Multi-Person Pose Regression with Distribution-Aware Single-Stage Models

Leyan Zhu*, Zitian Wang*, Si Liu†, Xuecheng Nie, Luoqi Liu, Bo Li

Abstract—Understanding human posture is a challenging topic, which encompasses several tasks, e.g., pose estimation, body mesh recovery and pose tracking. In this paper, we propose a novel Distribution-Aware Single-stage (DAS) model for the pose-related tasks. The proposed DAS model estimates human position and localizes joints simultaneously, which requires only a single pass. Meanwhile, we utilize normalizing flow to enable DAS to learn the true distribution of joint locations, rather than making simple Gaussian or Laplacian assumptions. This provides a pivotal prior and greatly boosts the accuracy of regression-based methods, thus making DAS achieve comparable performance to the volumetric-based methods. We also introduce a recursively update strategy to progressively approach the regression target, reducing the difficulty of regression and improving the regression performance. We further adapt DAS to multi-person mesh recovery and pose tracking tasks and achieve considerable performance on both tasks. Comprehensive experiments on CMU Panoptic and MuPoTS-3D demonstrate the superior efficiency of DAS, specifically 1.5 times speedup over previous best method, and its state-of-the-art accuracy for multi-person pose estimation. Extensive experiments on 3DPW and PoseTrack2018 indicate the effectiveness and efficiency of DAS for human body mesh recovery and pose tracking, respectively, which prove the generality of our proposed DAS model.

Index Terms—Multi-Person Pose Estimation, Single-Stage Model, Normalizing Flow, Recursive Update Strategy

1 Introduction

Human pose estimation is a widely studied task, which aims to localize the positions of people and their joints in images or videos. Recently, 3D pose estimation has drawn much attention thanks to the rapid expansion of AR/VR [2], [3], gaming [4], [5], human-computer interaction [6], [7], etc.

Most of the existing methods achieve human pose estimation in a two-stage manner, which can be divided into *top-down* methods and *bottom-up* methods. The top-down scheme [8], [9], [10] first localizes absolute 3D positions of people and separately estimates the root-relative joint locations for each person. The top-down methods usually require the involvement of human detectors due to the need for the localization of human bodies. The bottom-up scheme [11], [12], [13] detects all joints in the first stage and groups them into the corresponding people in the second stage, which is usually achieved by adopting a manually designed matching algorithm. Although achieve good results, they also suffer from problems such as redundant calculations and complicated post-processing, which result in unsatisfactory performance during deployment. Additionally, despite many methods can be extended to new tasks by adding branches, this may affect the parallelism of the model and increase the inference latency. There are also some methods designed for specific tasks, which lack flexibility and may require large changes when extending to new tasks.

Based on the above considerations, we propose to design a simple and flexible pipeline for human pose estimation and related tasks to advance the application of these techniques in realistic scenarios. Inspired by previous works in the 2D counterpart [14], [15], [16], we aim to design a single-stage pipeline that localizes the human body and their joints simultaneously. However, the extension from 2D to 3D is non-trivial, due to the ill-posed nature of the setup for deriving depth information from a monocular RGB image without prior knowledge of the data distribution. Moreover, designing a unified generic architecture for multiple tasks is difficult because the structures and distributions of data for

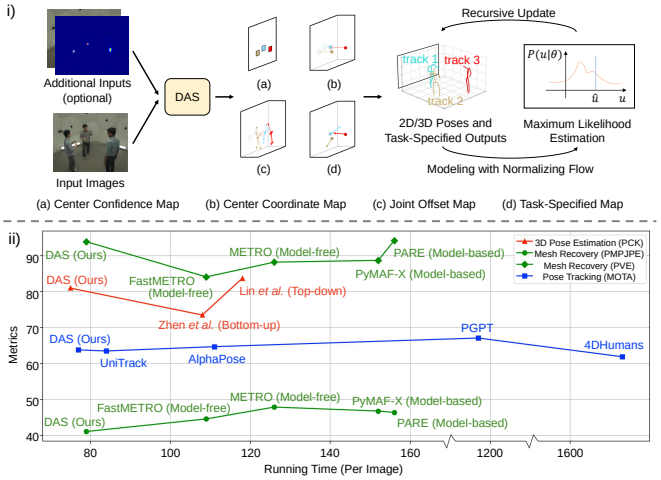


Fig. 1. Overview of our Distribution-Aware Single-Stage model for pose-related tasks. i) The brief pipeline of DAS. Taking pose tracking as an example for the additional task. The arrows in the task-specific map indicate the displacements of the human body centers between adjacent frames. ii) Comparison with state-of-the-art methods on MuPoTS-3D, 3DPW, and PoseTrack2018 datasets. DAS outperforms all methods in the figure in terms of efficiency. For PCK and MOTA, higher values indicate better performance, while for PMPJPE and PVE, lower values indicate better performance. Better zoom in and view in color.

- *Leyan Zhu, Zitian Wang, Si Liu, and Bo Li are with Institute of Artificial Intelligence, Beihang University, and also with Hangzhou Innovation Institute, Beihang University. The first two authors contribute equally to this work and † indicates the corresponding author.*
- *Xuecheng Nie and Luoqi Liu are with MT Lab, Meitu Inc.*

A preliminary version of this work has been published in CVPR 2022 [1].

different tasks often vary greatly.

In order to achieve this goal, we introduce a novel Distribution-Aware Single-stage (DAS) model in this paper. The proposed DAS model tackles the ill-posed problem of multi-person 3D pose estimation from two aspects: 1) DAS represents the 3D pose with a 2.5D human center together with 3D center-relative joint offsets. The human center coordinate is 2.5D because it is estimated from a combination of a 2D center confidence map and a 3D center coordinate map. This relocates the problem of depth estimation from the camera coordinate system to the image coordinate system while unifying the 3D localization of person position and body joints, making the monocular-based one-pass solution possible. 2) DAS learns the true distribution of body joint locations during optimization instead of simply assuming that the joint locations fit a Gaussian or Laplacian distribution. This provides a reliable prior for predicting the locations of the joints, thus boosting the performance of the regression-based model. To alleviate the distribution estimation difficulty, DAS exploits a recursive update strategy to progressively approach the targets. In this way, DAS can efficiently estimate accurate 3D poses from a single RGB image. Additionally, for different targets, DAS estimates them with a regression-based scheme. Despite the various structures and distribution of different targets, they can be regressed similarly. As a result, DAS requires only minor changes to regress on the new target, and benefiting from the recursive update strategy, DAS gives good results for various regression targets.

Specifically, we implement the DAS model with a regression-based pipeline, which outputs 3D human poses via a single forward inference from an input image. As illustrated in Figure 1 i) (a) and (b), DAS models the human center with a center confidence map and a center coordinate map. The former is used to localize the projected human centers in the 2D image coordinate space, while the latter is for estimating the pixel-wise absolute center positions in 3D camera space. DAS leverages a joint offset map to densely encode 3D center-relative locations of body joints, as shown in Figure 1 i) (c). By inserting task-specific branches, DAS is capable of producing additional maps for extended tasks, as illustrated in Figure 1 i) (d). In this way, DAS produces various kinds of maps in parallel and then easily reconstructs multiple 3D poses, mesh vertices and pose trajectories with them, while avoiding redundant computation and complex association. With this compact single-stage pipeline, DAS can achieve superior efficiency over prior two-stage methods.

In previous works, it is most common to use conventional L1 or L2 loss for supervision when optimizing regression models. However, it is proved that this kind of supervision actually makes a simple Laplacian or Gaussian assumption on data distribution, which deviates significantly from the true distribution according to [17]. In contrast, DAS learns the underlying distribution of 3D body joint locations via exploiting the normalizing flows [18], [19]. This helps derive a suitable distribution for model output, thus providing valuable priors for the regression of body joint coordinates. The distribution learning module is optimized through maximum likelihood estimation, together with the pose regression modules during the training phase, and it is removed in the inference phrase. In this way, DAS boosts the regression performance without additional computation costs. Meanwhile, DAS iteratively updates joint offsets by leveraging the informative predictions around regression targets to further facilitate the localization of joints. With this distribution-aware design, DAS achieves higher accuracy compared to bottom-up methods while remaining competitive with top-down ones, as shown in Figure 1 ii).

We implement DAS with a fully convolutional neural network in an end-to-end learnable manner. In order to demonstrate the flexibility and versatility of DAS, we further extend it to pose-related tasks, including body mesh recovery and pose tracking. Only minor adjustments are made during the migration. For body mesh recovery, we add a branch to predict SMPL [20] parameters for each person, which can be encoded into human body mesh vertices using the SMPL model. For pose tracking, we alter the input module to allow learning of the temporal information. The displacement branch which predicts human center offset between adjacent frames and greedy matching algorithms are utilized to help with the prediction of pose trajectories. Analysis is conducted on both tasks and our proposed DAS model is shown to be effective for these extensive tasks.

Comprehensive experiments are conducted on benchmarks CMU Panoptic [21], MuPoTS-3D [11], 3DPW [22], and PoseTrack2018 [23], and the results show the superiority of our proposed DAS model.

In summary, our contributions are in three folds:

- We present a novel single-stage model, for multi-person 3D pose estimation from a monocular RGB image, which overcomes the drawbacks of computation cost and model complexity that occur in two-stage methods.
- We introduce a recursive update strategy for optimization and utilize normalizing flow to learn the true distribution of body joints, thus alleviating the regression difficulty and boosting the performance.
- We propose a method to cost-efficiently migrate DAS to new tasks and extend DAS to pose-related tasks, e.g., body mesh recovery and pose tracking. Experiments show that our proposed method yields good results on various regression tasks with high efficiency.

