Joint-Limb Compound Triangulation With Co-Fixing for Stereoscopic Human Pose Estimation

Zhuo Chen, Xiaoyue Wan, Yiming Bao, and Xu Zhao, *Member, IEEE*

***Abstract*—As a special subset of multi-view settings for 3D human pose estimation, stereoscopic settings show promising applications in practice since they are not ill-posed but could be as mobile as monocular ones. However, when there are only two views, the problems of occlusions and "double counting" (ambiguity between symmetric joints) pose greater challenges that are not addressed by previous approaches. On this concern, we propose a novel framework to detect limb orientations in field form and incorporate them explicitly with joint features. Two modules are proposed to realize the fusion. At 3D level, we design *compound triangulation* as an explicit module that produces the optimal pose using 2D joint locations and limb orientations. The module is derived from reformulating triangulation in 3D space, and expanding it with the optimization of limb orientations. At 2D level, we propose a parameter-free module named *co-fixing* to enable joint and limb features to fix each other to alleviate the impact of "double counting." Features from both parts are first used to infer each other via simple convolutions and then fixed by the inferred ones respectively. We test our method on two public benchmarks, Human3.6M and Total Capture, and our method achieves state-of-the-art performance on stereoscopic settings and comparable results on common 4-view benchmarks.**

***Index Terms*—Human pose estimation, triangulation, machine learning.**

I. Introduction

3D HUMAN Pose Estimation (3D HPE) is a fundamental and important task in multimedia, which aims to locate anatomy key points of human body in 3D space. Due to its wide application in intelligent medicare [1], action recognition [2], sports [3], human-computer interaction, etc., 3D HPE has drawn great attention in the last decades.

Recently, multi-view HPE methods have shown great advance in estimation performance [1], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], most of which follow the methodology to first detect 2D key-points and then calculate 3D poses via triangulation frameworks [16]. Compared to single RGB images [17], [18], [19], multi-view settings effectively eliminate depth uncertainty, thus leading to more plausible and accurate poses. However, most multi-view settings require enough indoor space, careful installment, and accurate calibrations. Such conditions could sometimes be cumbersome and impractical.

Actually, to eliminate depth uncertainty, the number of cameras can be as small as 2, which is known as stereo [20]. Stereo systems are much easier to set up and calibrate than those with more views. They could even be made portable to fit in limited room or outdoor scenes. In contrast to their promising applications, explorations on them are however quite limited. The sparsity of views magnifies some problems that could not be well addressed by current multi-view HPE methods. Firstly, as a long-standing problem, occlusions are typically tackled by cross-view feature fusion [6], [8], [10], [21] or learnable weights [4]. However, these methods work under the assumption that visible views are enough to locate the joint, which is apparently invalid in stereo settings. Secondly, as an innate problem in learning-based 2D human pose estimation, "double counting" (*i.e.*, the ambiguity between symmetric joints) [22] also impedes accurate joint locations. Though selecting view subsets to generate hypotheses presents a promising solution [14], it is impossible under stereoscopic settings as both views are necessary for unique joint locations.

The key to the above problems is to design and detect features that encode different aspects other than 2D joint locations and provide necessary information for monocular 3D pose estimation. Therefore, the features should focus on body parts between joints, *i.e.*, limbs, and indicate *3D limb orientations*. Fig. 1 shows where these features are located and to what extent they can help reconstruct a 3D pose. The orientations eliminate the depth ambiguity between joints so features from merely one view are enough for reconstructing relative 3D poses. If more than one view is available, then the extra information helps refine the 3D poses by posing an well-determined setting.

According to our trial experiments, limb orientation regression should be combined with positional implications for better convergence, so limb fields are suitable descriptors. Previous methods have provided detailed studies on the application of such fields in HPE [17], [18], [19], [23], [24], [25]. They first prevailed as 2D fields named Part Affinity Fields (PAFs) in OpenPose [23], [24]. Inspired by PAFs, some recent monocular 3D HPE methods [17], [18], [19], [25] utilize similar fields to imply limb orientations or depths and have validated the

Manuscript received 7 December 2023; revised 7 March 2024 and 3 May 2024; accepted 25 May 2024. Date of publication 6 June 2024; date of current version 14 November 2024. This work was supported in part by the NSFC under Grant 62176156 and in part by the Medical Engineering Cross Research Fund of Shanghai Jiao Tong University under Grant YG2023ZD12. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wei-Ta Chu. *(Corresponding author: Xu Zhao.)*

The authors are with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: chzh9311@sjtu.edu.cn; sherrywaan@sjtu.edu.cn; yiming.bao@sjtu.edu.cn; zhaoxu@sjtu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMM.2024.3410514, provided by the authors.

Digital Object Identifier 10.1109/TMM.2024.3410514

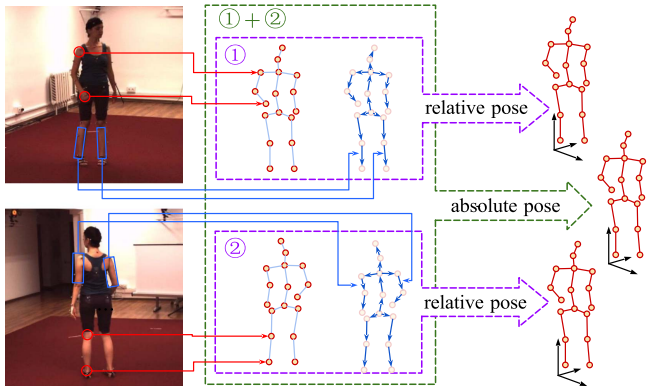

Fig. 1. Illustration of feature sources and estimation results. Blue rectangles imply the focus of limb features, which are different from joint features shown in red circles. With the 2D joint locations and 3D limb orientations from merely one view, we can estimate the 3D poses relative to scale change, as in ① and ② separately. If the features from both views are available, then it becomes ① + ② and is well-determined, leading to the estimation of an absolute 3D pose.

effects of such features. In this work, we describe limb features using smoothed Limb Orientation Fields (LOFs), which provide point-level orientation predictions over the target limb. Based on LOFs, we can take a closer look at the aforementioned problems and build the framework shown in Fig. 2.

In the proposed framework, occlusions are solved by fusing limb orientations and joint locations naturally as indicated in Fig. 1. The fusion module, known as *compound triangulation*, is an explicit and differentiable function of detected joint and limb features, allowing the whole framework to be trained end-to-end. To derive it, we review traditional linear triangulation [26] in 3D space and model both the re-projection error of joint positions and error of limb estimations in 3D Euclidean space. The function is simply the solution of minimizing the sum of the two error terms. Moreover, we add learnable weights to lower the influence of occluded views, so that the result is derived from visible features and is therefore more reliable.

Since compound triangulation incorporates features after the regression of heatmaps, it cannot filter out points influenced by "double counting." Therefore, we propose *co-fixing* module to utilize joint and limb features to fix each other at heatmap level. The general process is to first fix LOFs by multiplying fields inferred from joint confidence maps, and then fix joints by inferred maps from LOFs. The bi-directional inference is done by convolutions with a carefully designed kernel, so the procedure is simple and computationally efficient. Essentially, co-fixing module utilizes neighboring joint and limb estimations to correct the current joint. By taking a broader range of body parts into consideration, the randomly mixed symmetric joints tend to be distinguished.

We conduct experiments on Human3.6M and Total Capture Datasets, both on common 4-view settings and stereoscopic settings. Compared to previous methods, our method achieves $\geq 3.6\%$ error drop in stereoscopic scenes, which aligns with our goal. On common 4-view benchmarks, the result of our method is also comparable to previous methods. We also report a detailed analysis to explore the principle in stereo performance promotion and analyze the effect of every submodule.

In all, our contributions include:

1) We propose compound triangulation, an algebraic fusion method for multi-view joint position and limb pose estimations. It explicitly incorporates Limb Orientation Field to multi-view 3D pose estimation.
2) We propose co-fixing module which leverages limb and joint predictions to fix each other bi-directionally, as well as the rules to filter out negative fixes.
3) We design and conduct experiments on stereoscopic scenes in Human3.6M and Total Capture datasets and our framework achieves state-of-the-art result.

The rest of the paper is organized as follows: Section II provides a literature review of studies related to human pose estimation. Section III describes our method, and Section IV discusses the relative technical details. In Section V, experiment settings, results and corresponding analysis are reported. Finally, Section VI draws a conclusion and indicates future work.

II. Related Work

In this section, we first review general multi-view HPE methods, then present works on limb features and refining methods that apply similar conceptions as ours.

