Context Modeling in 3D Human Pose Estimation: A Unified Perspective

Xiaoxuan Ma[1,3*], Jiajun Su[2*], Chunyu Wang[4], Hai Ci[1,5], Yizhou Wang [1]
[1]Dept. of Computer Science, Center on Frontiers of Computing Studies, Peking University
[2]Center for Data Science, Adv. Inst. of Info. Tech., Peking University
[3]Advanced Innovation Center For Future Visual Entertainment (AICFVE), Beijing Film Academy
[4]Microsoft Research Asia [5]Deepwise AI Lab
{maxiaoxuan, sujiajun, cihai, yizhou.wang}@pku.edu.cn, chnuwa@microsoft.com

Abstract

Estimating 3D human pose from a single image suffers from severe ambiguity since multiple 3D joint configurations may have the same 2D projection. The state-of-the-art methods often rely on context modeling methods such as pictorial structure model (PSM) or graph neural network (GNN) to reduce ambiguity. However, there is no study that rigorously compares them side by side. So we first present a general formula for context modeling in which both PSM and GNN are its special cases. By comparing the two methods, we found that the end-to-end training scheme in GNN and the limb length constraints in PSM are two complementary factors to improve results. To combine their advantages, we propose ***ContextPose*** *based on attention mechanism that allows enforcing soft limb length constraints in a deep network. The approach effectively reduces the chance of getting absurd 3D pose estimates with incorrect limb lengths and achieves state-of-the-art results on two benchmark datasets. More importantly, the introduction of limb length constraints into deep networks enables the approach to achieve much better generalization performance.*

1. Introduction

Monocular 3D human pose estimation has attracted much attention [5, 21, 23, 31, 35, 42] because it can benefit many applications such as virtual reality and intelligent video analysis. The task is more difficult than 2D pose estimation [7, 30, 32] because it needs to estimate relative depth between body joints which suffers from severe ambiguity. Psychology experiments [4] show that *context* plays an important role in resolving ambiguity in human visual system. Following this idea, body joints can serve as mutual context to each other in human pose estimation— localizing one facilitates the localization of the other. For example, elbow is more likely to be found at a distance from shoulder depending on the length of upper arm. Some work [12] also explores surrounding environment as context for joints to further narrow down the space.

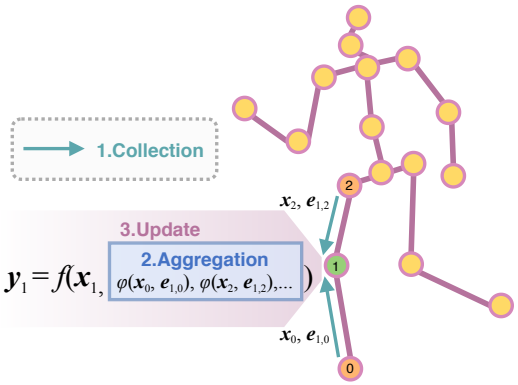

Figure 1: A general formula of context modeling in the 3D human pose estimation task. To update features of a particular joint, the approach first collects features from its contextual joints (defined by the input graph structure), aggregates the collected features, and uses the features to update the joint of interest.

The success of CNN in 2D pose estimation [7, 24, 32] has promoted a shift from model-based 3D pose estimators [9, 19, 29, 37] to discriminative ones [10, 16, 21, 31]. In particular, Martinez *et al.* [21] propose to estimate 3D pose from estimated 2D pose by a Fully Connected Network (FCN). It achieves notably smaller error than previous methods due to its strong capability of fitting large amounts of data and improved 2D pose estimation accuracy. But it does not explicitly explore context which may result in poor results in challenging cases [10].

* denotes equal contribution.

GNN [11] computes features for each node by aggregating those of its neighbors. The interaction among nodes makes it suitable for modeling context. For example, Ci *et al*. [10] treat each joint as a node and perform feature passing among the nodes to estimate their 3D locations. The method is more robust to inaccurate 2D poses which validates the values of context. But they cannot explicitly model spatial relation between joints such as limb length constraints which is a big limitation— limb length is useful to reduce ambiguity when some joints are occluded.