This paper is built upon our work in CVPR 2022 [1] and significantly extends it. First, we propose a way to migrate the DAS model to a new task. Due to the flexibility of the single-stage model, DAS can be extended to new tasks by simply altering the input and adding task-specific branches. Second, we extend the DAS to multi-person body mesh recovery, which involves predicting dense surface shapes for each person in the given images. We conduct experiments on the popular yet challenging benchmark 3DPW, and our extended DAS outperforms previous model-based methods in terms of MPJPE and PMPJPE metrics. The extended model also achieves state-of-the-art performance on PMPJPE among all model-based and model-free methods, while achieving comparable performance to model-based methods on other metrics. Third, we explore the performance of DAS on pose tracking, which requires the model to capture temporal information in videos. The extended model outperforms previous bottom-up SOTA methods in terms of average precision of joints and achieves satisfying performance on multiple object tracking accuracy. Fourth,

we further conduct more experiments including an ablation study and qualitative analysis in order to demonstrate the effectiveness of our method on sparse pose, dense pose, and temporal scenarios. These experiments highlight the generalizability of our model across a variety of tasks. We also provide the running time of our model on all tasks. Compared to other similar methods, our method achieves minimal inference latency, demonstrating the superiority of our method in efficiency.

2 Related Work

2.1 Single-Stage Human Pose Estimation

Most of the methods [8], [11], [24], [25] achieve human pose estimation in a two-stage manner. These approaches can be further divided into top-down methods and bottom-up methods. The top-down methods [8], [9], [26] utilize human detectors to extract person bounding boxes and a single-person pose estimator is used to generate human poses. The bottom-up methods [11], [12], [13] first localize the instance-agnostic joints and then associate them to form complete human poses.

Dissimilar to two-stage methods, the single-stage methods regard human pose estimation as a combination of body center localization and center-to-joint regression problem. Instead of separately predicting the positions of the body and joints, these approaches estimate body center and joint offsets in parallel. In this way, joint positions are acquired by correlating body center and joint offsets, thus avoiding manually designed grouping post-processing and making the model end-to-end trainable. Zhou *et al.* [16] achieve single-stage human pose estimation by directly regressing joint locations from the human center. Instead of regressing all joints from the center, Nie *et al.* [14] propose a hierarchical structured pose representation to better predict long-range displacements for certain joints. Wei *et al.* [15] focuses on initialization for pose estimation and introduces prior human poses through point-set anchors. Zhou *et al.* [27] achieves bottom-up human pose estimation in an end-to-end manner. This method parses human body features at multiple granularities and associates them with sparse joints through the proposed dense to sparse projection fields.

In this paper, we build our model based on the single-stage paradigm. Different from existing single-stage methods, we focus on the regression process. Normalizing flow is utilized to learn the true distribution of the targets and recursive flow-based optimization is used to improve the understanding of joint distribution.

2.2 Single-Stage Human Body Mesh Estimation

Skinned Multi-Person Linear (*abbr.* SMPL) model [20] is one of the most widely used parametric human body models. It describes the human body shape with low-dimensional shape and pose statistical parameters, and is widely used to encode high-dimensional human body mesh vertices. Based on whether parametric human body models are required to produce body mesh vertices, approaches to human body mesh estimation can be divided into model-free methods and model-based methods.

Model-free methods are capable of predicting mesh vertices without the participation of parametric human body models. Most of the model-free single-stage methods are implemented based on transformer architecture. Lin *et al.* introduce a transformer-based method, METRO [28], to predict 3D joint coordinates and mesh vertices from a single image, and they also present a graph-convolution-reinforced transformer termed Mesh Graphormer [29] to effectively model both global and local interactions among 3D mesh vertices and body joints. Cho *et al.* [30] follows METRO [28] and Mesh Graphormer [29] and improved the transformer architecture, resulting in higher efficiency and accuracy.

Instead of directly producing mesh vertices, model-based methods firstly estimate SMPL parameters from images, and then the parameters are encoded as mesh vertices using SMPL models. Kanazawa *et al.* [31] propose an end-to-end model, which is optimized by minimizing the reprojection loss of keypoints. Adversarial learning is also introduced as a further constraint. Arnab *et al.* [32] presents a 3D human pose and mesh estimation algorithm, which utilizes a bundle adjustment module to output accurate SMPL and camera parameters. Sun *et al.* [33] achieve multi-person mesh estimation with ROMP, a one-stage pipeline, which obtains the 3D mesh by parsing the body center heatmap and sampling the mesh parameter heatmap.

In this paper, our method is implemented in a model-based manner, which regresses SMPL parameters with a trainable head and encodes the parameters into mesh vertices. Compared to the single-stage method ROMP [33], we incorporate the SMPL branch into our model as a plugin and utilize normalizing flow to enhance the regression of SMPL parameters.

2.3 Normalizing Flow for Pose and Mesh Estimation

Normalizing flows are capable of transforming a standard Gaussian distribution into a complex posterior distribution. Some recent works introduce normalizing flow to optimize the regression of human poses. Xu *et al.* [34] propose a statistical model to obtain 3D articulated human shape, which leverages normalizing flow to build the prior. Biggs *et al.* [35] and Wehrbein *et al.* [36] utilize normalizing flow to regress human poses and shapes from ambiguous and occluded images. Wandt *et al.* [37] introduce an unsupervised approach that estimates the 3D pose that is most likely over random projections, with the likelihood estimated using normalizing flows on 2D poses. Compared with methods that improve regression performance by introducing structural priors using grammar models [38], [39], [40], normalizing flow provides higher flexibility.

Unlike the above methods, we follow Li *et al.* [17] to introduce a flow-based optimization scheme for our regression model. The underlying distribution of human body joints and SMPL parameters are modeled with normalizing flow, and the joint locations are updated recursively. The regression model is optimized through maximum likelihood estimation.

2.4 Human Pose Tracking

Human pose tracking [23], [41], [42], [43] is an emerging task involving temporal information, which aims at estimating human poses from videos and assigning unique IDs to each keypoint across video frames. PoseTrack [23], [41] and MPII

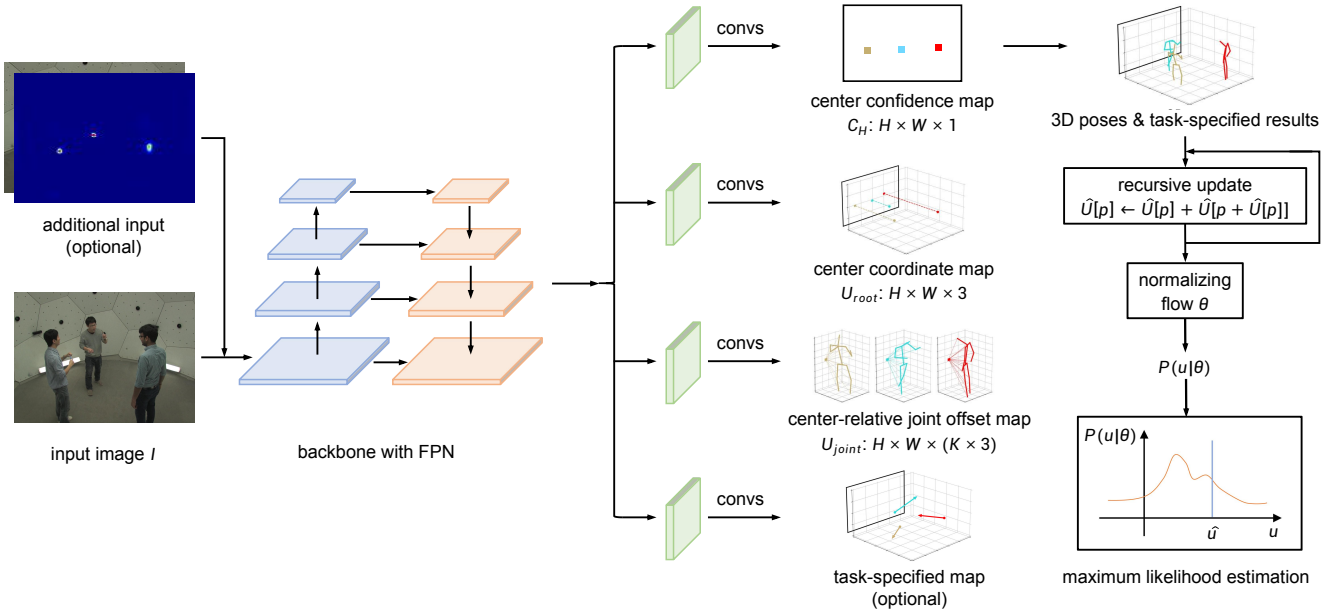


Fig. 2. Illustration of our proposed DAS pipeline. The input images are fed into a 2D CNN backbone for feature extraction. A followed FPN neck is utilized to generate multi-level feature maps of different sizes. Shared prediction heads are built upon the FPN feature maps, which are responsible for predicting center confidence maps, center coordinate maps, and joint offset maps. The 3D joint locations can be reconstructed with the predicted human centers and joint offsets. The probability distribution of 3D joint locations is modeled by normalizing flow with recursive updates. Maximum likelihood estimation is applied to assist the learning process. By introducing additional inputs and task-specified branches, the pipeline can be extended to other tasks. Here we take pose tracking as an example of an additional task to show the structure of the model. The arrows in the tasks-specific map indicate the displacements of human body centers between adjacent frames.