A. 3D HPE From Multi-View Images

Early trials on multi-view 3D human pose estimation are mostly segmentation-based, where hand-crafted features based on body silhouettes and textures are used [27], [28], [29], [30]. The pose was generated by optimizing a parametric model using probabilistic analysis and only achieved limited performance. As deep learning largely promotes 2D pose estimation [22], [31], [32], [33], [34], the dominant framework of multi-view HPE gradually shifts to a two-staged procedure: First estimate 2D poses from each view, then leverage them to 3D space by fusing multi-view estimations via geometric methods [4], [5].

On that basis, recent attempts to further promote 3D HPE mainly focus on sufficiently exploiting the complementary relationships between different views. Such methods include cross-view feature fusion [6], [8], [10], [21], introducing learnable weights [4], [14], and utilizing volumetric representations [4], [9], [15]. Though proved to be effective in dealing with occlusions on public benchmarks, they are not well suited for stereo scenes as only one view is not enough to complement the other occluded one. Our method overcomes the problem by exploiting features between joints and makes better predictions under stereoscopic settings.

B. Limb Features in Human Pose Estimation

The features on anatomical key points, *i.e.* joints, are already well-studied, but the features over limbs are still under exploration. OpenPose [23], [24] proposes Part Affinity Field (PAF) to describe the connections between joints, which indicates that features in between are somehow available. After the success of OpenPose, plenty of works utilize similar frameworks to solve various problems [3], [35], [36]. OpenPose focuses on bottom-up multi-person pose estimation, where the connections

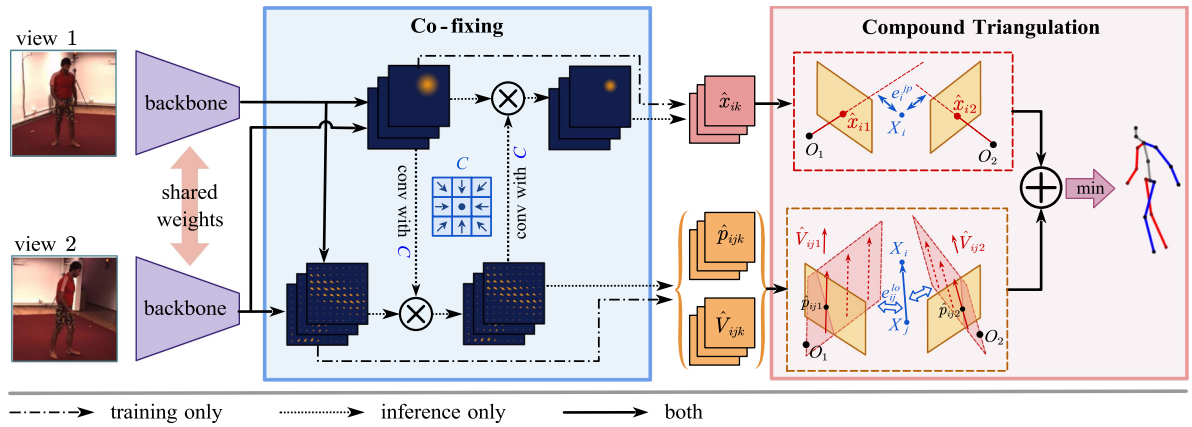

Fig. 2. The overall framework to detect and fuse limb orientations in multi-view 3D human pose estimation. A 2D backbone is first utilized to detect joint confidence maps and LOFs. In co-fixing module, the previously derived joint and limb feature maps get fixed by the inferred feature maps from each other. Note that this stage is for post-processing and merely activated in testing. For joint i and limb (i, j) on camera k, joint positions x_{ik}, limb positions p_{ijk} and orientations n_{ijk} are calculated from previous heatmaps, and then fed into *compound triangulation*. It is an optimization of the aggregation of error terms over joints (e^{jp}_{ik}, blue errors in the upper image) and limbs (e^{lo}_{ijk}, blue hollow errors in the lower image).

between joints are not implied by joint labels so PAF is necessary and effective.

Later, those features were found capable of encoding depth so similar conceptions were applied to 3D reconstruction. For limb features, two types of fields are extracted: 3D orientation fields [17], [18], [25] used to extract limb orientations, and depth maps [19] used to indicate explicit depths of the body parts. Both achieved improvement on monocular 3D HPE over previous methods. Such ideas can be transferred to multi-view settings, but little attention has yet been paid probably because joint features are already sufficient to solve a pose. But this could be wrong when occlusions occur, especially in stereo. Moreover, our method proves that incorporating limb features can also benefit general multi-view scenes.

C. 3D and 2D Human Pose Refinement

Some recent works try to refine pose in 3D space, generally by fusing visual features with other modalities like IMUs [7], [13] and pose priors [11], [21]. Though similar to our methods, they emphasize too much on the non-vision counterpart due to its certainty compared to visual estimations. To cope with the uncertainty of pose priors, Pictorial Structure Models (*i.e.* PSM) [21], [37], [38] are used to build a probabilistic optimization problem, and the optimal pose is derived from searching in the feasible pose region. However, the searching procedure is non-differentiable and computationally expensive, yet our method is free from these drawbacks.

Since 3D poses are based on 2D detections, refining 2D pose estimations is also important for HPE tasks. Though stacked structures [22], [31], [33] are proved effective in solving joint displacement or "double counting" [39], they cannot cover all circumstances due to the variety of body poses. Recently Ke et al [40] proposes strcture-aware loss to strenthen the matching of keypoints. Kamel et al [41] propose pose correction branch (CNet) to allow for larger corrections. These methods successfully encode the innate connections from training data, yet our co-fixing method provides a parameter-free solution that leads to good performance.

III. Method

Fig. 2 shows the framework of our method. In this section, we first list the main procedures, and then describe the technical details. We follow the order of training - inference so the inference-only module, co-fixing, is introduced at last.

1) *Limb Orientation Field (LOF):* LOFs are 2D maps composed of 3D vectors indicating the orientation of the target limb. They are estimated along the joint confidence maps using a shared backbone.
2) *Parameter regression:* The 2D joint locations, 2D limb positions and 3D limb orientations are regressed from the joint confidence maps and LOFs. The regression is done by soft-argmax and weighted average.
3) *Compound triangulation:* The regressed parameters are fed into a closed-form triangulation function to produce the optimal pose. The function, named Compound Triangulation, is the solution to a compound minimization of joint and limb estimation errors.
4) *Co-fixing during inference:* While inference, the joint confidences are used to infer limb features, and then the latter fix the original LOFs via multiplication. LOFs are meanwhile used to fix joint estimations in the same way.

A. Definition of Limb Orientation Field

Limb Orientation Field (LOF) is designed to indicate two aspects of a limb: the orientation in 3D space and position on 2D image plane. These vectors are densely distributed local regressers like PAF [23] and POF [17]. Each LOF vector of one limb represents the orientation with its own, so it points from

one end joint of the limb to the other. It also predicts the confidence that it is on the limb by its norm. So similar to joint confidence maps, the vector gets unit length when on the 2D limb, and shortens as it is located away from the target.

Suppose $X_{ik}, X_{jk} \in \mathbb{R}^3$ are the 3D positions of adjacent joints $i, j \in [1, n^j]$ under the local coordinate system of camera $k \in [1, n^c]$, and $x_{ik}, x_{jk} \in \mathbb{R}^2$ are their projections. Then the ground truth direction of LOF is defined as $V_{ijk} = (X_{ik} - X_{jk})/\|X_{ik} - X_{jk}\|$. Meanwhile, the vector norms, referred to as *norm multipliers*, are defined as a Gaussian mapping of the distance to the 2D limb line segment. Suppose a feature map size of $H \times W$. Use $D = \{[u, v]^T \in \mathbb{N}^2 | 0 \le u < W, 0 \le v < H\}$ to represnet the point set of the feature map, then $\forall x \in D$, the distance is

$$d_{ijk}(x) = \begin{cases} \|x - x_{ik}\|, & \text{if } v_{ijk}^T(x - x_{ik}) < 0; \\ \|x - x_{jk}\|, & \text{if } v_{ijk}^T(x - x_{jk}) > 0; \\ |v_{ijk\perp}^T(x - x_{ik})|, & \text{otherwise.} \end{cases} \quad (1)$$

where $v_{ijk} \in \mathbb{R}^2$ is the unit vector pointing from x_{ik} to x_{jk} on the image plane, and $v_{ijk\perp} \in \mathbb{R}^2$ is the unit vector perpendicular to v_{ijk}.

The norm multiplier is then Gaussian mapping of the distance in (1). With a predefined deviation σ, the ground truth LOF (represented by $F_{ijk}^{lo} \in \mathbb{R}^{H \times W \times 3}$), is generated by refining vector norms via the following formula:

$$F_{ijk}^{lo}(x) = V_{ijk} \exp\left\{-\frac{d_{ijk}(x)^2}{\sigma^2}\right\}. \quad (2)$$

Compared to PAF, LOF appends an extra dimension perpendicular to the image plane, allowing it to encode depth information. If only the first two dimensions are considered, then it becomes the same form as a PAF. We name this *PAF-subset* of the LOF, represented by $F_{ijk}^{pa} \in \mathbb{R}^{H \times W \times 2}$.