PSM [1, 3, 18, 26, 28] had been commonly used for both 2D and 3D pose estimation before deep networks dominate the field. The key idea is to determine optimal joint locations by simultaneously considering their appearance and spatial relation. For example, Qiu *et al*. [28] divide the 3D motion space by regular voxels and assign each joint to the optimal voxel by minimizing an energy function defined on all joints. The approach may get accurate 3D estimates for occluded joints based on their neighbors. Some works [8, 26, 28] also combine PSM with deep learning by first applying CNN to estimate features and then using PSM to do inference on the features. However, the improvement is limited because it cannot be trained end-to-end.

To our best knowledge, there is no work discussing the pros and cons of PSM [1, 3, 18, 26] and GNN [10, 40] since they were developed in different fields. But this is actually very important. To that end, starting from their standard formulation, we develop a general formula for the two methods which allows us to clearly understand their relations and differences. In the meanwhile, we can compare their advantages and disadvantages side by side. The basic idea is sketched in Figure 1. It has three steps: for each joint of interest, it first *collects* features from its contextual joints which are determined by the input human graph. Then it *aggregates* the collected features as context which in turn is used to *update* the features of the joint.

In particular, we find in our empirical study that the GNN-based methods [10, 21, 40] powered by end-to-end learning get more accurate estimates than PSM in general cases. We believe this is mainly because deep neural networks have strong capability to fit a large amount of data. On the other hand, PSM-based methods [1, 3, 18, 26, 28] are more robust to occlusion and get better out-of-distribution generalization performance. It is worth noting that PSM is mainly used in the multiview setting. Our experiment in the monocular setting shows that PSM alone gets very bad results because of its limited capability to reduce ambiguity (3D pose estimates may still be inaccurate although their limb lengths are correct). The observation motivates us to combine PSM and GNN in order to benefit from their advantages. Note that the task is non-trivial because PSM requires solving the discrete optimization function.

Method	Formula	Voxel Based	End-to-End	Cyclic Graph	Limb Length Prior
PSM [1, 28]	1	✓	✗	✗	✓
GNN [10, 40]	2	✗	✓	✓	✗
ContextPose (Ours)	4	✓	✓	✓	✓

Table 1: Comparison of different context modeling methods. Please refer to Section 3.5 for more details.

To that end, we present an approach termed as *ContextPose* on top of the general formula which is inspired by the attention mechanism [34]. It is built on the voxel representation [14, 33] and allows enforcing soft limb length constraints by *paying more attention to information passed between locations that satisfy limb length constraints*. More importantly, the approach avoids solving the discrete optimization problem and can be trained end-to-end. Table 1 briefly summarizes different methods.

1.1. Overview

Figure 2 shows how ContextPose is leveraged by the state-of-the-art method [14] for 3D pose estimation. Given an input image, it first estimates 2D features by a 2D network (CNN). Then it inversely projects them to the 3D voxels using camera parameters and uses a 3D network to estimate 3D heatmaps representing the likelihood of each voxel having each body joint. ContextPose can be inserted into the 3D network to fuse features from different joints at different locations. Specifically, *it updates the features of a joint at a voxel by a linear combination of the features of its contextual joints at all voxels*. The weights in linear combination are determined by their spatial relation (pairwise attention) and appearance (global attention) of the contextual joints. The bottom section of Figure 2 shows more details of how we compute global attention and pairwise attention with the knee joint as an example.

In summary, we make three contributions:

1) We develop a general formula for context modeling methods in 3D human pose estimation which allows us to clearly understand their pros and cons. We also empirically compare them in a rigorous way.

2) We propose *ContextPose* on top of the general formula which combines the advantages of PSM and GNN. In particular, it allows leveraging limb length constraints and can be leveraged by 3D pose estimation networks for end-to-end training.

3) We demonstrate the state-of-the-art performance on two benchmark datasets. More importantly, ContextPose shows better generalization results on out-of-distribution data. The code and models will be released in order to inspire more research in this direction.

2. Context Modeling: A Unified Perspective

We first introduce some notations and then reformulate PSM and GNN, respectively. Based on the reformulation, we develop a general formula for context modeling and show that both PSM and GNN are its special cases.