Video Pose dataset [42] are primary benchmarks of this task. Most human pose tracking methods are two-stage, that is, human pose estimation and temporal correlation. Girdhar *et al.* [44] introduce a detect-and-track method, which predicts person keypoints and then links them over time using bipartite matching. Similar to [44], Xiu *et al.* [45] further introduce pose flow building and pose flow non-maximum suppression to achieve pose tracking. Ning *et al.* [46] propose a top-down method for online pose tracking.

Different from the above methods, we follow CenterTrack [47] to build a single-stage model, achieving pose tracking in an end-to-end manner. The keypoint displacements are predicted among frames, thus obtaining pose trajectories without the use of manually designed rules.

3 Method

3.1 Overview

We first provide an overall introduction to the proposed Distribution-Aware Single-Stage (DAS) model for multi-person pose estimation, mesh recovery, and pose tracking in this section.

The overall architecture of our model is illustrated in Figure 2. Given an RGB image $I^{H\times W\times 3}$ as input, the CNN backbone with FPN [48] first extracts multi-level feature maps from I, which are leveraged to handle humans of different scales. On top of the feature maps, three parallel branches are responsible for human center detection, center coordinate regression and center-relative joint offset regression, respectively. By combining and decoding the outputs of the branches, the multi-person 3D poses can be obtained. To enhance the model's joint localization capability, a recursive flow-based optimization scheme is proposed for precise regression of the joint location distribution. The model can be extended to other tasks by adding additional branches as plugins. For 3D multi-person mesh recovery, we add a branch to regress SMPL parameters for each person. Human body mesh vertices can be recovered by decoding the estimated parameters with the SMPL model [20]. For pose tracking, inspired by [47], we make adjustments to the inputs of the model and add a temporal correlation branch, enabling the tracking of human body joint trajectories. Details will be presented in the sections below.

3.2 Human Center Localization

We achieve human center localization with two parallel branches: one for detecting the human body center, and another for regressing the center coordinate. We adopt a ground truth assignment strategy to handle multi-scale features.

Ground truth assignment. To better handle multi-scale human body information, we assign people of different sizes to different feature maps. For a given person with joint location $H^{\text{img}} = \{(x_k^{\text{img}}, y_k^{\text{img}}, d_k^{\text{img}}) \mid k \in [1...K]\}$, depending on the maximum Gaussian distance of its joints from the root $r_{\max} = \max_k \sqrt{(x_k^{\text{img}} - x_{\text{root}}^{\text{img}})^2 + (y_k^{\text{img}} - y_{\text{root}}^{\text{img}})^2}$, it is assigned to a specific FPN feature map F_l with a downsample stride s_l. Formally, the regression range for F_l is set to $[r_{m-1}, r_m]$. H will be assigned to the lth feature map if $r_{\max}$ is within the range of $[r_{m-1}, r_m]$. Then the coordinates are scaled to $H = \{(x_k, y_k, d_k) \mid k \in [1...K]\}$ to match the downsample stride, s.t. $x_k = x_k^{\text{img}}/s_l$ and $y_k = y_k^{\text{img}}/s_l$.

Human center detection. We use the root joint (i.e., pelvis) as the human body center and treat the human center detection as binary classification. The center confidence map

$C_{\text{center}}^{H\times W\times 1}$ measures whether the pixels represent any body center. In addition to the centroid pixel, the N_{pos} nearest pixels around each body center are also treated as positive samples. In the ground truth, we assign 1 to the confidence of positive samples and 0 to others. We supervise the training with the loss function:

$$L_{cls} = \text{FocalLoss}(\hat{C}_{\text{center}}, C_{\text{center}}), \tag{1}$$

where $\hat{C}_{\text{center}}$ is the predicted confidence map, C_{center} is the ground truth, and FocalLoss is proposed in [49]. Meanwhile, a centerness branch is adopted to measure the quality of the center. The design of the centerness target and loss function L_{center} follows [49]. Specifically, we adopt binary cross entropy (BCE) loss for L_{center}:

$$L_{\text{center}} = \text{BCE}(\hat{C}_{\text{center}}, C_{\text{center}}). \tag{2}$$

Center coordinate regression. For a given positive sample p, the center coordinates can be regressed from p with the center coordinate map $U_{\text{root}}^{H\times W\times 3}$. The regression target for the center $j_{\text{root}} = (x_{\text{root}}, y_{\text{root}}, d_{\text{root}})$ represented by p is set as the offset $U_{\text{root}}[p] = (x_{\text{root}} - x_p, y_{\text{root}} - y_p, d_{\text{root}})$. We adopt L1 loss for center coordinate regression:

$$L_{\text{root}} = \sum_p ||\hat{U}_{\text{root}}[p] - U_{\text{root}}[p]||_1, \tag{3}$$

where $\hat{U}_{\text{root}}[p]$ is the predicted center offset, and $U_{\text{root}}[p]$ is the ground truth.

3.3 3D Body Joint Localization

Given that the body center coordinates are obtained, it's necessary to estimate the center-relative joint offset to obtain the joint coordinates. We estimate the offsets in a similar way to the body center and model the joint location distribution with a normalizing flow. We also employ a recursive update strategy to optimize the learned distribution.

Center-relative joint offset regression. Unlike heatmap-based methods which require an additional stage for joint association, this regression-based method makes joint localization and identification a holistic process. The center-relative offsets are regressed from each positive sample, denoted as p. We predict the center-relative offset map $\hat{U}_{\text{joint}} = \{\hat{U}_1, ..., \hat{U}_K\}$, where $\hat{U}_k^{H\times W\times 3}$ is responsible for the 3D center-relative offset of the kth joints. Same as center coordinate regression, the regression target for $H = \{j_k \mid k \in [1...K]\}$ at p is set as $U_k[p] = j_k - j_{\text{root}}$. Commonly used regression loss (e.g., L1 loss) can be adopted for optimization. The objective can be written as:

$$L_{\text{pose}} = \sum_k \sum_p ||\hat{U}_k[p] - U_k[p]||_1, \tag{4}$$

where $\hat{U}_k[p]$ represents the predicted center-relative joint offset, and $U_k[p]$ represents the ground truth.

Joint location distribution modeling. Compared to the heatmap-based methods which commonly represent the joint distribution as Gaussian distribution, regression-based methods represent the joint distribution deterministically, which could be more inclined to be affected by label noises, occlusions, and invisibilities. Taking the above factors into account, we utilize a normalizing flow [18], [19], [50], [51] to model the center-relative joint location in a probability distribution form: $u \sim P(u)$, where u is center-relative joint location.

Following [17], we model the distribution with reparameterization. Formally, $P(u)$ is transformed from a zero-mean distribution $z \sim P_Z(z)$ by scaling and shifting. The transformation function can be written as $u = \bar{u} + \sigma \cdot z$, where $\bar{u}$ represents the expectation of joint location, and σ indicates the scale of the distribution. Given this transformation function, the density function of $P(u)$ can be calculated as:

$$\log P(u) = \log P_Z(z) - \log \sigma. \tag{5}$$

In this way, instead of regressing the deterministic center-relative joint location u, we regress the expectation $\bar{u}$ and scale indicator σ. We model the zero-mean distribution $P_Z(z)$ with a normalizing flow (e.g., real NVP [50]). The expectation $\bar{u}$ is the only parameter to be calculated to obtain the joint location during the inference phase.

Recursive flow based optimization. Since the feature used to regress the pose is selected from the center of the body, it may be less representative of joints that are far from the root joint. Due to the complexity of the human body, this spatial misalignment between features and targets may lead to larger errors than their counterparts in bounding box regression methods [49], [52], [53].

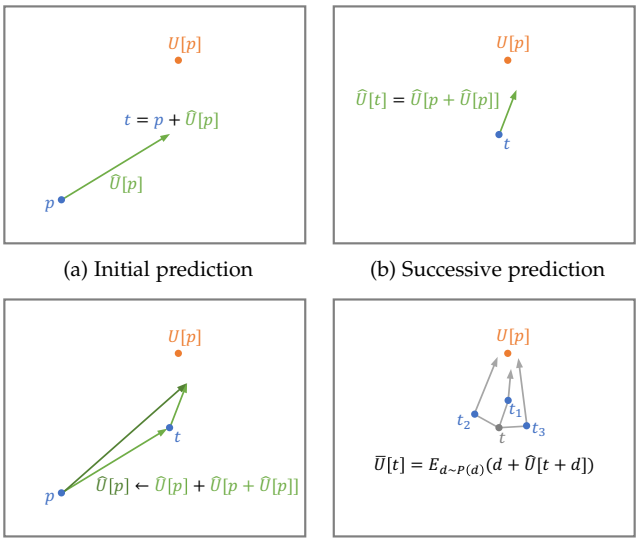


Fig. 3. Illustration for recursive update strategy in DAS model. Subscripts are omitted for brevity.

To deal with this problem, we propose a recursive update scheme to optimize the joint location expectation $\bar{u}$ iteratively, as shown in Figure 3. We take the offset map of the kth joint as an example to introduce the mechanism. Given the initial prediction $\bar{u} = \hat{U}_k^n[p]$ from positive sample p, it is updated by the *local prediction* from $p + \hat{U}_k^n[p]$:

$$\hat{U}_k^{n+1}[p] \leftarrow \hat{U}_k^n[p] + B(\hat{U}_k^{\text{local}}[p + \hat{U}_k^n[p]]), \tag{6}$$

where $B(\cdot)$ stands for bilinear interpolation function which is used to obtain values from non-integer coordinates.