B. Parameter Regression From Heatmaps

In the proposed framework, 2D backbone outputs a set of joint confidence maps H_{ik}^{jp} and limb orientation fields F_{ijk}^{lo}. Necessary parameters can be regressed from them, including 2D joint positions, 2D limb positions and 3D limb orientations.

The joint position $\hat{x}_{ik}$ is predicted by the max values location the heatmap. Recent methods [4] have proved soft-argmax function [42] to be a proper approximation for end-to-end training, so this function is utilized in this stage:

$$\hat{x}_{ik} = \frac{\sum_{x \in D} x \exp\{\beta H_{ik}^{jp}(x)\}}{\sum_{x \in D} \exp\{\beta H_{ik}^{jp}(x)\}}. \quad (3)$$

where β is a predefined "inverse temporature" to adjust the output.

Similarly, limbs are modeled as straight lines. As all vectors in LOF predict the same orientation, it is natural to aggregate the result via weighted average. Once the limb orientation $\hat{V}_{ijk}$ is known, the limb position is indicated by an arbitrary point $\hat{p}_{ijk}$ on the limb, which can also be calculated from the norm multiplier via soft-argmax like joint positions.

$$\hat{V}_{ijk} = \sum_{x \in D} \|F_{ijk}^{lo}(x)\| F_{ijk}^{lo}(x). \quad (4)$$

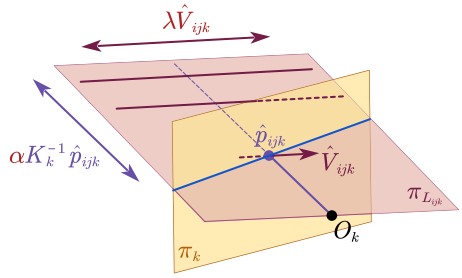

Fig. 3. Illustration of spatial co-relations between image plane (π_k in orange), camera center O_k, the regressed position $\hat{p}_{ijk}$ and orientation $\hat{V}_{ijk}$, and the plane containing the line set L_{ijk} defined by (6) ($\pi_{L_{ijk}}$ in red). $O_k, \hat{V}_{ijk} \in \pi_{L_{ijk}}$. $l_{ijk} \in \pi \cap \pi_{L_{ijk}}$ is the estimated limb projection line and $\hat{p}_{ijk} \in l_{ijk}$ is the actual parameter regressed from LOF. We can see where lines in L_{ijk} are located in space.

$$\hat{p}_{ijk} = \frac{\sum_{x \in D} x \exp\{\beta \|F_{ijk}^{lo}(x)\|\}}{\sum_{x \in D} \exp\{\beta \|F_{ijk}^{lo}(x)\|\}}, \quad (5)$$

The projection from a single view limits the limb pose to a line set shown in Fig. 3. Assume K_k represents the camera intrinsic, then this line set is:

$$L_{ijk} = \left\{\alpha K_k^{-1}\hat{p}_{ijk} + \lambda\hat{V}_{ijk}, \forall\lambda \in \mathbb{R} | \forall\alpha > 0\right\}. \quad (6)$$

Notably, if regarded as a weighted average of unit direction vectors, (4) is actually not weighted by the norms, but by their squares. We will discuss this special design in Section IV-A.

C. Compound Triangulation

The process to derive the most likely 3D position from 2D estimations in different views is usually known as *triangulation*. Minimizing re-projection error is a common methodology. For triangulation on points, the optimal solution is already proposed [16], and the linear versions [26] are widely used. However, re-projection error is not preferrable for limbs because it drops the necessary depth feature. Therefore, the compound objective function of joints and limbs is reconsidered in 3D space.

We first review linear triangulation on joints in 3D space, which is achieved via linearizing reprojection error. Suppose $d_k(X_i)$ is the distance from joint X_i to the image plane k and $\hat{x}_{ik}$ is the 2D estimation, then the error is reformulated as the distance between the joint and the re-projection line along the image plane (See Section IV-B for more details):

$$e_{ik}^{jp} = \|X_i - d_k(X_i)K_k^{-1}\hat{x}_{ik}\|^2. \quad (7)$$

Similar definition can be applied to limbs with some modifications. A limb is modeled as a line segment connecting joints X_i and X_j, *i.e.*, a point set. The projection limits the limb to a line set L_{ijk} (defined by (6)). One simple definition is the minimum distance between two elements from both sets, but it finally becomes the distance between the near extremity and the plane of L_{ijk} ($\pi_{L_{ijk}}$ in Fig. 3). The other one is totally unmeasured. To tackle this, we calculate the minimum distance between the two

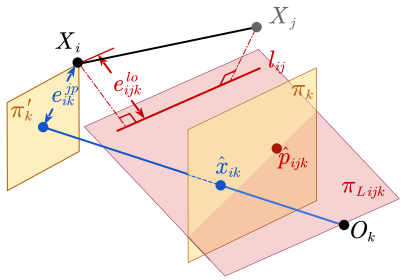

Fig. 4. Two types of error terms, e^{jp}_{ik} and e^{lo}_{ijk}, in compound triangulation under one view. X_i is the target joint position. $\hat{x}_{ik}$ and $\hat{p}_{ijk}$ are the estimated 2D position of joint i and position indicator of limb (i,j) on image plane π_k. $\pi'_k /\!\!/ \pi_k$ and $X_i \in \pi'_k$. Then e^{jp}_{ijk} is the distance between X_i and re-projection line along π'_k, while e^{lo}_{ik} is the distance from X_i to $l_{ijk} \in L_{ijk}$ with the least distance to X_i and X_j.

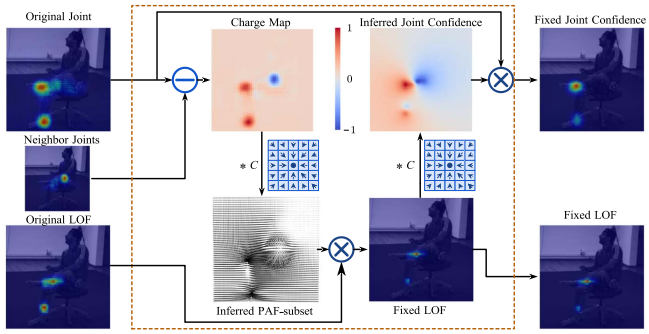

Fig. 5. Illustration of the co-fixing algorithm. The algorithm starts with the subtraction of the target joint confidence map and one of its neighbors. Then the map is convolved by C to get an inferred field, which is applied to the original LOF with a mixed multiplication shown as (16). If the fixing is validated by predefined rules, the fixed LOF is convolved by C to produce th inferred joint confidence. Otherwise, the original LOF will be used. Finally, the inferred joint confidence maps from all neighbors are multiplied to the initial target heatmap to get it fixed.

extremities and an arbitrary line in L then sum them up.

$$e^{lo}_{ijk} = \min_{\alpha,\lambda_i,\lambda_j} \sum_{t\in\{i,j\}} \|X_t - \left(\alpha K_k^{-1}\hat{p}_{ijk} + \lambda_t \hat{V}_{ijk}\right)\|^2, \quad (8)$$

where two distance terms share parameter α because a common line from L is used, but have independent parameter λ as the pedals are different. Detailed expression of e^{lo} is available in Supp. I.A. Fig. 4 shows the two error terms in 3D space.

Finally, we iterate (7) over all joints and (8) over all limbs and sum the results up to get the final optimization problem:

$$\min_{X_i, 1\le i\le n^j} \sum_{k=1}^{n^c} \left(\sum_{i=1}^{n^j} w^{jp}_{ik} e^{jp}_{ik} + \sum_{(i,j)\in\mathcal{E}} w^{lo}_{ijk} e^{lo}_{ijk} \right), \quad (9)$$

where w^{jp}_i and w^{lo}_i are learnable weights and $\mathcal{E}$ is the set of connected joint pairs ($|\mathcal{E}| = n^l$). In (9), every single term is quadratic so it is a quadratic optimization problem. We can trivially get its solution as a closed-form differentiable function that is applied to end-to-end training

$$\hat{X}_{[1:n^j]} = f\left(\hat{x}, \hat{V}, \hat{p}, w; P\right). \quad (10)$$

Among all the inputs, the predicted joint positions $\hat{x}$, limb orientations $\hat{V}$ and positions $\hat{p}$, and weights w are backbone outputs and receive gradients in back-propagation, while the camera parameters P are constant. Additionally, the most computationally complex calculation in f is the inverse of a $3n^j \times 3n^j$ matrix, so the complexity is $O(n^{j3})$. It is called *compound triangulation* given that it combines the triangulation over points and limbs.