2.1. Notations

As shown in Figure 1, we represent human body by a graph $\mathcal{G} = (\mathcal{J}, \mathcal{E})$ where $\mathcal{J} = \{J_0, J_1, \cdots, J_{N-1}\}$ represents N body joints. The set $\mathcal{E}$ represents edges that connect pairs of joints. We define the joints that are connected by edges to be **contextual joints** of each other. The goal of monocular 3D pose estimation is to estimate the 3D locations of the joints from a single image.

2.2. Reformulate PSM

PSM is commonly used in multiview 3D pose estimation [26,28]. It first divides the 3D space by regular voxels Ω with each having a discrete location $\boldsymbol{q} \in \mathcal{R}^3$. The goal of PSM is to assign each joint to one of the voxels by minimizing an energy function defined on all joints. When the human graph is acyclic, PSM can be optimized by dynamic programming in which messages are sequentially passed from child nodes. In particular, the likelihood of a sub-tree with root joint J_u at voxel $\boldsymbol{q}$ is computed as

$$y_{u,\boldsymbol{q}} = x_{u,\boldsymbol{q}} \cdot \prod_{J_v \in \text{child}(J_u)} (\max_{\boldsymbol{k} \in \Omega} \{\psi(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v}) \cdot y_{v,\boldsymbol{k}}\}), \quad (1)$$

where $\text{child}(J_u)$ denotes the children of J_u and $x_{u,\boldsymbol{q}}$ is the confidence of J_u at $\boldsymbol{q}$ determined by appearance.

The formula can be interpreted by three steps: (1) for each non-leaf node J_u, it first collects features from each of its children J_v by $\psi(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v}) \cdot y_{v,\boldsymbol{k}}$ where $y_{v,\boldsymbol{k}}$ represents J_v's likelihood of being at $\boldsymbol{k} \in \mathcal{R}^3$ which in turn is determined by its own children. The pairwise term $\psi(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v})$ encodes the limb length constraint measuring whether the distance between $\boldsymbol{q}$ and $\boldsymbol{k}$ satisfies the limb length prior in $\boldsymbol{e}_{u,v}$. The maximum score over all voxel locations Ω represents the message passed from joint J_v to J_u. This step collects such information from all of its children; (2) then the context features collected from its children are aggregated by $\prod$; (3) finally, it updates $y_{u,\boldsymbol{q}}$ by multiplying the aggregated context with the confidence $x_{u,\boldsymbol{q}}$.

2.3. Reformulate GNN

Ci *et al*. [10] present a formula which unifies FCN [21], GNN [40], and LCN [10]. We further reformulate it such that it has a similar form as PSM

$$\boldsymbol{y}_u = f(\boldsymbol{x}_u, \sum_{J_v \in \mathcal{J}} (\boldsymbol{e}_{u,v} \cdot \boldsymbol{W}_{u,v} \boldsymbol{x}_v)), \quad (2)$$

where $\boldsymbol{x}_u \in \mathcal{R}^{M_{input}}$ represents the features of J_u obtained from the previous layer or input, and $\boldsymbol{y}_u \in \mathcal{R}^{M_{output}}$ denotes the updated features of J_u. It is important to note that these methods do not discretize the 3D space but directly estimate continuous locations. So we do not compute features for each discrete location $\boldsymbol{q}$ as in Eq. (1). The binary scalar $\boldsymbol{e}_{u,v}$ encodes the pairwise relation between joint J_u and J_v, and is set to be one if J_v is a contextual joint of J_u. $\boldsymbol{W}_{u,v} \in \mathcal{R}^{M_{output} \times M_{input}}$ is a learnable weight matrix. We can also interpret the formula by three steps in a similar way as PSM. It first collects features by $\boldsymbol{e}_{u,v} \cdot \boldsymbol{W}_{u,v} \boldsymbol{x}_v$ from its contextual joints, then aggregates them using the sum operator $\sum$ and finally uses multilayer perceptron (MLP) f to update the joint of interest.

The difference between FCN, GNN and LCN lies in how to compute $\boldsymbol{e}_{u,v}$ and $\boldsymbol{W}_{u,v}$. *FCN* [21] does not use human graph when collecting features. Instead, $\boldsymbol{e}_{u,v}$ is set to be one for every joint pair (J_u, J_v). In contrast, in *GNN* [40] and *LCN* [10], $\boldsymbol{e}_{u,v}$ is set with special consideration. Generally, $\boldsymbol{e}_{u,v}$ is non-zero only when the two joints are connected according to the human graph. In other words, they only collect features from contextual joints. So their main difference lies in the collection step. Please refer to [10] for more details.