The same prediction map as $\hat{U}_k^n$, i.e., $\hat{U}_k^{\text{local}} = \hat{U}_k^n$, can be used for local prediction. After the prediction is updated,

the features are more representative of target joints and thus produce higher-quality joint offsets. In this way, $\bar{u}$ is updated recursively as in Figure 3c:

$$\begin{aligned} \hat{U}_k^{n+1}[p] &\leftarrow \hat{U}_k^n[p] + \hat{U}_k^n[p + \hat{U}_k^n[p]] \\ \bar{u} &\leftarrow \hat{U}_k^{n+1}[p], \end{aligned} \tag{7}$$

where the interpolation is omitted for simplicity.

Additionally, we consider another multi-source update strategy that approximates the expectation for better modeling $\bar{u}$. In this case, $\hat{U}_k^{\text{local}}[t]$ is calculated by gathering multiple predictions around t, as shown in Figure 3d:

$$\begin{aligned} \hat{U}_k^{\text{local}}[t] &= E_{d\sim P_D(d)}(d + \hat{U}_k[t+d]) \\ &\approx \sum_m P_D(d_m)(d_m + \hat{U}_k[t+d_m]), \end{aligned} \tag{8}$$

where d_m and $P_D(d_m)$ are the sample location and probability for the mth sample source generated by MLP.

We implement the recursive update strategy with convolutional layers and interpolation layers. By stacking update layers on the top of the joint offset map, $\bar{u}$ can be optimized progressively without modification to the model pipeline. This formulation also avoids manual target assignment at each position of the joint offset map.

We exploit maximum likelihood estimation for parameter optimization in the training phrase to take advantage of the distribution-aware representation. Joint location distribution can be expressed as in Equation 5 given the obtained $\bar{u}$ and σ. When modeling $P_Z(z)$ using a normalizing flow model θ, the maximum likelihood estimation (MLE) objective is formulated as:

$$\begin{aligned} L_{\text{MLE}} &= -\log P(u)|_{u=U} \\ &= -\log P_z(\hat{z}|\theta) + \log\sigma, \end{aligned} \tag{9}$$

where $\hat{z} = (U - \bar{u})/\sigma$, U is the ground truth location and $\bar{u}$ is the estimated expectation of center-relative joint location. By optimizing L_{MLE}, the distribution $P_Z(z|\theta)$ can be learned along with $\bar{u}$.

In this work, we further follow [17] to utilize residual log-likelihood estimation (RLE). RLE factorizes the distribution $P_Z(z)$ into one prior distribution $Q_Z(z)$, e.g., Laplace distribution and Gaussian distribution, and one learned distribution $G_Z(z|\theta)$. The RLE objective is formulated as:

$$L_{\text{RLE}} = -\log Q_Z(\hat{z}) - \log G_Z(\hat{z}|\theta) + \log\sigma \tag{10}$$

We recommend readers to refer to the original paper [17] for more details. In experiments, we implement the RLE objective to replace the L1 loss for L_{pose}.

Reconstruction of 3D pose. The joint coordinates can be obtained by adding center-relative offsets to the coordinates of the human center without requiring an association step. This allows us to obtain 3D locations of human joints in a single forward pass.

3.4 Extending to Additional Tasks

We design a model extension method based on plug-in modules, which allows extending the model to additional tasks without having to make massive adjustments to the model. When extending our model to other tasks, e.g. human mesh recovery and pose tracking, all that needs to be done is to modify the input and add task-specific branches, which demonstrates the flexibility of our method.

3D Multi-Person Mesh Recovery. We achieve human mesh recovery in a model-based manner. Following ROMP [33], we predict camera parameters and SMPL parameters, which can then be decoded into body mesh vertices. We drop the last 2 hand joints instead of using the full 24 joints SMPL model. The 3D rotation of the root joint represents the global orientation of the human body, while the rest represent the 3D orientation of each body part in the kinematic chain relative to its parent. For a given positive sample p, we regress camera and SMPL parameters from p with mesh parameter map $P^{H\times W\times 145}$. The length of the third dimension of the mesh parameter map is the sum of all parameters. In detail, the parameter includes camera parameters $\in \mathbb{R}^3$, pose parameters $\in \mathbb{R}^{22\times 6}$ and shape parameter $\in \mathbb{R}^{10}$. The regression target at p is set as $P[p] = \{P_{\text{camera}}, P_{\text{pose}}, P_{\text{shape}}\}$, where the parameters correspond to the person indicated by p. We supervise the regression with L2 loss L_{param}, which keeps the same design as [33]. Similar to pose estimation, L_{param} is replaced by the RLE objective implemented by us.

Multi-Person Pose Tracking. The goal of pose tracking is to predict the trajectories of people in videos, which requires the model to capture temporal features. Inspired by [47], we associate the detected body centers between adjacent frames with a center displacement map $D^{H\times W\times 2}$. For joints that appear in two consecutive frames, we calculate the horizontal and vertical displacements between these two frames. By adding the position and displacement of the joint in the previous frame, we obtain the position of this joint in the next frame, thereby achieving inter-frame matching. We regress the root joint displacements and perform matching as described above. Similar to joint offset regression, given the corresponding human body centers in frame p_t and in the previous frame p_{t-1}, the regression target at p_t is set as $p_{t-1} - p_t$. The objective can be expressed as:

$$L_{\text{displacement}} = \sum_t \sum_p ||\hat{D}_t[p] - D_t[p]||_1, \tag{11}$$

where $\hat{D}_t[p]$ is the predicted center displacement from the previous frame and $D_t[p]$ is the ground truth. As an extension, we further include the L1 supervision of each joint in $L_{\text{displacement}}$. In this case, the dimensionality of $D_t[p]$ becomes $H \times W \times 2K$, where K is the number of joints. To help the model capture temporal information, we alter the input into 3 parts, including the image of the current frame, the image of the previous frame and the body center map of the previous frame. The inputs are encoded and added before being fed into the backbone.

3.5 Training and Inference

Training. During the training phase, we first transform the 3D joint coordinate in the original data into image coordinate system using the camera intrinsic parameters. Considering the depth ambiguity of the same object captured by different cameras, the absolute representation of depth is hard for the model to learn directly. As a result, we follow previous work [54] and use normalized depth $d_{\text{norm}} = d/f$ as the target depth for center coordinate regression, where

f is the camera focal length. Apart from this, for center-relative pose offset, we do not normalize the depth value for training stabilization. The overall objective is as follows:

$$L = L_{\text{cls}} + \lambda_1 L_{\text{center}} + \lambda_2 L_{\text{root}} + \lambda_3 L_{\text{pose}} + \lambda_4 L_{\text{ext}}, \quad (12)$$

in which λ_i ($i \in \{1...4\}$) are loss weights and L_{ext} is loss for extending tasks. For mesh recovery, $L_{\text{ext}} = L_{\text{param}}$, and for pose tracking, $L_{\text{ext}} = L_{\text{displacement}}$.

Inference. The input images (and body center maps in the pose tracking task) are first fed into the model to produce all intermediate results. Then the center confidence map is filtered with a threshold (set as 0.05) and positions with high response are selected as positive samples. Human body centers and center-relative joint offsets corresponding to the positive samples are extracted from the maps to form the camera-centric 3D poses. Non-maximum suppression is adopted to reduce redundant pose hypotheses. The coordinate of each joint can be obtained by adding the body center coordinate and the center-relative joint offset.

For human body mesh recovery, camera and SMPL parameters are also extracted from the positive positions. The extraction process is the same as for the other parameters. We utilize an SMPL model [20] to encode the parameters into mesh vertices. For pose tracking, human center displacement is taken from the positive samples. Following [47], a greedy matching algorithm is adopted to match people across the frames according to the displacements. Specifically, first, add the center and displacement of the previous frame to obtain the expected position of the center in this frame, and then match the (expected position, actual position) pairs with the smallest Euclidean distance in sequence. In this way, the same human body can be matched between the consecutive frames.

4 Experiments

4.1 Datasets and Evaluation Metrics

We conduct experiments on tasks including 3D multi-person pose estimation, 3D multi-person mesh recovery and 2D multi-person pose tracking. For a fair comparison with previous methods, we use different datasets to train models for different tasks. Details are as follows:

3D Multi-Person Pose Estimation. We employ 2 dataset settings to conduct experiments on the pose estimation task.

The first setting is CMU Panoptic [21]. CMU Panoptic is a large-scale dataset that provides 3D pose annotations for real-world indoor human activities, containing 65 sequences of social activities. Following the evaluation protocol proposed by [55], we use videos from HD cameras indices 16 and 30. For training, we use a mixed dataset containing COCO [60] and video clips from 3 activities: Haggling, Mafia, and Ultimatum. For evaluation, we choose 9600 frames from 4 activities: Haggling, Mafia, Ultimatum, and Pizza as test set. There is no overlap between the training and test sets. The depth information is ignored in loss calculation for images with only 2D pose annotations. Following previous works [55], [61], [62], we utilize the Mean Per Joint Position Error (MPJPE) for performance evaluation. MPJPE is calculated after aligning the keypoints by the root joint.

The second setting includes MuCo-3DHP and MuPoTS-3D [11]. MuCo-3DHP and MuPoTS-3D are multi-person 3D pose datasets. We use both datasets for training and evaluation, respectively. MuCo-3DHP is a large-scale synchronized 3D pose dataset generated by randomly compositing the people from single-person 3D pose dataset MPI-INF-3DHP [63]. MuPoTS-3D is a realistic dataset captured from real-world outdoor scenes. MuPoTS-3D consists of 20 video sequences, each containing up to 3 subjects. We use a mixed dataset containing COCO and MuCo-3DHP for training, and MuPoTS-3D for evaluation. Following previous works [54], [64], we utilize 3D Percentage of Correct Points (3DPCK) for performance evaluation. Especially, PCK_{rel} is used to represent performance after root alignment, and PCK_{abs} is used to represent performance under camera coordinate system. One joint is judged to be correct if it is within a 15 cm distance from the matched ground truth.