D. Training Process and Loss Functions

Generally, we apply a two-staged training process: 1. Pre-training the 2D backbone using the two-branch outputs; 2. End-to-end training using 3D poses.

The backbone outputs predicted joint heatmaps H^{jp}_{ik} along with LOFs F^{lo}_{ijk}. We do backbone pre-training using MSE loss on both outputs before the model is trained end-to-end.

With the two maps, the estimated joint position $\hat{X}_i$ is derived via compound triangulation. With known ground truth joint positions X^{gt}_i, mean per joint position error, *i.e.* MPJPE, is generally a suitable loss function but may be unnecessarily sensitive to outliers considering the inverse algebra in optimizing (9). To weaken the influence of outliers, we utilize the soft version in [4] with ε =20 mm in experiment:

$$\mathcal{L}^{jp}(X_i) = \begin{cases} \|X_i - X^{gt}_i\|^2, & \text{if } \|X_i - X^{gt}_i\|^2 < \varepsilon^2 \\ \|X_i - X^{gt}_i\|^{0.2}\varepsilon^{1.8}, & \text{otherwise} \end{cases} \quad (11)$$

Aside from the final output, we also supervise the pose calculated from only 2D poses by linear triangulation X^{lt}_i because we find it leads to a better result.

However, the above supervisions do not ensure the convergence of LOF predictions. So we introduce another loss to regularize the pointwise orientation of LOF on limb (i,j), using the ground truth direction vector V^{gt}_{ijk}. Pointwise vector norms $m^{lo}_{ijk}(x) = \|F^{lo}_{ijk}(x)\|$ are used for normalization.

$$\mathcal{L}^v(F^{lo}_{ijk}) = \frac{\sum_{x\in D} \left\| m^{lo}_{ijk}(x) V^{gt}_{ijk} - F^{lo}_{ijk}(x) \right\|^2}{\sum_{x\in D} m^{lo}_{ijk}(x)^2} \quad (12)$$

The final loss is:

$$\mathcal{L} = \frac{1}{n^j}\sum_{i=1}^{n^j} \left(\mathcal{L}^{jp}(\hat{X}_i) + \mathcal{L}^{jp}\left(\hat{X}^{lt}_i\right) \right) + \frac{\mu}{n^l n^c} \sum_{\substack{(i,j)\in\mathcal{E} \\ 1\le k\le n^c}} \mathcal{L}^v\left(F^{lo}_{ijk}\right) \quad (13)$$

E. Co-Fixing Between Joints and Limbs

The double counting problem is usually not consistent in joint predictions and their neighbors. Namely, one joint with an ambiguous prediction can be contiguous to a clearly located limb or parent/child joint and vice versa, as shown in Fig. 5. This fact implies the viability of enhancing both joint and limb predictions at heatmap level based on each other. In practice, limb and

joint predictions could infer each other and the fixing is simply the multiplication of inferences and predictions. We refer to this algorithm as *co-fixing*.

1) Fixing Via Inferred Feature Maps: We first clarify the formulation of predictions. As co-fixing is applied on 2D image planes, only the first two dimensions of LOFs, *i.e.*, the PAF-subset $F^{pa}_{ijk} \in \mathbb{R}^{H\times W\times 2}$, is needed. Meanwhile, the predicted joint confidence maps are represented by $H^{jp}_{ijk} \in \mathbb{R}^{H\times W}$.

In practice, to control the memory usage, all the conversions are realized by simple 2D convolutions with a common kernel $C \in \mathbb{R}^{(2h-1)\times(2w-1)\times 2}$. The kernel is defined by $C(x) = ([w,h]^T - x)/\|[w,h]^T - x\|^{(1+\gamma)}$, where γ is the predefined fading factor. The inferred PAF-subsets $\hat{F}^{pa}_{ijk}$ from joints i and j and joint confidence maps $\hat{H}^{jp}_{ijk}$ from limb (i,j) are

$$\hat{F}^{pa}_{ijk} = \left(H^{jp}_{ik} - H^{jp}_{jk}\right) * C, \tag{14}$$

$$\hat{H}^{jp}_{ijk} = F^{pa}_{ijk} * C. \tag{15}$$

By sliding the kernel C over the target feature map of size $H \times W$ while zero-padding the boundaries, the center of the kernel traverses all positions inside. Note that the detailed calculations in (14) and (15) are different, depending on the target dimensions. For a joint confidence map of $H \times W$ in (14), the unit computation is multiplying values in the map to the corresponding vectors in the kernel. In this way, vectors in the resulting 2D field tend to point to or against joint positions. For a PAF-subset of $H \times W \times 2$ in (15), however, the unit computations are dot products. Thus, the more vectors point to one location, the larger response the location will manifest.

The next step is to generate fixing factor matrices from the inferred heatmaps to refine the original predictions via multiplication. For LOFs, the fixing factor matrices are the element-wise dot products of the original PAF-subsets and the inferred ones. For joints, the factor matrices are simply the inferred joint confidence maps. Use "$\circ$" to stand for element-wise numeric product, and "$\cdot$" for element-wise dot product, then the fixing process could be described as

$$F^{lo\prime}_{ijk} = \left(\hat{F}^{pa}_{ijk} \cdot F^{pa}_{ijk}\right) \circ F^{lo}_{ijk} \tag{16}$$

$$H^{jp\prime}_{ik} = H^{jp}_{ik} \circ \hat{H}^{jp}_{ij_1\,k} \circ \hat{H}^{jp}_{ij_2\,k} \circ \cdots \circ \hat{H}^{jp}_{ij_m k} \tag{17}$$

where $j_1, j_2, \ldots, j_m$ are all adjacent joints of i.

2) Rules to Filter Out Potentially Negative Fixes: The above fixing can sometimes be harmful, e.g. when the original predictions are accurate while their neighboring limbs and joints (which are used in co-fixing) are not. To avoid this defect, we leverage some rules to decide whether to apply this fix or not.

To start with, we generate all possible combinations by whether to fix or not under all views using a predefined score function f. Use x_k to stand for the original prediction of either one joint position or limb pose on view k, and x'_k for its correction. For convenience, we use $\delta_k \in \{0,1\}$ to represent the choice between x_k and x'_k, *i.e.*, $y_k = (1-\delta_k)x_k + \delta_k x'_k$. Then a specific combination is represented by a binary number $\delta_1\delta_2\cdots\delta_{n^c(2)}$ and its score is

$$S_{\delta_1\delta_2\cdots\delta_{n^c}} = f(y_1, y_2, \ldots, y_{n^c}). \tag{18}$$

The score function is defined based on cross-view consistency. We calculate scores between every possible combination and sum them up if there are more than 2 views. For joints, Symmetric Epipolar Distance (SED) [43] is used as the score function. For limbs, the function is defined as the variation of predicted unit orientation vectors under all views. Thus the score functions for joints and limbs are

$$f^{jp}(y_1,\ldots,y_{n^c}) = \frac{\sum_{\substack{1\le i,j\le n^c \\ i\neq j}} w_i w_j SED(y_i, y_j)}{\sum_{\substack{1\le i,j\le n^c \\ i\neq j}} w_i w_j} \tag{19}$$

$$f^{lo}(y_1,\ldots,y_{n^c}) = var(y_1,\ldots,y_n) \tag{20}$$

In addition, we define *relative scores* as the score relative to pre-fixing samples, *i.e.* $s = S_{\delta_1\delta_2\cdots\delta_{n^c}}/S_{00\cdots0}$

The smallest score is preferable but not always the best, so some extra rules are needed. A combination is entitled a successful fix if: 1. (premise) its score is the smallest among all combinations, 2. (effectiveness) its relative score must be smaller than a certain threshold S_0, and 3. (necessity) the pre-fixing score must be larger than another threshold S_M. If any condition is failed, no fixing will be taken.

In conclusion, the general co-fixing process is to generate fixed PAF-subsets of LOFs, choose the successful ones to apply, then generate fixed joint confidence maps and select the successful combination as the final output. The pseudo-code for this algorithm is available in Supp. II. Note that due to the 2D convolutions on each feature map of size $H \times W$ with a kernel of size $(2H-1)\times(2W-1)$, co-fixing is of $O(n^c n^j H^2\, W^2)$ complexity.

IV. Discussion

In this section we provide detailed analysis on the design of the method.