2.4. General Formula

We introduce a general context modeling formula, which updates features $\boldsymbol{y}_u$ of joint J_u by

$$\boldsymbol{y}_u = f(\boldsymbol{x}_u, \text{ AGG}(\{ \phi(\boldsymbol{x}_v, \boldsymbol{e}_{u,v}) \mid \forall (J_u, J_v) \in \mathcal{E}\})), \quad (3)$$

where $\boldsymbol{x}_u$ denotes the features of joint J_u before updating, and $\boldsymbol{e}_{u,v}$ encodes the spatial relation prior (*e.g*. limb length) between J_u and J_v. There are three steps in the formula as will be detailed in the following.

1. Collection For each joint of interest, it collects features from its contextual joints as represented by $\phi(\cdot, \cdot)$ in the formula. This is the most complex step in context modeling which determines where and how to collect features from the graph nodes.

2. Aggregation This is denoted by $\text{AGG}(\cdot)$ in the formula. It is a permutation invariant function, *e.g*. sum or product function, defined on a set of contextual features. It aims to aggregate the collected features.

3. Update This is denoted by $f(\cdot, \cdot)$ in the formula. It updates the feature of a joint by transforming its own as well as the aggregated features.

It is straightforward to verify that both PSM and GNN can be interpreted by the formula. The advantage of PSM is that it can explicitly enforce limb length constraints while