3D Multi-Person Mesh Recovery. Following [33], we train our model with a mixture of multiple pose datasets. For a fair comparison with previous work, we use only a subset of the data used in [33] for training. The mixed dataset contains 3 large-scale 3D human body pose dataset (Human3.6M [65], MPI-INF-3DHP [63], MuCo-3DHP [11]) and 4 in-the-wild 2D pose datasets (Crowdpose [66], LSP [67], [68], MPII [69], MS COCO [60]). We sample from each dataset with a fixed probability to compose the training data, and the details are shown in Table 3. For evaluation, we adopt 3DPW [22] as the benchmark. 3DPW is an in-the-wild multi-person dataset with accurate 3D pose annotations, which contains 60 video sequences. Before evaluation, we fine-tune our model on 3DPW training set. We employ MPJPE and Procrustes-aligned MPJPE (PMPJPE) to evaluate 3D pose estimation performance. To evaluate the accuracy of recovered human body mesh vertices, we utilize the Per Vertex Error (PVE) to calculate the 3D surface recovery error.

Multi-Person Pose Tracking. For a fair comparison with existing methods [46], [70], we train and evaluate our model on PoseTrack2018 dataset [23], which contains 550 video sequences with 66374 frames. Following previous works [46], [70], we adopt the Average Precision (AP) to measure accuracy of joint detection and Multi-Object Tracking Accuracy (MOTA) to measure the performance of pose tracking.

4.2 Implementation Details

The parameters of the model vary slightly depending on the task. We introduce the basic settings first, and then we introduce the specific settings for extended tasks. Settings for the extended tasks take the same parameters as the basic settings unless specified otherwise.

Basic Settings. We adopt ResNet-50 [74] with FPN [48] pretrained on COCO dataset [60] for the backbone. When testing the influence of a larger backbone, we replace ResNet-50 with MSPN [75], which is also pretrained on COCO dataset. The regression branches are implemented with a 2D convolution followed by Group Normalization [76] and ReLU. The number of feature channels is set to 256 in FPN and all branches. We resize and pad the images during training and evaluation. For CMU Panoptic dataset, the images are transformed into 1333×640, and for MuCo-3DHP and MuPoTS-3D, the size is 1280×768. When assigning ground truths of different sizes to feature maps, we set the regression range of 4 feature maps as

TABLE 1
Comparison with 3D pose estimation SOTAs on CMU Panoptic dataset in MPJPE. The top-down methods and bottom-up methods are two-stage methods and the bottom row is our single-stage method. * means that mean MPJPE is recalculated by averaging over activities following [55].

Method	Haggling ↓	Mafia ↓	Ultimatum ↓	Pizza ↓	*Mean* ↓
Top-down methods					
Popa et al. [56]	217.9	187.3	193.6	221.3	203.4
Zanfir et al. [55]	140.0	165.9	150.7	156.0	153.4
Wang et al. [57]	50.9	**50.5**	50.7	68.2	55.1*
Bottom-up methods					
Zanfir et al. [58]	72.4	78.8	66.8	94.3	78.1*
Fabbri et al. [13]	**45**	95	58	79	69
Zhen et al. [59]	63.1	60.3	56.6	67.1	61.8
Ours	53.3	51.2	**49.1**	**61.5**	**53.8**

TABLE 2
Comparison with 3D pose estimation SOTAs on MuPoTS-3D dataset. PCK_{rel} and PCK_{abs} are calculated over all groundtruths. † indicates that the running time is reproduced based on the official repository and ‡ means the result is only reported on matched groundtruths.

Method	Runtime ↓	PCK_{rel} ↑	PCK_{abs} ↑
Top-down methods			
Rogez et al. [71]	N/A	70.6	N/A
Moon et al. [54]	107 ms †	81.8	31.5
Wang et al. [57]	N/A	82.0	**43.8**
Lin et al. [72]	118 ms †	**83.7** ‡	35.2 ‡
Bottom-up methods			
Mehta et al. [11]	N/A	65.0	N/A
Mehta et al. [73]	N/A	70.4	N/A
Zhen et al. [59]	108 ms †	73.5	35.4
Ours	**75 ms**	82.7	39.2

TABLE 3
Dataset used for training mesh recovery model.

Dataset	Length	Sample Probability	Expected Length
Human3.6M [65]	62224	0.3	207413
MPI-INF-3DHP [63]	40357	0.2	201785
MuCo-3DHP [11]	20000	0.1	200000
LSP [67], [68]	6829	0.05	136580
MPII [69]	9831	0.1	98310
MS COCO [60]	26455	0.125	211640
Crowdpose [66]	9963	0.125	79704

$[0, 80)$, $[80, 160)$, $[160, 320)$ and $[320, \infty)$ respectively. For each positive sample on the human body center map, we assign 1 to all pixels within a radius of 1.5 around the root joint and 0 to others. During training, we set the initial learning rate to 2×10^{-3}, and learning rate warming-up strategy is adopted in the first 250 iterations. We train the model for 25 epochs and reduce the learning rate by a factor of 10 after 16 and 20 epochs. The weight of each loss is set to the same value during training for 3D pose estimation: $\lambda_1 = \lambda_2 = \lambda_3 = 1$, $\lambda_4 = 0$. All models are trained on 4 NVIDIA RTX 3090 GPUs with 4 samples on each GPU, and inference speeds are measured with a single NVIDIA Tesla V100 GPU. Our method is implemented with Pytorch.

Settings for Mesh Recovery. Considering the complexity of human mesh recovery, we use a larger FPN with 256, 512, 1024, and 2048 input channels, while keeping the number of output channels at 256. We alter the regression ranges to $[0, 80)$, $[80, 160)$, and $[160, \infty)$ since we resize all images to a smaller size of 512×512. The model is trained for 50 epochs, and the learning rate is reduced after 30 and 40 epochs. Additionally, we fine-tune our model on the 3DPW training set for 10 epochs with an initial learning rate set as 2×10^{-4}. The loss function is composed of the 2D keypoint loss, 3D keypoint loss, SMPL pose parameter loss, SMPL shape parameter loss, and geometric prior loss, of which the weights are 40, 20, 8, 0.6, and 0.16 respectively. The models are trained with 12 samples on each GPU.

TABLE 4
Comparison with 3D human body mesh recovery SOTAs on 3DPW dataset. The listed methods are all single-stage methods. All running times are reproduced with the official repository.

Method	Runtime ↓	MPJPE ↓	PMPJPE ↓	PVE ↓
Model-free methods				
METRO [28]	126 ms	77.1	47.9	88.2
FastMETRO [30]	109 ms	**73.5**	44.6	**84.1**
Model-based methods				
SPIN [77]	N/A	96.9	59.2	116.4
PARE [78]	156 ms	79.1	46.4	94.2
ROMP [33]	N/A	89.3	53.5	105.6
PyMAF-X [79]	152 ms	76.8	46.8	88.7
ours	**79 ms**	76.4	**41.1**	93.9

Settings for Pose Tracking. Following the setting of [47], we adopt DLA-34 [80] for the backbone. DLA-34 is a smaller network than ResNet-50, with only 15.7M parameters compared to 25.5M for ResNet-50. For PoseTrack2018, we resize the images to 512×512. We optimize the model with Adam [81] for 70 epochs. The initial learning rate is set to 1.25×10^{-4}, and it is reduced by 10 times after 60 epochs. The model is trained on 4 GPUs with 8 samples on each GPU. Design of the matching algorithm is the same as [47].

4.3 Comparison with State-of-the-art Methods

To illustrate the effectiveness of our method, we conduct experiments on 3 tasks and compare the results with state-of-the-art methods.

3D Multi-Person Pose Estimation. We compare with both top-down and bottom-up SOTA methods on CMU Panoptic and MuPoTS-3D datasets. The results on CMU Panoptic dataset are shown in Table 1. We utilize MPJPE to evaluate 3D pose estimation performance after root alignment. It can be discovered that our single-stage model outperforms bottom-up methods in MPJPE and achieves comparable accuracy with top-down methods. It's also worth mentioning that our single-stage model outperforms all of the listed methods in 2 of the 4 activity categories. The results on MuPoTs-3D dataset are listed in Table 2. We

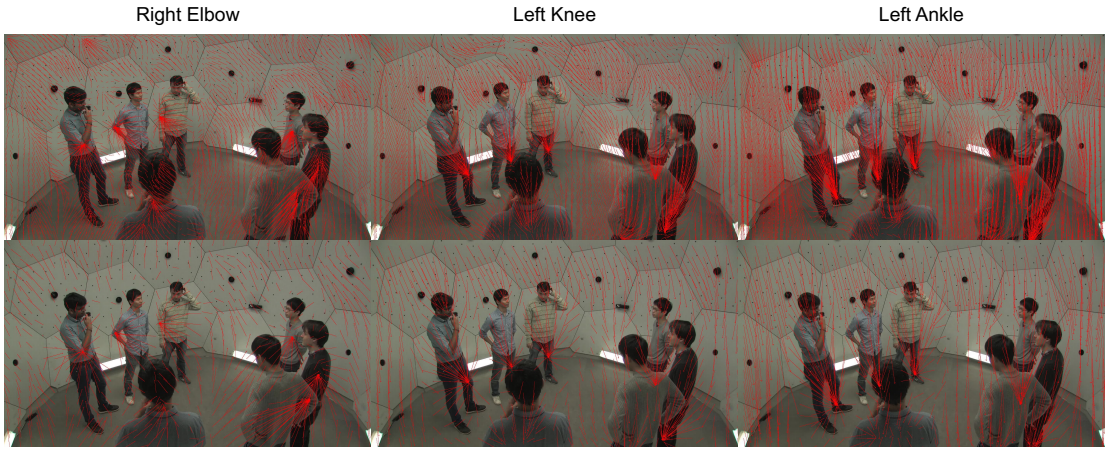


Fig. 4. Visualization of the joint offset maps by 2D projection. The top row is selected from the prediction maps with downsample stride 16 and the bottom row is selected from the prediction maps with downsample stride 32.