A. Regressing Limb Orientations: Self-Weighted Vs. No Weight

In (4), the special weight, $\|F^{lo}_{ijk}(x)\|$, is introduced for better convergence. Actually, if we directly take the average of all vectors without weights, then the direction will be $\hat{V}'_{ijk} = \sum_{x\in D} F^{lo}_{ijk}(x)$. The derivatives of $\hat{V}_{ijk}$ and $\hat{V}'_{ijk}$ are different:

$$\frac{\partial \hat{V}_{ijk}}{\partial F^{lo}_{ijk}(x)} = \|F^{lo}_{ijk}(x)\| \left(I + \frac{F^{lo}_{ijk}(x) F^{lo}_{ijk}(x)^T}{\|F^{lo}_{ijk}(x)\|^2}\right) \tag{21}$$

$$\frac{\partial \hat{V}'_{ijk}}{\partial F^{lo}_{ijk}(x)} = I \tag{22}$$

One major difference between them is when $\|F^{lo}_{ijk}(x)\| \to 0$, $\partial \hat{V}_{ijk}/\partial F^{lo}_{ijk}(x) \to 0$ but $\partial \hat{V}'_{ijk}/\partial F^{lo}_{ijk}(x) = I$. Since $\|F^{lo}_{ijk}(x)\| \approx 0$ usually means x is on the background, when the loss back-propagates to LOFs, our self-weighted average will hardly update the background points, while no-weight average will modify all vectors equally, leading to divergence.

B. Reprojection Error in 3D Space

The general process of lifting linear triangulation to 3D space is detailed here. Projection takes a simple form in projective space if the camera projection matrix P is known: $\bar{x} = P\bar{X}$, where $\bar{x} \in \mathbb{P}^2$ and $\bar{X} \in \mathbb{P}^3$ are the projected 2D point and global 3D point, both homogeneous. To measure the re-projection error, $\bar{x}$ must be converted to Euclidean space via a nonlinear process. This is where the approximation happens. The common method linearizes by multiplying the depth term d_X. It transforms the re-projection error to the following form (proof available in Supp. I.B.):

$$e = \left(X - d(X)K^{-1}\hat{\bar{x}}\right)^T \text{diag}\left\{f_x^2, f_y^2, 0\right\}\left(X - d(X)K^{-1}\hat{\bar{x}}\right), \tag{23}$$

where f_x and f_y are intrinsic parameters and $\hat{\bar{x}} = [\hat{x}^T, 1]^T$ is the homogeneous coordinate of the estimated 2D joint. Usually, there holds $f_x \approx f_y$, so we can directly eliminate $\text{diag}\{f_x^2, f_y^2, 0\}$ and the final objective function is measured in 3D Euclidean distance approximately proportional to (7).

C. Algebraic Advantage of Compound Triangulation

Besides fusing limb features, it is important to notice that Compound Triangulation is able to bind joints together in optimization. In (8), the shared α binds the two unknowns X_i, X_j together. The two points are therefore no longer optimized independently. Consequently, all key points are connected this way throughout the tree structure. The triangulation becomes holistic over all joints, allowing the aggregation of global features.

D. Co-Fixing in Physical Perspective

The relationship between joint confidence maps and LOFs is like that of electric charges and fields. Actually, the convolutional kernel C is the same as the electrical field of a single negative charge located in (h, w). Regard H_{ik}^{jp} as a set of grid-arranged electric charges and F_{ijk}^{pa} as electric fields. By convolving C over H_{ik}^{jp}, we can get the combined electric field, which resembles the PAF-subset. Meanwhile, by convolving C over F_{ijk}^{pa}, we can find the most likely distribution of electrical charges to generate such field, thus inferring joint estimations.

V. Experiment

A. Datasets and Metrics

1) Human3.6M: The Human3.6M Dataset [44] is currently the largest available single-person 3D HPE benchmark. More than 3.6M images are captured by 4 cameras at a framerate of 50 Hz. The motions are completed by 11 actors and corresponding image sets are marked as S1∼S11. In 3D HPE tasks, by tradition, S1, S5, S6, S7, and S8 are used as training sets, and samples of every 64 frames in S9 and S11 are used as test sets. The annotations are in 33-joint forms, and a 17-joint subset is used in our experiments as a common benchmark.

2) Total Capture: The Total Capture Dataset [45] is another large-scale single-person motion capture dataset. 8 HD cameras are used to capture around 1.9M images at a framerate of 60 Hz. The image data are organized by intersections of subjects and actions, where "Walking-2" (W2), "Freestyle-3" (FS3), and "Acting-3" (A3) of all subjects are used as test sets, and the rest actions of S1, S2, and S3 are used as the training set. So there are both seen and unseen subjects in testing. Similarly to Human3.6M, we sample every 64 frames while testing. Pose annotations in Total Capture are in 21-joint form, and a 16-joint subset is used.

3) Metrics: 3D joint estimations are evaluated by the common Mean Per Joint Position Error (MPJPE) in millimeters, which is the average Euclidean distance between estimated joints and ground truth. Two versions are usually used. Absolute MPJPE (MPJPE-ab) takes the average directly in the world coordinate system, while relative MPJPE (MPJPE-re) is calculated after aligning the pelvis. Note that when evaluating on Human3.6M with MPJPE-abs, we follow the previous work [4] to remove actions with shifted labels. Moreover, we utilize Limb Angular Error (LAE), the mean angles between GT and predicted limb orientations, to analyze the angular error of poses and LOFs.

B. Implementation Details

1) Hyperparameters:

- While regressing parameters using (3) and (5), the inverse temperature β is an important hyperparameter. Larger β drives soft-argmax closer to argmax function. This gives better results in the early training stage and helps with faster convergence, but increases the difficulty to cope with quantization error. We follow the previous work [4] to set $\beta = 100$ in experiments as a balance.
- In the overall loss function (13), we set the hyperparameter $\mu = 10^3$ to balance the order of magnitudes of 3D losses and 2D losses. It keeps $\mu\mathcal{L}^v$ within $(0.01\mathcal{L}^{jp}, 0.1\mathcal{L}^{jp})$ most of the time. Since the supervision of pointwise vector directions is just auxiliary, this setting keeps it functional but far from dominant.
- Hyperparameters in co-fixing are mostly set empirically. The fading factor γ is set to 0.5 for Human3.6M and 2 for Total Capture, We set $S_0 = 0.25$ for both SEDs and limb orientation variance, and the necessity threshold S_M is set to 400 for the former and 0.1 for the latter.

2) Training Settings: In experiments, ResNet152 [46] is used as our backbone, following 2 branches of deconvolutional layers in SimpleBaseline [34], one for extracting joint heatmaps and the other for LOFs. In the test on Human3.6M, we utilize the pre-trained backbone weights by Iskakov et al. [4], which is trained on COCO dataset [47] and fine-tuned jointly on Human3.6M and MPII [48] datasets. The images are cropped by ground truth bounding boxes and resized to 384×384px, with a heatmap size of 96×96. In the test on Total Capture, no extra data are used. We initialize the backbone by the weights pre-trained on ImageNet [49] and use only Total Capture data to train. The images are also cropped by ground truth bounding boxes but resized to 320×320px, with a heatmap size of 80×80.

Both training procedures follow the two-stage pattern in Section III-D, with 10 epochs in each stage. We use Adam optimizer [50]. The learning rates are 10^{-3} in pre-training and 10^{-4}