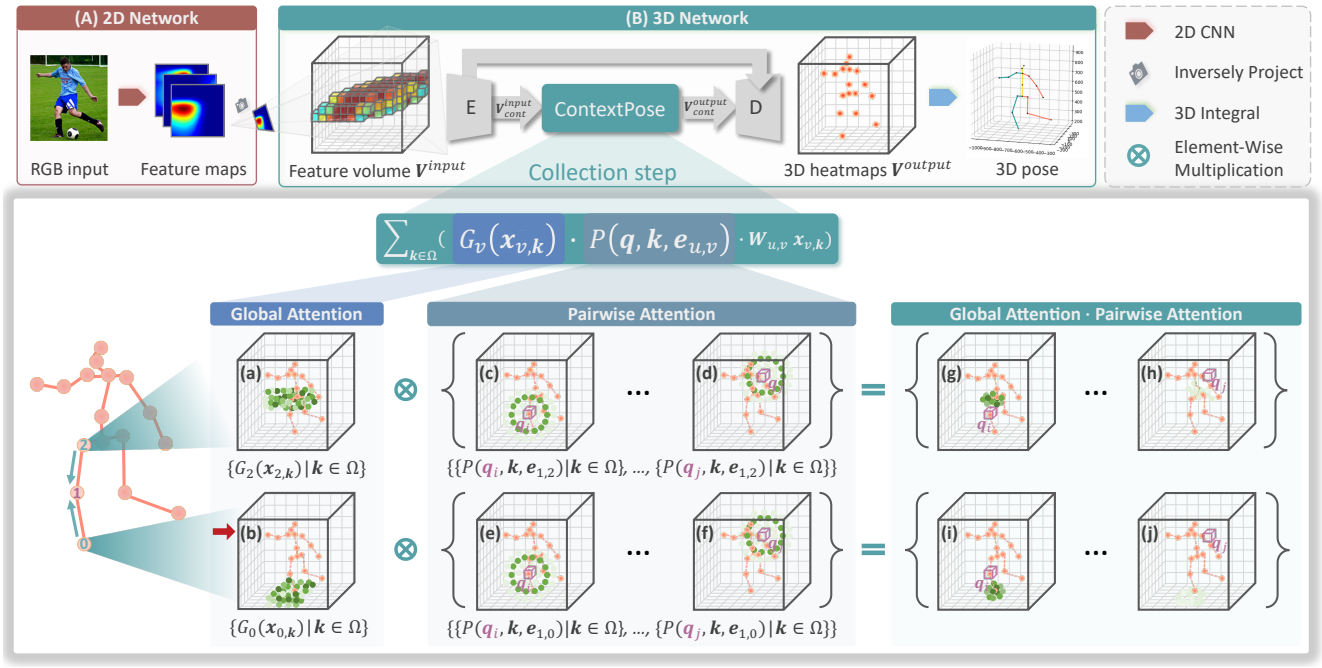

Figure 2: An example pipeline of using ContextPose for 3D pose estimation. The bottom shows how ContextPose collects features from contextual joints based on global and pairwise attention. Global attention, for example G_0 in **(b)**, represents the likelihood of J_0 at each voxel $\boldsymbol{k}$. For each voxel $\boldsymbol{q}$ of joint J_1, for example $\boldsymbol{q}_i$ in **(e)** or $\boldsymbol{q}_j$ in **(f)**, pairwise attention $P(\boldsymbol{q},\boldsymbol{k},\boldsymbol{e}_{1,0})$ traverses every voxel $\boldsymbol{k}$ of joint J_0 and computes a spatial compatibility score between $\boldsymbol{q}$ and $\boldsymbol{k}$. The product of global attention and pairwise attention gives the weight in linear combination as shown in **(i)**-**(j)**.

GNN can learn implicit priors from a large amount of data. In the following, we present an approach to combine their advantages on top of the general formula.

3. ContextPose

This section introduces the details of *ContextPose*. We first present an overview of how it can be leveraged by an existing method [14] to estimate 3D human pose in Section 3.1. Then we dive into the technical and training details of ContextPose in the following three sub-sections. Finally, we discuss the differences between ContextPose and other context modeling methods in Section 3.5.

3.1. Architecture Overview

We adopt the state-of-the-art 3D pose estimator [14] as our baseline. As shown in Figure 2, it first constructs a 3D feature volume by inversely projecting image features to the 3D space using camera parameters. Then the feature volume is fed to an encoder-decoder network to estimate 3D heatmaps. In particular, it predicts N scores for each voxel representing the likelihood of N joints. Finally, we compute expectation over the 3D heatmaps of each joint to obtain its 3D location [31]. ContextPose is inserted between the encoder and decoder network.

3.2. ContextPose

Denote the input tensor of ContextPose as $\boldsymbol{V}^{input}_{cont} \in \mathcal{R}^{NM\times D\times H\times W}$ which represents the features of N joints at $D\times H\times W$ voxels. We split $\boldsymbol{V}^{input}_{cont}$ into N groups along the channel dimension such that each group corresponds to the features of one joint. Inspired by the attention mechanism [34], ContextPose updates the features of a joint J_u at voxel $\boldsymbol{q}$ by a linear combination of the features of its contextual joints at all voxels

$$\boldsymbol{y}_{u,\boldsymbol{q}} = \boldsymbol{x}_{u,\boldsymbol{q}} + \sum_{J_v\in\mathcal{J}}[\sum_{\boldsymbol{k}\in\Omega}(G_v(\boldsymbol{x}_{v,\boldsymbol{k}}) \cdot P(\boldsymbol{q},\boldsymbol{k},\boldsymbol{e}_{u,v}) \cdot \boldsymbol{W}_{u,v}\boldsymbol{x}_{v,\boldsymbol{k}})], \tag{4}$$

where Ω denotes the set of voxels, $\boldsymbol{x}_{v,\boldsymbol{k}} \in \mathcal{R}^M$ denotes the features of joint J_v at voxel $\boldsymbol{k}$. The global attention $G_v(\boldsymbol{x}_{v,\boldsymbol{k}})$ and pairwise attention $P(\boldsymbol{q},\boldsymbol{k},\boldsymbol{e}_{u,v})$ determines the weight in linear combination. $\boldsymbol{W}_{u,v} \in \mathcal{R}^{M\times M}$ is a learnable matrix to transform features.