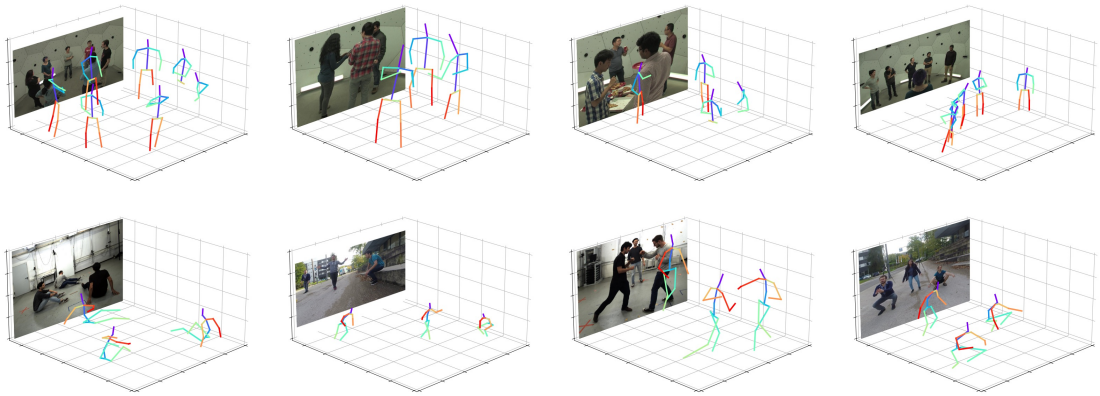


Fig. 5. Qualitative results of multi-person 3D pose estimation by our proposed DAS model. Best viewed in color.

adopt PCK_{rel} and PCK_{abs} as evaluation metrics. According to Table 2, our method outperforms all bottom-up methods and most of the top-down methods, especially on PCK_{rel}.

The above results show that our model can effectively estimate the human body structure. Compared to top-down methods, our model achieves comparable performance without the participation of a second stage for single-person pose estimation.

3D Multi-Person Mesh Recovery. We compare our method with previous single-stage SOTA methods. The compared methods include both model-free and model-based methods. Table 4 contains the detailed results. According to Table 4, our method outperforms all previous methods on PMPJPE. Meanwhile, our method achieves the best MPJPE among the model-based methods and surpasses most of the model-based methods on the PVE metric. The results indicate that our method can achieve considerable performance on human pose estimation while maintaining high accuracy on human body mesh recovery.

The results show that our proposed DAS can handle a wide range of geometric features. Besides, after adding regression targets, the original regression targets can still maintain high accuracy.

Multi-Person Pose Tracking. We compare our method with existing top-down and bottom-up methods on PoseTrack2018 validation set, as shown in Table 5. On both metrics, the best performance is with the top-down methods. Our method outperforms all bottom-up methods on joint detection metric AP and achieves comparable performance on joint tracking metric MOTA. This shows that our model is able to learn temporal features efficiently without using a feature extractor containing 3D convolution.

4.4 Ablation Study

To prove the effectiveness of our innovation, we conduct an ablation study on all of the 3 tasks.

3D Multi-Person Pose Estimation. The ablation study is conducted on the CMU Panoptic dataset. We implement our model with ResNet-50 [74] and FPN [48]. The results are shown in Table 6 and Table 7.

We first analyze the components of our model, as shown in Table 6. Compared with using the bounding box center as the human center, MPJPE of our model is improved by 2.8 mm when using the root joint as the human center. After introducing a recursive update strategy, MPJPE has been further improved by 4.9 mm. When modeling human body joints normalizing flow, the application of maximum likelihood estimation (MLE) during training boosts the localization capability, thus improving MPJPE by 3.7 mm. The combination of the recursive update strategy and MLE makes MPJPE achieve 56.3 mm, which exceeds the baseline by 7.2 mm. When equipped with a larger backbone, MSPN [75], our method achieves a better MPJPE of 54.4 mm.

Fig. 6. Visualization of mesh recovery results on the validation set of the 3DPW dataset.

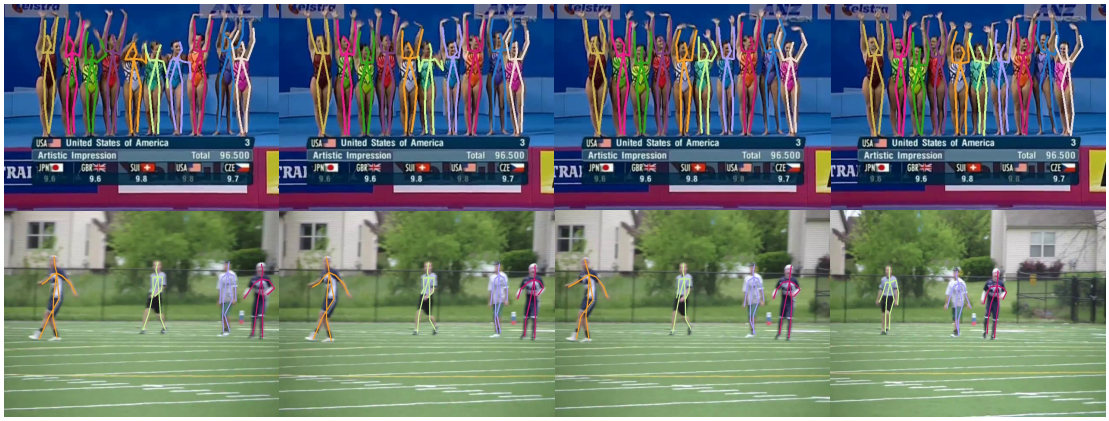


Fig. 7. Visualization of the pose tracking results. The images in the second row are four adjacent frames in the validation set. Each human body trajectory is represented by a unique color. Best zoom-in and view in color.

TABLE 5
Comparison with pose tracking methods on PoseTrack2018 dataset. † means that the running time is reproduced with the official repository.

Method	Runtime ↓	AP ↑	MOTA ↑
Raaj *et al.* [82]	N/A	70.4	60.9
Hwang *et al.* [83]	N/A	74.6	65.7
Wang *et al.* [70]	N/A	**81.5**	**68.7**
PGPT [84]	1197 ms †	76.8	67.1
UniTrack [85]	84 ms †	N/A	63.5
AlphaPose [86]	111 ms †	74.7	64.7
4DHumans [87]	1613 ms †	N/A	61.9
ours	**77 ms**	75.1	63.8

TABLE 6
Component analysis of our method on CMU Panoptic dataset.

Center Type	Recursive Update	MLE	Larger Backbone	MPJPE ↓
bbox				65.3
root				62.5
root	✓			57.6
root		✓		58.8
root	✓	✓		56.3
root	✓	✓	✓	**54.4**

TABLE 7
Comparison of recursive update settings on CMU Panoptic dataset.

Recursive Update	Multi-source Update	Update Layers	MPJPE ↓
			62.5
✓		1	58.6
✓	✓	1	58.2
✓	✓	2	57.9
✓	✓	3	**57.6**

We then compare different settings of recursive update module, as shown in Table 7. With application of recursive update strategy, MPJPE is improved by 3.8 mm compared to the baseline. By adopting multi-source update, MPJPE is further improved by 0.4 mm. Slight performance improvement can be obtained if stacking more update layers.

3D Multi-Person Mesh Recovery. For the mesh recovery task, we first study the impact of newly added SMPL parameter branch. Then we analyze the influence of each part of the loss function. The results are shown in Table 8. PVE metric is not applicable in the first row of table due to lack of branch which predicts SMPL parameters. According to Table 8, 3D pose estimation suffers from an accuracy decrease, which increasing task difficulty can adversely affect accuracy. By introducing our proposed recursive update strategy, all of the 3 metrics gain considerable improvement compared to the baseline. It's also worth mentioning that both MPJPE and PMPJPE outperform the model without an SMPL branch, which proves that normalizing flow is effective for a variety of tasks, and it can neutralize the impact of more difficult task.

Multi-Person Pose Tracking. We test the impact of each component added to the model on the accuracy of the pose tracking task. For the baseline model, we replace the displacement branch with a trivial greedy matching algorithm. This algorithm calculates the average joint distance between all human pairs between adjacent frames and then associates the two with the shortest distance. With the introduction of the center displacement branch, the accuracy of joint detection has decreased, while the accuracy of tracking has increased significantly. By using

the recursive update strategy, the accuracy drop of joint detection is compensated and improved compared to the baseline. By further supervising the displacement of each joint, the tracking performance can be further improved slightly, which indicates there is an accuracy trade between joint detection and tracking.

TABLE 8
Ablation study results for body mesh recovery on 3DPW dataset.

SMPL Branch	MLE	MPJPE ↓	PMPJPE ↓	PVE ↓
		79.1	43.7	N/A
✓		85.5	47.5	105.9
✓	✓	**76.4**	**41.1**	**93.9**

TABLE 9
Ablation study results for pose tracking on PoseTrack2018 dataset. * means that the association is implemented with a trivial greedy matching algorithm based on the nearest mean joint distance.