TABLE I
COMPARISON TO PREVIOUS STATE-OF-THE-ART METHODS IN MPJPE (mm) ON HUMAN3.6M DATASET

Relative MPJPE on 4 view setting	Dire.	Disc.	Eat	Greet.	Phone.	Photo.	Pose.	Purch.	Sit.	SitD.	Smoke.	Wait.	WalkD.	Walk.	WalkT.	Avg.
Pavlakos *et al.* [51]	41.2	49.2	42.8	43.4	55.6	46.9	40.3	63.7	97.6	119.0	52.1	42.7	51.9	41.8	39.4	56.9
Tome *et al.* [52]	43.3	49.6	42.0	48.8	51.1	64.3	40.3	43.3	66.0	95.2	50.2	52.2	51.1	43.9	45.3	52.8
Kadkhodamohammadi and Padoy [1]	39.4	46.9	41.0	42.7	53.6	54.8	41.4	50.0	59.9	78.8	49.8	46.2	51.1	40.5	41.0	49.1
Qiu *et al.* [21]	24.0	26.7	23.2	24.3	24.8	22.8	24.1	28.6	32.1	26.9	31.0	25.6	25.0	28.1	24.4	26.2
AlgTri. by Iskakov *et al.* [4]	20.4	22.6	20.5	19.7	22.1	20.6	19.5	23.0	25.8	33.0	23.0	21.6	20.7	23.7	21.3	22.6
VolTri. by Iskakov *et al.* [4]	19.9	20.0	18.9	**18.5**	20.5	19.4	**18.4**	22.1	**22.5**	28.7	21.2	20.8	19.7	22.1	20.2	20.8
Zhe *et al.* [10]	**17.8**	**19.5**	**17.6**	20.7	**19.3**	**16.8**	18.9	**20.2**	25.7	**20.1**	**19.2**	20.5	**17.2**	20.5	**17.3**	**19.5**
Ours	17.9	20.6	19.9	19.0	20.0	18.5	21.4	23.0	27.8	21.0	20.4	**20.4**	22.0	**19.2**	19.7	20.7
Relative MPJPE on stereo settings	**Dire.**	**Disc.**	**Eat**	**Greet.**	**Phone.**	**Photo.**	**Pose.**	**Purch.**	**Sit.**	**SitD.**	**Smoke.**	**Wait.**	**WalkD.**	**Walk.**	**WalkT.**	**Avg.**
AlgTri. by Iskakov *et al.* [4]	56.9	51.8	39.9	56.1	48.0	50.7	48.5	48.4	51.1	54.5	47.8	51.7	48.0	37.0	40.9	49.0
VolTri. by Iskakov *et al.* [4]	47.7	45.2	42.6	47.6	46.0	52.7	37.0	44.1	49.2	54.3	44.8	44.6	41.4	32.0	34.0	44.7
Zhe *et al.* [10]	49.5	43.8	36.3	49.1	40.3	93.1	38.8	40.9	47.6	39.4	53.2	44.8	**30.4**	40.8	32.5	44.6
Ours	**32.1**	**34.1**	**30.8**	**34.0**	**34.8**	**36.4**	**28.9**	**33.0**	**39.4**	**48.1**	**35.3**	**34.1**	35.3	**28.1**	**29.7**	**34.5**
Absolute MPJPE on stereo settings	**Dire.**	**Disc.**	**Eat**	**Greet.**	**Phone.**	**Photo.**	**Pose.**	**Purch.**	**Sit.**	**SitD.**	**Smoke.**	**Wait.**	**WalkD.**	**Walk.**	**WalkT.**	**Avg.**
AlgTri. by Iskakov *et al.* [4]	56.3	49.2	38.5	52.9	45.4	48.0	47.8	45.6	46.8	44.1	45.2	48.6	38.5	45.3	40.9	46.3
VolTri. by Iskakov *et al.* [4]	46.0	43.1	40.2	44.9	41.8	49.9	34.8	39.4	43.8	45.9	40.7	41.0	30.1	38.6	31.8	41.1
Zhe *et al.* [10]	47.1	41.2	34.7	70.7	38.3	68.9	36.4	38.6	70.1	**37.4**	50.0	64.3	29.1	38.5	30.9	45.6
Ours	**31.0**	**32.6**	**28.3**	**30.4**	**32.1**	**33.2**	**27.6**	**29.7**	**35.4**	40.6	**32.6**	**31.0**	**26.5**	**32.1**	**27.5**	**31.6**

Tests are done on 4 view settings and allcombinations of 2 view settings. In each column, the best results are marked in bold, and underlined numbers indicate the second best.

in end-to-end training. On the Linux server with a 16-core Intel E5-2620 CPU, 32 G RAM, and two NVIDIA TITAN X GPUs, we set batch size to 6 for 4-view Human3.6M training, 8 for Total Capture Training, and 16 for Total Capture stereo training. Correspondingly, the training time per batch is 0.93 s, 0.93 s and 0.99 s.

C. Quantitative Analysis

In this section, our method is compared to previous state-of-the-art methods on two public datasets. In addition to the common 4-view test settings, we focus on 2-view settings in order to test stereoscopic performance. By "Baseline," we refer to Algebraic Triangulation (AlgTri.) by Iskakov et al. [4].

1) MPJPE Results on Human3.6M Dataset: The MPJPE results on Human3.6M are presented in Table I. In traditional 4-view benchmark, our method outperforms the baseline in all actions with an 8% average error drop. The average performance exceeds the Volumetric Triangulation method and is comparable to the previous state-of-the-art AdaFuse [10] method. Specifically, our method performs well on some hard sub-action sets like Waiting and SittingDown due to the superiority in handling occlusions. We will discuss this advantage further in Section V-D-2.

In stereoscopic setting tests, the weights trained on 4-view settings are used. The reported errors in Table I are the average of all possible 2-out-of-4 combinations for the purpose of generality. In this criterion, our method brings a decrease of 19.2% in MPJPE-re and 23.1% in MPJPE-ab compared to previous state-of-the-art methods. As the tests are done on camera settings different from training, it also indicates the adaptability from more views to less.

2) MPJPE Results on Total Capture: In Total Capture dataset, our test is also composed of 4-view and 2-view (stereo) parts, but in all camera settings, models are trained and tested separately. Following previous works [5], [21], we use cameras 1, 3, 5, and 7 in 4-view tests, which are referred to as G4. For stereo settings, we use 3 groups, *i.e.*, G1: 5 & 6, G2: 1 & 3, and G3: 1 & 4. As the 8 cameras in Total Capture are located clockwise, the three groups can represent 3 different lengths of baselines between stereo camera pairs. We report the 4-view results in Table II and stereo results in Table III, both in absolute MPJPE (mm). The experiment results show that our method achieves SoTA performance under 4-view settings, exceeding the previous SoTA method by Remelli et al. [5] by 9.8% and the strong baseline by 5.3%, which is remarkable considering the boundary effects. Moreover, our method performs especially well in "freestyle" and "action" subsets, which again implies that our method is well suited for hard actions with less common movements and more severe self-occlusions.

TABLE II
TEST RESULTS ON TOTAL CAPTURE DATASET

Methods	Seen Subjects			Unseen Subjects			Avg.
	W2	A3	FS3	W2	A3	FS3	
Tri-CPM [33]	79	106	112	79	73	149	99
RPSM [21]	28	30	42	45	46	74	41
AutoEnc [53]	13.0	23.0	47.0	**21.8**	40.9	68.5	34.1
Remelli *et al.* [5]	10.6	16.3	30.4	27.0	34.2	65.0	27.5
AlgTri. [4]	7.9	13.5	30.9	23.9	35.6	64.6	26.2
Ours	**7.6**	**13.0**	**28.3**	22.6	**31.6**	**63.7**	**24.8**

The cameras used are 1, 3, 5, 7. The best results in each subtable are marked in bold.

The Detailed MPJPEs on all stereoscopic groups are reported in Table III in comparison with baseline and several SoTA methods. In these stereoscopic settings, our method gets the best performance, exceeding the baseline by 33.6% and previous SoTA methods by at least 3.6%. We can draw some conclusions by analyzing the underlying principles. Cross-view complement in Epipolar Transformer and AdaFuse ensures good adaptation to settings with 4 cameras or more, but cannot fit in stereoscopic

TABLE III
MPJPEs (mm) RESULTS ON 3 STEREOSCOPIC SETTINGS IN TOTAL CAPTURE DATASET

Methods	G1, seen			G1, unseen			G2, seen			G2, unseen			G3, seen			G3, unseen			Avg.
	W2	A3	FS3	W2	A3	FS3	W2	A3	FS3	W2	A3	FS3	W2	A3	FS3	W2	A3	FS3	
Baseline (AlgTri.)	83.3	71.3	101.5	106.1	95.7	174.3	25.6	27.2	61.8	47.2	67.5	117.8	14.0	25.5	**48.3**	35.1	51.1	92.9	64.3
Epipolar Transformer [8]	59.8	65.3	144.1	83.8	81.3	187.1	22.5	28.3	86.9	42.6	58.8	139.5	15.4	25.1	69.9	36.4	55.8	134.2	69.3
AdaFuse [10]	71.5	77.2	122.4	79.8	123.5	177.6	55.6	60.2	79.5	60.6	67.5	126.8	62.7	71.3	94.3	69.4	91.3	130.4	86.7
Faster-VoxelPose[1] [15]	36.6	35.7	90.2	56.6	82.3	147.2	64.8	71.0	101.4	86.3	90.2	154.2	25.6	41.1	94.1	54.5	79.7	159.1	76.0
VolTri. [4]	19.2	**33.5**	65.3	40.1	59.1	119.3	14.0	20.5	54.8	31.9	54.3	112.1	**13.7**	**23.5**	48.9	**32.3**	**45.4**	93.1	44.3
Ours	**17.1**	**33.5**	**64.1**	**39.1**	**55.1**	**103.0**	**13.1**	**19.3**	**52.2**	**32.0**	**48.1**	**100.9**	14.8	26.4	52.7	33.8	45.7	**90.5**	**42.7**

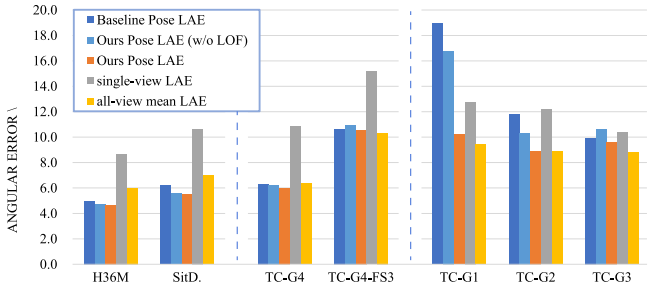

Fig. 6. Angular errors on two datasets and representative sub-actions sets. "TC" and "H36 M" represent Total Capture and Human3.6M datasets in order, and the following subtitles refer to groups of cameras and sub-actions. Barely *LAE* measures limb orientations regressed from LOFs, while *Pose LAE* measures those calculated from estimated 3D poses. By "w/o LOF," we refer to the triangulation result of merely joint locations from our model. By "single-view," limb orientations are measured separately in each view, whereas by "all-view," evaluated orientations are the average of the same limbs over all views.