Global Attention (GA) We estimate a confidence score for each joint J_v at a voxel $\boldsymbol{k}$ representing to what extent should this feature contribute to other joints. Intuitively, we expect a lower score for non-person voxels in order to reduce the risk of corrupting good features. In other words, we expect large scores for voxels that are likely to include joint J_v. As

a result, joint J_u can focus on features from high likelihood voxels of joint J_v (see Figure 2 **(a)** and **(b)**). The GA for joint J_v is defined as

$$G_v(\boldsymbol{x}_{v,\boldsymbol{k}}) \propto exp(\boldsymbol{d}_v^T \boldsymbol{x}_{v,\boldsymbol{k}}), \tag{5}$$

which is normalized such that $\sum_{\boldsymbol{k}\in\Omega} G_v(\boldsymbol{x}_{v,\boldsymbol{k}}) = 1$. $\boldsymbol{d}_v \in \mathcal{R}^M$ is a learnable vector.

Pairwise Attention (PA) PA explores spatial relation between a pair of joints. The general idea is to give larger weights to features passed from locations of a joint that satisfy the pre-defined spatial relation. In this work, we focus on limb length constraints. But this can be extended to other priors such as limb orientations. If joint J_v is connected to J_u by a rigid bone, then their distance in the 3D space is fixed for the same person which is independent of human postures. Offline, we compute the average distance $\mu_{u,v}$ and the standard deviation $\sigma_{u,v}$ in the training set as the limb length distribution prior and let $\boldsymbol{e}_{u,v} = (\mu_{u,v}, \sigma_{u,v})$ as the limb pre-defined parameters. The pairwise attention for the joint pair is defined as

$$P(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v}) \propto exp(-\frac{(||\boldsymbol{q}-\boldsymbol{k}||_2 - \mu_{u,v})^2}{2\alpha\sigma_{u,v}^2 + \epsilon}). \tag{6}$$

The pairwise attention is normalized over all voxels such that $\sum_{\boldsymbol{k}\in\Omega} G_v(\boldsymbol{x}_{v,\boldsymbol{k}}) \cdot P(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v}) = 1$. The hyper-parameter α is used to adjust the tolerance to limb length errors, which is empirically set to be 1500 in this work. The parameter ϵ is used to improve numerical robustness. See Figure 2 **(c)-(f)**. Besides, if joint J_v is not connected to J_u by a rigid bone, the features from joint J_v may also be helpful to J_u. For example, left hand may also help the detection of right hand. In this case, we simply set the pairwise term to be $P(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{e}_{u,v}) = 1$ and completely rely on the global attention to determine the weights.

3.3. Regression of 3D Human Pose

The decoder network transforms the output of ContextPose $\boldsymbol{V}_{cont}^{output} \in \mathcal{R}^{NM\times D\times H\times W}$ to 3D heatmaps $\boldsymbol{V}^{output} \in \mathcal{R}^{N\times D'\times H'\times W'}$ of N body joints which represents the likelihood of each joint at each location. Then the 3D location $\boldsymbol{J}_u$ for joint J_u is obtained by computing the expectation of $\boldsymbol{V}_u^{output} \in \mathcal{R}^{D'\times H'\times W'}$ with the common integral technique [31] according to the following formula

$$\boldsymbol{J}_u = \sum_{x=1}^{D'}\sum_{y=1}^{H'}\sum_{z=1}^{W'}(x,y,z)\cdot \boldsymbol{V}_u^{output}(x,y,z). \tag{7}$$

3.4. Training

The parameters in ContextPose are jointly learned with the 2D CNN and the encoder-decoder network by enforcing two losses:

$$\mathcal{L} = \mathcal{L}_{3D} + \lambda\mathcal{L}_{GA}, \tag{8}$$

in which $\mathcal{L}_{3D}$ and $\mathcal{L}_{GA}$ are the loss functions enforced on the 3D joint locations and global attention maps, respectively.

Same as [14], we compute the L_1 loss between the ground-truth 3D pose $\boldsymbol{J}^{gt}$ and the estimated 3D pose $\boldsymbol{J}$ with a weak heatmap regularizer which promotes Gaussian shape distribution for the estimated 3D heatmaps as

$$\mathcal{L}_{3D} = \frac{1}{N}\sum_{J_u\in\mathcal{J}}(||\boldsymbol{J}_u - \boldsymbol{J}_u^{gt}||_1 - \beta\cdot log(\boldsymbol{V}_u^{output}(\boldsymbol{J}_u^{gt}))). \tag{9}$$

In addition, to help the GA focus on the voxels that are likely to have joint J_u, we enforce an L_2 loss:

$$\mathcal{L}_{GA} = \frac{1}{NDHW}\sum_{J_u\in\mathcal{J}}||\boldsymbol{G}_u - \boldsymbol{G}_u^{gt}||_2^2, \tag{10}$$

where $\boldsymbol{G}_u \in \mathcal{R}^{D\times H\times W}$ is the GA map for joint J_u and $\boldsymbol{G}_u^{gt} \in \mathcal{R}^{D\times H\times W}$ is the ground-truth heatmap generated by applying a 3D Gaussian centered at the ground truth location of the joint J_u.