Center Displacement	MLE	Per-joint Displacement	AP ↑	MOTA ↑
			71.4	53.2*
✓			68.7	60.1
✓	✓		**75.9**	63.3
✓	✓	✓	75.1	**63.8**

4.5 Running Time Analysis

We measure the running time of our method on all of the 3 discussed tasks and compare it with SOTA methods.

For 3D human pose estimation, running time comparison with top-down and bottom-up SOTA methods is shown in Table 2. We implement our model with 2-stage MSPN [75], and the running time is measured on a single NVIDIA Tesla V100 GPU with batch size 1. For each method, we initially perform 10 inferences as a warm-up, and then average the time of the subsequent 100 inferences as the result. Compared with top-down methods, our method is approximately 1/3 faster while achieving comparable performance. Considering the small amount of people on MuPoTS-3D dataset (no more than 3 people per image on average), time consumption of top-down methods can increase rapidly in complex scenes. Compared with bottom-up methods, our method is faster while surpassing them in accuracy since high-resolution 2D or 3D heatmaps are not necessary.

We also measure the running time on human body mesh recovery and pose tracking tasks. The results are shown in Table 4 and Table 5. Since the newly added branch can run parallel to the existing branch, the inference time has not changed significantly. For both tasks, our method runs the fastest while achieving comparable accuracy to other SOTA methods, indicating the efficiency of our method.

4.6 Qualitative Analysis

Visualization of the distribution learned by normalizing flow is shown in Figure 8. According to the figures, the learned distribution is more flexible than prior distributions, which demonstrates the importance of the distribution-aware optimization scheme. Additionally, by introducing the recursive update strategy, the variances in the learned distribution are significantly reduced, indicating the recursive update strategy assists in accurately modeling the distribution of human body joints. We also visualize the learned joint offset maps by 2D projection in Figure 4. It can be found that by introducing recursive flow-based optimization, the model learns to regress the joint location either around the human body center or around the target at prediction maps. The prediction of long-distance joints can also benefit from more precise local predictions, which shows the effectiveness of the recursive update strategy. Results of 3D pose estimation are provided in Figure 5. According to the visualization, our proposed DAS model can accurately predict 3D human poses in various scenarios, e.g., pose changes, person truncation, and cluster backgrounds.

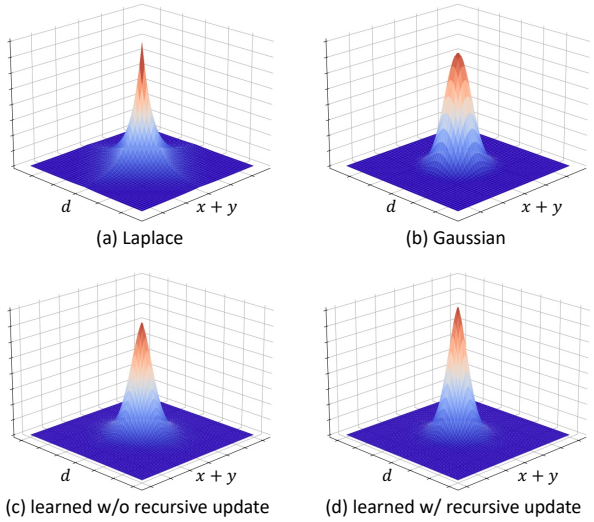


Fig. 8. Visualization of the learned distribution. It can be found that the learned distribution is more flexible than prior distributions, and the recursive update strategy can effectively reduce variances in distribution.

We visualize the results on 3DPW and PoseTrack2018 respectively to demonstrate the effectiveness of our method. As shown in Figure 6, our proposed method achieves good performance on the mesh recovery task. Analysis of the visualization results shows that our model gives accurate results in a variety of scenarios including complex poses, multi-person interactions and shooting angle changes, etc. The results of pose tracking are shown in Figure 7. The images in the first row show that our model performs well in crowded scenes. Meanwhile, we selected the results of four consecutive frames in the motion scene, which proves that our model can effectively associate the same target in the case of a large displacement between two adjacent frames.

5 Conclusion and Discussion

In this paper, we focus on the challenging multi-person 3D pose estimation task and propose a novel Distribution-Aware Single-Stage (DAS) model for the task. We further extend our proposed model to mesh recovery and pose tracking tasks to demonstrate the generality of our method. Dissimilar to two-stage methods, DAS is single-stage and can produce 3D pose estimations with a single forward pass. The DAS model predicts human center locations and center-relative joint offsets in parallel, and the 3D pose can be calculated from the predictions. In the same way, by adding

more output targets, the model can be extended to more tasks. The pipeline is simple and flexible, thus overcoming the drawbacks of previous methods on high computation cost and model complexity. Additionally, DAS introduces normalizing flow to model the distribution of human joints in the training phase, and a recursive update strategy is adopted to progressively refine the location distribution. In this way, DAS learns the true underlying distribution of joints, thus boosting the regression performance. We propose an efficient way to extend the model without making major changes to the model. By simply altering the input and adding branches, DAS can achieve considerable performance on extended tasks. Comprehensive experiments are conducted on multiple benchmarks and the effectiveness and efficiency of our proposed DAS are verified.

Limitation and future work. Our model has a limited ability to handle highly overlapping persons. Since DAS uses the combination of the human body center point and joint offset to represent the human body, when the human bodies are highly overlapped, the human body centers may also overlap, resulting in missed detection of the occluded human bodies. For the extended pose tracking task, DAS may have a large cumulative error when tracking for a long time. This is because DAS only learns the displacement between adjacent frames in the temporal dimension, and lacks global information. In the future, we plan to explore a more efficient way to represent global information and how humans interact, which is beneficial for understanding the scenes and solving the occlusion problem.

Acknowledgements. This research is supported in part by National Key R&D Program of China (2022ZD0115502), National Natural Science Foundation of China (NO. 62122010, U23B2010), Zhejiang Provincial Natural Science Foundation of China under Grant No. LDT23F02022F02, Key Research and Development Program of Zhejiang Province under Grant 2022C01082.

REFERENCES

[1] Z. Wang, X. Nie, X. Qu, Y. Chen, and S. Liu, "Distribution-aware single-stage models for multi-person 3d pose estimation," in *CVPR*, 2022, pp. 13 096–13 105.
[2] H. Belghit, A. Bellarbi, N. Zenati, and S. Otmane, "Vision-based pose estimation for augmented reality: a comparison study," *arXiv preprint arXiv:1806.09316*, 2018.
[3] H.-Y. Lin and T.-W. Chen, "Augmented reality with human body interaction based on monocular 3d pose estimation," in *ACIVS*. Citeseer, 2010, pp. 321–331.
[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*. Ieee, 2011, pp. 1297–1304.
[5] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman *et al.*, "Efficient human pose estimation from single depth images," *TPAMI*, vol. 35, no. 12, pp. 2821–2840, 2012.
[6] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *CVIU*, vol. 108, no. 1-2, pp. 52–73, 2007.
[7] Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *TIIS*, vol. 2, no. 1, pp. 1–28, 2012.
[8] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *CVPR*, 2017, pp. 7025–7034.
[9] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *ICCV*, 2017, pp. 2602–2611.
[10] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *ECCV*, 2018, pp. 529–545.
[11] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *3DV*. IEEE, 2018, pp. 120–130.
[12] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017, pp. 7291–7299.
[13] M. Fabbri, F. Lanzi, S. Calderara, S. Alletto, and R. Cucchiara, "Compressed volumetric heatmaps for multi-person 3d pose estimation," in *CVPR*, 2020, pp. 7204–7213.
[14] X. Nie, J. Feng, J. Zhang, and S. Yan, "Single-stage multi-person pose machines," in *ICCV*, 2019, pp. 6951–6960.
[15] F. Wei, X. Sun, H. Li, J. Wang, and S. Lin, "Point-set anchors for object detection, instance segmentation and pose estimation," in *ECCV*. Springer, 2020, pp. 527–544.
[16] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
[17] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, and C. Lu, "Human pose regression with residual log-likelihood estimation," in *ICCV*, 2021, pp. 11 025–11 034.
[18] J. P. Agnelli, M. Cadeiras, E. G. Tabak, C. V. Turner, and E. Vanden-Eijnden, "Clustering and classification through normalizing flows in feature space," *Multiscale Modeling & Simulation*, vol. 8, no. 5, pp. 1784–1802, 2010.
[19] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.
[20] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *TOG*, vol. 34, no. 6, pp. 1–16, 2015.
[21] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *ICCV*, 2015, pp. 3334–3342.
[22] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *ECCV*, 2018, pp. 601–617.
[23] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "Posetrack: A benchmark for human pose estimation and tracking," in *CVPR*, 2018, pp. 5167–5176.
[24] S. Liu, T. Hui, S. Huang, Y. Wei, B. Li, and G. Li, "Cross-modal progressive comprehension for referring segmentation," *TPAMI*, vol. 44, no. 9, pp. 4761–4775, 2021.
[25] T. Hui, S. Liu, Z. Ding, S. Huang, G. Li, W. Wang, L. Liu, and J. Han, "Language-aware spatial-temporal collaboration for referring video segmentation," *TPAMI*, 2023.
[26] Z. Yang, A. Zeng, C. Yuan, and Y. Li, "Effective whole-body pose estimation with two-stages distillation," *arXiv preprint arXiv:2307.15880*, 2023.
[27] T. Zhou, Y. Yang, and W. Wang, "Differentiable multi-granularity human parsing," *TPAMI*, 2023.
[28] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *CVPR*, 2021, pp. 1954–1963.
[29] ——, "Mesh graphormer," in *ICCV*, 2021, pp. 12 939–12 948.
[30] J. Cho, K. Youwang, and T.-H. Oh, "Cross-attention of disentangled modalities for 3d human mesh recovery with transformers," in *ECCV*. Springer, 2022, pp. 342–359.
[31] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *CVPR*, 2018, pp. 7122–7131.
[32] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3d human pose estimation in the wild," in *CVPR*, 2019, pp. 3395–3404.
[33] Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, "Monocular, one-stage, regression of multiple 3d people," in *ICCV*, 2021, pp. 11 179–11 188.
[34] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Ghum & ghuml: Generative 3d human shape and articulated pose models," in *CVPR*, 2020, pp. 6184–6193.
[35] B. Biggs, D. Novotny, S. Ehrhardt, H. Joo, B. Graham, and A. Vedaldi, "3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data," *NeurIPS*, vol. 33, pp. 20 496–20 507, 2020.
[36] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt, "Probabilistic monocular 3d human pose estimation with normalizing flows," in *ICCV*, 2021, pp. 11 199–11 208.