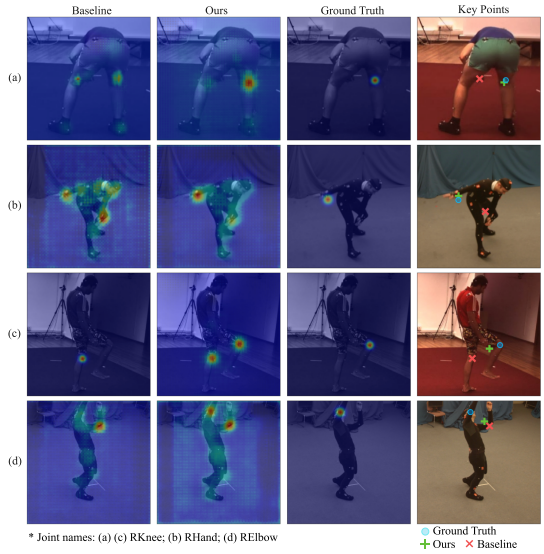

Fig. 7. Joint confidence map comparisons between Baseline method and ours. The leftmost 3 columns represent the confidence maps of Baseline, ours, and ground truth. The studied joint names are listed below. In the rightmost column, the studied key points from different models are marked according to the bottom legends.

settings because the sparsity of information makes the complement unpractical. The volume-based VolTri. method is adaptive to stereoscopic settings because of its capability to encode implicit pose priors via 3D convolutions, and Faster-Voxelpose[1] performs worse probably due to the disuse of them. However, given the accuracy, our methodology to extract more information from images is potentially more effective.

3) Angular Analysis: To study the reason for performance enhancing in quantitative aspect, we focus our attention mainly on the limb orientations, using the angular metric LAE introduced in Section V-A-3. Fig. 6 shows the angular metrics on both datasets and some particular subsets.

Considering the Pose LAEs from all action sets, the angular estimations are clearly improved on average. The improvement could be considered in two stages: 1. From baseline to our model without LOF branch, where only the trained weights are shifted; 2. Applying LOF branch to correct our estimations. Considering the first 3 bars in each group in Fig. 6, it seems the first stage may not necessarily improve the angular metrics, but the second stage is proved to bring about positive effects.

Actually, compound triangulation is a process to fix pose estimations using limb orientations, so as the latter become more accurate, the improvement magnifies. This is validated by data in Fig. 6. It also explains why our method fits stereoscopic settings well: LOFs are not so sensitive to camera numbers as triangulation is, so in stereoscopic cases, limb orientations from LOFs are more accurate than triangulation results, which leads to effective pose corrections.

In addition, Although limb estimations can sometimes be less accurate than poses, the fusion process mostly brings positive correction. This is mainly because the learnable weights w^{jp} and w^{lo} are capable of eliminating noisy LOFs. Moreover, in hard cases, limb orientations tend to be more stable than poses and thus contribute to the robustness.

D. Qualitative Analysis

In this section, we study the qualitative aspects for performance boosts in stereoscopic settings.

1) Double Counting Correction: The double counting problem, *i.e.* the ambiguity between symmetric joints in 2D detection, harms 3D pose estimation, especially in stereo. The corrections happen in two stages: the supervision of LOF in end-to-end training and post-processing by co-fixing module.

First, the supervision of LOFs shows an implicit and generally positive effect on 2D estimations. We compare samples from our

[1]Faster-Voxelpose is designed for multi-person 3D HPE, so it includes a Human Detection Net and a Joint Localization Net. Since Total Capture is a single-person dataset, we use only Joint Localization Net in our test and use AlgTri. to provide rough volume positions.

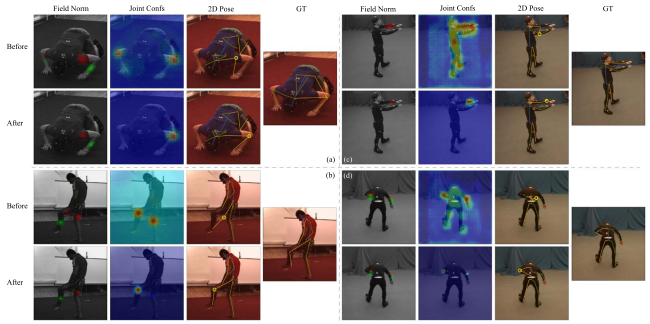

Fig. 8. Case study of co-fixing algorithm. Four cases are illustrated. Aside from the rightmost GT 2D pose, the rest 6 figures are labeled according to columns of field norm (pointwise norms of LOFs), joint confidence, and 2D pose (Remarkably fixed points are marked in yellow) and rows of before and after co-fixing. Field norm illustrations may contain multiple adjacent limbs, and they are marked with different colors, e.g., red and green.

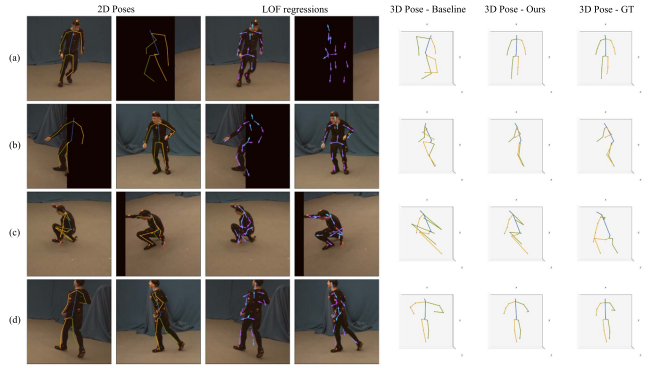

Fig. 9. Case study of compound triangulation. For each case, the figures are 2D poses of the two available views, limb predictions from LOFs from both views and 3D poses of baseline (linear triangulation), ours(compound triangulation), and ground truth.

method and the baseline and report results in Fig. 7. Improvements include: shifting displaced or ambiguous predictions to the correct position like Fig. 7(a), (b), and changing from wrong to ambiguous state like Fig. 7(c), (d). As LOFs focus on encoding features from limbs, they may force the model to gradually learn to correlate features of joints and limbs and consequently form a big picture of the whole limb.

Next, the effects of co-fixing module with specific cases are reported in Fig. 8, where we conclude that both limb fields and joint confidence maps tend to shift for the better. Focusing on joint predictions, it is obvious that the incorrect responses are repressed. Precise limb predictions clearly contribute to it, as shown in Fig. 8(a), (c). However, we also find in Fig. 8(b), (d) that initially ambiguous LOFs can also lead to correct fixes. The reason can be found in co-fixing procedure. As the reverse correction incorporates both adjacent joints instead of one, the ambiguity of limbs is more likely to be solved or alleviated. Thus the correction is still valid.

2) Effects of LOFs and Compound Triangulation.: In stereoscopic settings, the incorporation of limb predictions in triangulation is also important. We illustrate some frames from experimental results of G1 in Fig. 9.

TABLE IV
ABLATION STUDY

Methods	Components and Modules			MPJPE (mm)	
	LOF	CompTri.	Co-Fixing	G4	G2
Baseline				26.8	46.3
LOF	✓			25.8	49.7
PAF + CompTri.		✓		27.0	47.6
LOF + CompTri.	✓	✓		25.3	33.2
LOF + CompTri. + Co-fixing	✓	✓	✓	**24.9**	**32.9**

Methods are trained on G4 (4 views) and tested on both G4 and G2 (stereo). CompTri. efers to compound triangulation.

