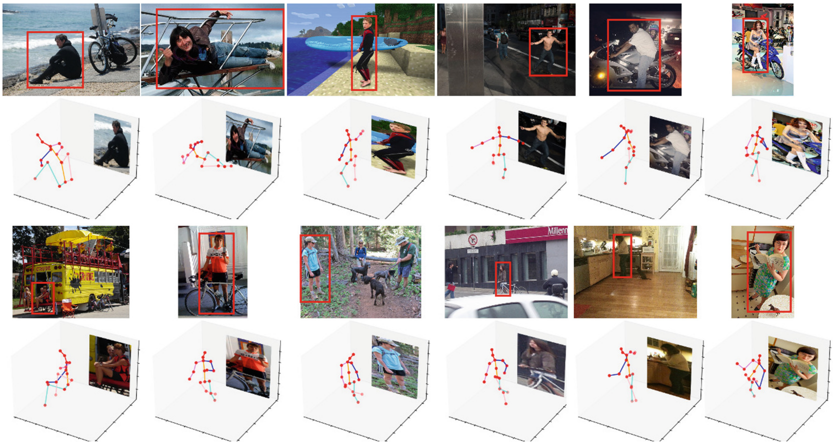

Fig. 7. Example of zero shot lifting in the wild on images from the COCO dataset. The first row are visually correct prediction, while the last row presents ’failure’ cases, mostly due to right leg learnt a bias of leaning backward.

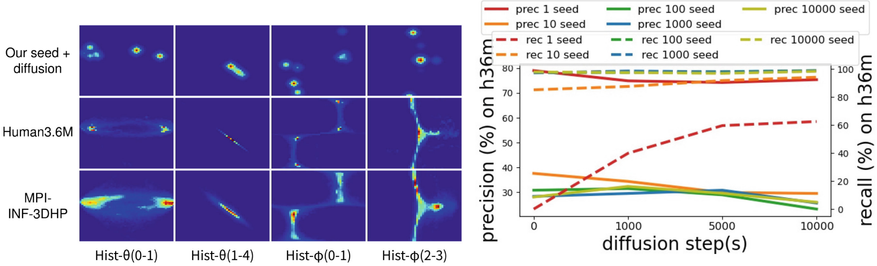

Fig. 8. Left: Examples of distributions of angle θ and ϕ from same parent-child pairs computed on Human3.6M, MPI-INF-3DHP, and our diffusion process. **Right**: Precision and recall evaluated with 5k generated samples and 5k real 2D samples from h36m.

human pose angles and their frequencies which are independent of camera view point. Our seeds+diffuse strategy produces a Gaussian mixture which succeed in covering big parts of real dataset’s distribution.

Quantitatively we apply a precision/recall test, as is common practice with GANs [31]. We sample 5000 real and 5000 synthetic poses and project them to 2D plane using the scaleless projection in 5 and the Euclidean distance. Precision (resp. Recall) is defined as percentage of synthetic samples (resp. real samples) inside the union of the balls centered on each real sample (resp. synthetic sample) and with a radius of the distance to its 10-th nearest real sample neighbor (resp. synthetic sample neighbor). In our case, we already know that most synthetic skeleton generated by our Markov tree are biologically possible thanks to the limits in the generation intervals. As such, we are more interested in a very high recall so as to not miss the diversity of real skeletons. All our seed sets have

Table 2. Results on the 24-keypoint SMPL model, compared to the state-of-the-art

Method	Labeled training data	MPJPE↓
CLIFF (ECCV22)	H36m + 3DHP + COCO + MPII + 3DPW	**52.8**
DynaBOA (TPAMI22)	H36m + 3DPW	65.5
Ours	24 samples from 3DPW	61.09 ± 2.16

more than 70% recall and highest one achieves 91.8% recall. The precision, on the other hand, is around 40%, with 47.1% as the highest, which is still good considering we only start with 10 manually lifted initial poses for each seed.

5.2 Effect of Diffusion

We want to see why diffusion process is essential to our method. We take respectively 1, 10, 100, 1000 and 10000 samples of 3D poses on Human3.6M dataset as initial seed to make distribution graphs, and apply our 2D precision recall test after diffusion process. The result is shown in Fig. 8. We can see that diffusion generally increase recall value at the cost of precision value. The distribution using 1 samples as seed is much worse with the others in recall which means it can only cover around 60% of samples from real dataset even with diffusion process, while the distribution using 100 samples or more are close in performances. The diffusion process can reduce the gap between the distribution using 10 samples as seeds and those using 100 or more samples, which is important to us considering we want to avoid handcrafting a lot of initial poses.

5.3 Layout Adaptation

We show that our synthetic generation and training method also work on a different keypoint layout by applying the whole process on a newly defined hierarchic Markov tree based on 24 keypoints of SMPL model [24] and evaluating on 3DPW dataset [27]. We use 24 samples from its training set (one frame from each video) using our 2D variance based criterion for the seeds. Since our training method is scaleless, we rescale the predicted 3D poses by the average Forbenius norm of the 24 samples in the seed. The average MPJPE of 10 different seeds is shown in Table 2. This validates the generalization capability of our method.

6 Conclusion

We present an algorithm which allows to generate synthetic 3D human skeletons on the fly during the training, following a Markov-tree type distribution which evolve through out time to create unseen poses. We propose a scaleless multiview training process based on purely synthetic data generated from a few handcrafted poses. We evaluate our approach on the two benchmark datasets Human3.6M and MPI-INF-3DHP and achieve promising results in a zero shot setup.