[37] B. Wandt, J. J. Little, and H. Rhodin, "Elepose: Unsupervised 3d human pose estimation by predicting camera elevation and learning normalizing flows on 2d poses," in *CVPR*, 2022, pp. 6635–6645.

[38] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *AAAI*, vol. 32, no. 1, 2018.

[39] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu, "Attentive fashion grammar network for fashion landmark detection and clothing category classification," in *CVPR*, 2018, pp. 4271–4280.

[40] Y. Xu, W. Wang, T. Liu, X. Liu, J. Xie, and S.-C. Zhu, "Monocular 3d pose estimation via pose grammar and data augmentation," *TPAMI*, vol. 44, no. 10, pp. 6327–6344, 2021.

[41] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *CVPR*, 2017, pp. 2011–2020.

[42] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "Arttrack: Articulated multi-person tracking in the wild," in *CVPR*, 2017, pp. 6457–6465.

[43] C. Gao, S. Liu, J. Chen, L. Wang, Q. Wu, B. Li, and Q. Tian, "Room-object entity prompting and reasoning for embodied referring expression," *TPAMI*, no. 01, pp. 1–16, 2023.

[44] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *CVPR*, 2018, pp. 350–359.

[45] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *arXiv preprint arXiv:1802.00977*, 2018.

[46] G. Ning, J. Pei, and H. Huang, "Lighttrack: A generic framework for online top-down human pose tracking," in *CVPR Workshops*, 2020, pp. 1034–1035.

[47] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *ECCV*. Springer, 2020, pp. 474–490.

[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.

[50] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[51] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *ICML*. PMLR, 2015, pp. 1530–1538.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, vol. 28, 2015.

[53] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019, pp. 9627–9636.

[54] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *ICCV*, 2019, pp. 10 133–10 142.

[55] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints," in *CVPR*, 2018, pp. 2148–2157.

[56] A.-I. Popa, M. Zanfir, and C. Sminchisescu, "Deep multitask architecture for integrated 2d and 3d human sensing," in *CVPR*, 2017, pp. 6289–6298.

[57] C. Wang, J. Li, W. Liu, C. Qian, and C. Lu, "Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation," in *ECCV*. Springer, 2020, pp. 242–259.

[58] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu, "Deep network for the integrated 3d sensing of multiple people in natural images," *NeurIPS*, vol. 31, 2018.

[59] J. Zhen, Q. Fang, J. Sun, W. Liu, W. Jiang, H. Bao, and X. Zhou, "Smap: Single-shot multi-person absolute 3d pose estimation," in *ECCV*. Springer, 2020, pp. 550–566.

[60] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[61] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3d human pose machines with self-supervised learning," *TPAMI*, vol. 42, no. 5, pp. 1069–1082, 2019.

[62] D. C. Luvizon, D. Picard, and H. Tabia, "Multi-task deep learning for real-time 3d human pose estimation and action recognition," *TPAMI*, vol. 43, no. 8, pp. 2752–2764, 2020.

[63] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *3DV*. IEEE, 2017, pp. 506–516.

[64] J. Zhang, D. Yu, J. H. Liew, X. Nie, and J. Feng, "Body meshes as points," in *CVPR*, 2021, pp. 546–556.

[65] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *TPAMI*, vol. 36, no. 7, pp. 1325–1339, 2013.

[66] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "Crowdpose: Efficient crowded scenes pose estimation and a new benchmark," in *CVPR*, 2019, pp. 10 863–10 872.

[67] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation." in *BMVC*, vol. 2, no. 4. Aberystwyth, UK, 2010, p. 5.

[68] ——, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*. IEEE, 2011, pp. 1465–1472.

[69] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014, pp. 3686–3693.

[70] M. Wang, J. Tighe, and D. Modolo, "Combining detection and tracking for human pose estimation in videos," in *CVPR*, 2020, pp. 11 088–11 096.

[71] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net++: Multi-person 2d and 3d pose detection in natural images," *TPAMI*, vol. 42, no. 5, pp. 1146–1161, 2019.

[72] J. Lin and G. H. Lee, "Hdnet: Human depth estimation for multi-person camera-space localization," in *ECCV*. Springer, 2020, pp. 633–648.

[73] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d motion capture with a single rgb camera," *TOG*, vol. 39, no. 4, pp. 82–1, 2020.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[75] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, "Rethinking on multi-stage networks for human pose estimation," *arXiv preprint arXiv:1901.00148*, 2019.

[76] Y. Wu and K. He, "Group normalization," in *ECCV*, 2018, pp. 3–19.

[77] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *ICCV*, 2019, pp. 2252–2261.

[78] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "Pare: Part attention regressor for 3d human body estimation," in *ICCV*, 2021, pp. 11 127–11 137.

[79] H. Zhang, Y. Tian, Y. Zhang, M. Li, L. An, Z. Sun, and Y. Liu, "Pymaf-x: Towards well-aligned full-body model regression from monocular images," *arXiv preprint arXiv:2207.06400*, 2022.

[80] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *CVPR*, 2018, pp. 2403–2412.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[82] Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh, "Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields," in *CVPR*, 2019, pp. 4620–4628.

[83] J. Hwang, J. Lee, S. Park, and N. Kwak, "Pose estimator and tracker using temporal flow maps for limbs," in *IJCNN*. IEEE, 2019, pp. 1–8.

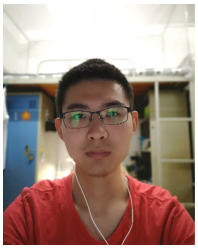
[84] Q. Bao, W. Liu, Y. Cheng, B. Zhou, and T. Mei, "Pose-guided tracking-by-detection: Robust multi-person pose tracking," *TMM*, vol. 23, pp. 161–175, 2020.

[85] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. Torr, and L. Bertinetto, "Do different tracking tasks require different appearance models?" *NeurIPS*, vol. 34, pp. 726–738, 2021.

[86] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *TPAMI*, 2022.

[87] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, "Humans in 4d: Reconstructing and tracking humans with transformers," *arXiv preprint arXiv:2305.20091*, 2023.

Leyan Zhu is currently an M.Eng. candidate at Institute of Artificial Intelligence, Beihang University. He received the B.Eng. degree from Beihang University. He has won the Best Paper Award of ACM MM 2021 and the Best Video Award of IJCAI 2021 Video Competition. His research interest is human pose estimation.



Zitian Wang is currently a Ph.D. candidate from Computer Science and Engineering, Beihang University. He received the Bachelor's degree and Master's degree from Beihang University. His research interests include 3D object detection, 3D human pose estimation and action recognition.

Si Liu is currently a full Professor at Beihang University. She received her Ph.D degree from Institute of Automation, Chinese Academy of Sciences. She has been a research assistant and postdoc in National University of Singapore. Her research interests include computer vision and multimedia analysis. She has published over 40 cutting-edge papers on vision-language understanding, image/video segmentation hand human parsing, etc. She was the recipient of the National Science Fund for Excellent Young Scholars. She has won the Best Paper Awards of ACM MM 2021 and 2013, the Best Demo Award of ACM MM 2012. She was the Champion of CVPR 2017 Look Into Person Challenge and the organizer of ECCV 2018, ICCV 2019 and CVPR 2021 Person in Context Challenges.

Xuecheng Nie is a senior researcher at MT Lab, Meitu Inc. His research interest focus on computer vision and deep learning, specially, their applications on human pose estimation, action recognition, image generation and synthesis, etc. He obtained the Ph.D. degree in 2020 at Learning and Vision Lab of Electrical and Computer Engineering Department, National University of Singapore, advised by Associate Professor Jiashi Feng and Shuicheng Yan. He achieved both his bachelor and master degrees from Tianjin University under the supervision of Professor Wei Feng.

Luoqi Liu is currently Vice President of Meitu Inc and Director of Meitu Image & Vision Lab. He is mainly working on multimedia, graphics and computer vision including video understanding, image editing and augmented reality. He got his Ph.D. from National University of Singapore (NUS), advised by Associate Professor Shuicheng Yan. Till now, he has published over 30 papers in top international journals and conferences, such as CVPR, NeurIPS, ECCV, ICCV, and TPAMI. He also received ACM Multimedia Best Paper Award 2013 and PREMIA Best Student Paper Gold Prize 2014.

Bo Li received the B.S. degree in computer science from Chongqing University, China, the M.S. degree in computer science from Xian Jiaotong University, China, and the Ph.D. degree in computer science from Beihang University, China. He joined the School of Computer Sicence and Engineering, Beihang University. He has published more than 100 academic papers in diverse research fields, including intelligent perception, big data intelligence, remote sensing image fusion, and intelligent hardware.