In Fig. 9(a), (b), one of the two views is thoroughly or partly occluded. Traditional triangulation requires ≥ 2 views to accurately locate a joint so the baseline fails to produce plausible poses. Our method, however, does not require the same. Compound triangulation is able to construct accurate 3D poses even if only one view is available. It is similar to monocular reconstruction methods [18], [25], but our method is more flexible in the capability to tackle multi-view settings: The visible parts of other views are engaged in triangulation, while the occluded parts are simply filtered out by learnable weights.

In Fig. 9(c), (d), the persons are fully in view, but self-occlusions exist. In this case, our method could provide more clues by LOFs, e.g. the orientations of visible body parts like right forearms in (c) and (d), which are crucial in driving relative limbs to the right orientations. To sum up, compound triangulation excels in incorporating the most possible information, and producing the potentially most accurate 3D pose.

E. Ablation Study

In this section, we study the effects of LOFs, compound triangulation, and co-fixing module. The test results are reported in Table IV. As PAFs possess purely 2D information, the distance analogous to e^{lo} in (8) is defined as the distance between re-projection points and the limb on the image plane (details available in Supp. III). Note that there is no factor to bound adjacent joints together like α in (8), so the joints are optimized independently.

The results in Table IV suggest the necessity of each component. For LOF, the potential lies in the extra dimension compared to PAF. This dimension is capable of connoting information perpendicular to the image plane, and correlating adjacent joints in triangulation stage, forcing the model to optimize human pose as a whole. The performance drops while changing from LOF to PAF in both camera settings validate the above analysis. Additionally, considering the results of Baseline, LOF, and "LOF + CompTri.," we conclude that LOF does not significantly benefit pre-training, but is crucial as an integral component of compound triangulation. The effect increases as the number of views decreases. By comparing the last two lines, it is also evident that co-fixing module promotes general performance. The marginal error drop is mostly due to the infrequency of double counting.

VI. Conclusion

In this paper, we propose a novel framework to incorporate Limb Orientation Field, *i.e.* LOF to promote stereoscopic 3D human pose estimation. Major contributions include an explicit module known as compound triangulation to fuse multi-view limb estimations with 2D poses, and a post-processing module named co-fixing to eliminate ambiguity between joints. The experiment results validate the effect in stereoscopic settings and the adaptability to general multi-view scenes. The effect of each module is also validated in the ablation study. Future work is planned to improve co-fixing module. Powered by convolutions, it is potentially possible to be integrated into end-to-end training, which can eliminate the need for predefined filtering parameters and increase its generality. The fixing process could also incorporate cross-view information for better accuracy.

References

[1] A. Kadkhodamohammadi and N. Padoy, "A generalizable approach for multi-view 3D human pose regression," *Mach. Vis. Appl.*, vol. 32, no. 1, 2021, Art. no. 6.

[2] M. Ghafoor and A. Mahmood, "Quantification of occlusion handling capability of 3D human pose estimation framework," *IEEE Trans. Multimedia*, vol. 25, pp. 3311–3318, 2023.

[3] N. Nakano et al., "Evaluation of 3D markerless motion capture accuracy using OpenPose with multiple video cameras," *Front. sports Act. Living*, vol. 2, 2020, Art. no. 50.

[4] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7718–7727.

[5] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang, "Lightweight multi-view 3D pose estimation through camera-disentangled representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6040–6049.

[6] R. Xie, C. Wang, and Y. Wang, "MetaFuse: A pre-trained fusion model for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13686–13695.

[7] Z. Zhang, C. Wang, W. Qin, and W. Zeng, "Fusing wearable IMUs with multi-view images for human pose estimation: A geometric approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2200–2209.

[8] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu, "Epipolar transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7779–7788.

[9] H. Tu, C. Wang, and W. Zeng, "VoxelPose: Towards multi-camera 3D human pose estimation in wild environment," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2020, pp. 197–212.

[10] Z. Zhang, C. Wang, W. Qiu, W. Qin, and W. Zeng, "AdaFuse: Adaptive multiview fusion for accurate human pose estimation in the wild," *Int. J. Comput. Vis.*, vol. 129, pp. 703–718, 2021.

[11] Z. Chen, X. Zhao, and X. Wan, "Structural triangulation: A closed-form solution to constrained 3D human pose estiation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 695–711.

[12] H. Ma et al., "PPT: Token-pruned pose transformer for monocular and multi-view human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 424–442.

[13] Y. Bao, X. Zhao, and D. Qian, "FusePose: IMU-vision sensor fusion in kinematic space for parametric human pose estimation," *IEEE Trans. Multimedia*, vol. 25, pp. 7736–7746, 2023.

[14] K. Bartol, D. Bojanić, T. Petković, and T. Pribanić, "Generalizable human pose triangulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11028–11037.

[15] H. Ye, W. Zhu, C. Wang, R. Wu, and Y. Wang, "Faster voxelpose: Real-time 3D human pose estimation by orthographic projection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 142–159.

[16] R. I. Hartley and P. Sturm, "Triangulation," *Comput. Vis. Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[17] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10965–10974.

[18] D. Liu et al., "Improving 3D human pose estimation via 3D part affinity fields," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1004–1013.

[19] H. Wu and B. Xiao, "3D human pose estimation via explicit compositional depth maps," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12378–12385.

[20] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Comput. Surv.*, vol. 14, no. 4, pp. 553–572, 1982.

[21] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3D human pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4342–4351.

[22] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2014, pp. 33–47.

[23] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7291–7299.

[25] C. Luo, X. Chu, and A. Yuille, "OriNet: A fully convolutional network for 3D human pose estimation," 2018, *arXiv:1811.04989*.

[26] R. I. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1992, pp. 761–764.

[27] X. Zhao, Y. Fu, H. Ning, Y. Liu, and T. S. Huang, "Human pose regression through multiview visual fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 7, pp. 957–966, Jul. 2010.

[28] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel, "Optimization and filtering for human motion capture: A multi-layer framework," *Int. J. Comput. Vis.*, vol. 87, pp. 75–92, 2010.

[29] M. Hofmann and D. M. Gavrila, "Multi-view 3d human pose estimation in complex environment," *Int. J. Comput. Vis.*, vol. 96, pp. 103–124, 2012.

[30] Y. Liu, C. Stoll, J. Gall, H.-P, Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1249–1256.

[31] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[32] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.

[33] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4724–4732.

[34] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2018, pp. 466–481.

[35] G. Hidalgo et al., "Single-network whole-body pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6982–6991.

[36] D. Osokin, "Real-time 2D multi-person pose estimation on CPU: Lightweight openpose," 2018, *arXiv:1811.12004*.

[37] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, Jan. 1973.

[38] M. Burenius, J. Sullivan, and S. Carlsson, "3D pictorial structures for multiple view articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3618–3625.

[39] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1385–1392.

[40] L. Ke, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 713–728.

[41] A. Kamel, B. Sheng, P. Li, J. Kim, and D. D. Feng, "Hybrid refinement-correction heatmaps for human pose estimation," *IEEE Trans. Multimedia*, vol. 23, pp. 1330–1342, 2021.

[42] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2016.

[43] M. E. Fathy, A. S. Hussein, and M. F. Tolba, "Fundamental matrix estimation: A study of error criteria," *Pattern Recognit. Lett.*, vol. 32, no. 2, pp. 383–391, 2011.

[44] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6 m: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[45] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3D human pose estimation fusing video and inertial sensors," in *Proc. 28th Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[47] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[48] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3686–3693.
[49] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[51] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Harvesting multiple views for marker-less 3D human pose annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6988–6997.
[52] D. Tome, M. Toso, L. Agapito, and C. Russell, "Rethinking pose in 3D: Multi-stage refinement and recovery for markerless motion capture," in *Proc. IEEE Int. Conf. 3D Vis.*, 2018, pp. 474–483.
[53] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse, "Deep autoencoder for combined human pose estimation and body model upscaling," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2018, pp. 784–800.

Zhuo Chen received the B.E. degree in 2021 from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the M.S. degree. His research interests include human pose estimation, multiview geometry, and machine learning.

Xiaoyue Wan received the B.S. degree in automation and the M.S. degree in control engineering from the Southeast University, Nanjing, China, in 2015 and 2018, respectively. She is currently working toward the Ph.D. degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. Her research interests include computer vision and human pose estimation.

Yiming Bao received the B.S. degree in biomedical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently working toward the Ph.D. degree with the Department of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include human motion, visual-inertial fusion, and deep learning.

Xu Zhao (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent system from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2011. He is currently a Full Professor with the Department of Automation, School of Electronic Information and Electrical Engineering, SJTU. He was a Visiting Scholar with the Beckman Institute, University of Illinois Urbana-Champaign, Urbana, IL, USA, from 2007 to 2008, and a Postdoc Research Fellow with North eastern University, Boston, MA, USA, from 2012 to 2013. His research interests include visual analysis of human motion, machine learning, and image/video processing.