References

1. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 44–58 (2006)
2. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3d human pose reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
3. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: new benchmark and state of the art analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
4. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM Transactions on Graph (2005)
5. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning (2009)
6. Bhatnagar, B.L., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: Combining implicit function learning and parametric models for 3D human reconstruction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 311–329. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_19
7. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep It SMPL: automatic estimation of 3D human pose and shape from a single image. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 561–578. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_34
8. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: Proceedings of 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231) (1998)
9. Choi, H., Moon, G., Chang, J.Y., Lee, K.M.: Beyond static features for temporally consistent 3d human pose and shape from a video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
10. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Trans. Inf. Theory. **14**, 462–467 (1968)
11. Clever, H.M., Erickson, Z., Kapusta, A., Turk, G., Liu, K., Kemp, C.C.: Bodies at rest: 3d human pose and shape estimation from a pressure image using synthetic data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
12. Du, Y., et al.: Marker-less 3D human motion capture with monocular image sequence and height-maps. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_2
13. Fang, Q., Shuai, Q., Dong, J., Bao, H., Zhou, X.: Reconstructing 3d human pose by watching humans in the mirror. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
14. Gärtner, E., Pirinen, A., Sminchisescu, C.: Deep reinforcement learning for active human pose estimation. In: AAAI (2020)
15. Ghezelghieh, M.F., Kasturi, R., Sarkar, S.: Learning camera viewpoint using CNN to improve 3d body pose estimation. In: 3D Vision (2016)
16. Gong, K., Zhang, J., Feng, J.: PoseAug: a differentiable pose augmentation framework for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

17. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6 m: large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans. Pattern. Anal. Mach. Intell. **36**, 1325–1339 (2014)
18. Iqbal, U., Molchanov, P., Kautz, J.: Weakly-supervised 3d human pose learning via multi-view images in the wild. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
19. Lab, C.G.: Motion capture database (2001). http://mocap.cs.cmu.edu
20. Lehrmann, A.M., Gehler, P.V., Nowozin, S.: A non-parametric Bayesian network prior of human pose. In: 2013 IEEE International Conference on Computer Vision (2013)
21. Li, S., et al.: Cascaded deep monocular 3d human pose estimation with evolutionary training data. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
22. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
23. Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.c., Asari, V.: Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
24. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) (2015)
25. Luo, Y., Li, Y., Foshey, M., Shou, W., Sharma, P., Palacios, T., Torralba, A., Matusik, W.: Intelligent carpet: Inferring 3d human pose from tactile signals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
26. Ma, X., Su, J., Wang, C., Ci, H., Wang, Y.: Context modeling in 3d human pose estimation: A unified perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
27. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 614–631. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_37
28. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings IEEE International Conference on Computer Vision (ICCV) (2017)
29. Mehta, D., et al.: Monocular 3d human pose estimation in the wild using improved CNN supervision. In: 2017 Fifth International Conference on 3D Vision (3DV) (2017)
30. Mitra, R., Gundavarapu, N.B., Sharma, A., Jain, A.: Multiview-consistent semi-supervised learning for 3d human pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
31. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: International Conference on Machine Learning (2020)
32. Okada, R., Soatto, S.: Relevant feature selection for human pose estimation and localization in cluttered images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 434–445. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88688-4_32

33. Pavlakos, G., et al.: Expressive body capture: 3d hands, face, and body from a single image. In: CVPR (2019)
34. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: CVPR (2017)
35. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. In: ICCV (2021)
36. Pumarola, A., Sanchez, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: modeling the geometry of dressed humans. In: International Conference in Computer Vision (ICCV) (2019)
37. Rapczyński, M., Werner, P., Handrich, S., Al-Hamadi, A.: A baseline for cross-database 3d human pose estimation. Sensors. **31**, 3769 (2021)
38. Rhodin, H., et al.: Learning monocular 3d human pose estimation from multi-view images. In: Proceedings/CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2018)
39. Schmidtke, L., Vlontzos, A., Ellershaw, S., Lukens, A., Arichi, T., Kainz, B.: Unsupervised human pose estimation through transforming shape templates. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
40. Shotton, J., et al.: Real-time human pose recognition in parts from single depth images. In: CVPR 2011 (2011)
41. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45053-X_45
42. Sigal, L., Isard, M., Haussecker, H., Black, M.J.: Loose-limbed people: estimating 3D human pose and motion using non-parametric belief propagation. Int. J. Comput. Vision. **98**, 15–48 (2011)
43. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Learning joint top-down and bottom-up processes for 3d visual inference. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006) (2006)
44. Varol, G., et al.: Learning from synthetic humans. In: CVPR (2017)
45. Wandt, B., Rudolph, M., Zell, P., Rhodin, H., Rosenhahn, B.: CanonPose: self-supervised monocular 3d human pose estimation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
46. Wang, Z., Shin, D., Fowlkes, C.C.: Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. CoRR (2020)
47. Xu, H., Bazavan, E.G., Zanfir, A., Freeman, W.T., Sukthankar, R., Sminchisescu, C.: Ghum & Ghuml: generative 3d human shape and articulated pose models. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
48. Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., Zhang, W.: Deep kinematics analysis for monocular 3d human pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
49. Xu, T., Takano, W.: Graph stacked hourglass networks for 3d human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

50. Zanfir, A., Bazavan, E.G., Xu, H., Freeman, W.T., Sukthankar, R., Sminchisescu, C.: Weakly supervised 3D human pose and shape reconstruction with normalizing flows. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 465–481. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_28
51. Zhang, S.H., et al.: Pose2seg: detection free human instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
52. Zhou, X., Zhu, M., Leonardos, S., Derpanis, K.G., Daniilidis, K.: Sparseness meets deepness: 3d human pose estimation from monocular video. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)