In our experiment, we set β and λ to be 10^{-2} and 10^6.

3.5. Comparison of PSM, GNN and ContextPose

It is easy to verify that PSM, GNN, and ContextPose are all special cases of the general formula Eq. (3). The main difference between them lies in the *collection* step which includes the structures of human graph $\mathcal{G}$, pairwise relation $\boldsymbol{e}_{u,v}$, the collection function $\phi(\cdot,\cdot)$, and training scheme. We will compare them side by side from the above aspects hoping to clearly understand their advantages and disadvantages.

Graph Structures PSM often uses acyclic graphs in order to get optimum solution. In contrast, ContextPose is not subject to this restriction. Cyclic graph offers greater flexibility to represent more powerful and natural context. For example, in ContextPose, we can add connections between left and right shoulders to the human graph and require that they cannot be at the same location which helps solve the "double counting" problem. We can even add connections between joints in neighboring frames to promote smoothness in future work. GNN can also use cyclic graphs but it cannot explicitly express and enforce natural rules on the joints. It is not clear what kind of pairwise relation does GNN learns from data which makes it a black box.

Pairwise Relation In PSM, the pairwise relation is often implemented as limb length constraints. As discussed in Eq. (1), it encourages detections of a pair of joints that satisfy the limb length prior. In GNN, the pairwise term reflects the similarity between the features of two nodes. Although

the features also encode some location information, it is hardly possible that GNN will implicitly learn limb length constraints. ContextPose does not enforce hard limb length constraints as PSM. But it encourages pose estimates to have reasonable limb length by focusing on features that are passed between locations that satisfy limb length constraints.

End-to-End Learning PSM requires solving a discrete optimization problem in order to obtain optimal locations for all joints. In particular, it uses the argmax operator to identify optimal voxels for each joint which makes the approach non-differentiable. In contrast, the GNN-based methods can be trained end-to-end because all operators in the collection, aggregation and update functions are differentiable. ContextPose can also be trained end-to-end which combines the advantages of PSM and GNN.

Quantization Error The PSM-based methods and ContextPose both work on discrete voxels. So their accuracy depends on the size of each voxel. Using a smaller voxel decreases quantization error but meanwhile increases computation time. In [14], the authors propose to compute expectation over the heatmaps to obtain continuous 3D locations which notably decreases the impact of quantization.

4. Experiments

4.1. Datasets

Human3.6M (H36M) [13] Following [10], we use the subjects S1, S5, S6, S7, and S8 for training, and S9, S11 for testing. The Mean Per Joint Position Error (MPJPE) metric is computed under two protocols: Protocol #1 computes MPJPE between the ground-truth (GT) and the estimated 3D poses after aligning their root (mid-hip) joints; Protocol #2 reports MPJPE after the 3D estimate is aligned with the GT via a rigid transformation. Additionally, we present two new metrics to comprehensively measure the quality of the 3D pose estimates: (1) Mean Per Limb Length Error (MPLLE) computes the average limb length error between the GT and estimated poses over 16 limbs (*i.e*. the purple edges in Figure 1), and (2) Mean Per Limb Angle Error (MPLAE) measures the average limb angle error between the GT and the estimated poses.

MPI-INF-3DHP (3DHP) [22] This dataset provides monocular videos of six subjects acting in three different scenes which include green screen indoor scenes, indoor scenes and outdoor scenes. This dataset is often used to evaluate the generalization performance of different models. Following the convention, we directly apply our model trained on the H36M dataset to this dataset without re-training. We report results using two metrics: Percentage of Correctly estimated Keypoints (PCK) [2] and Area Under the Curve (AUC) [22].

4.2. Implementation Details

We use the state-of-the-art 3D pose estimator [14] as our baseline to estimate 3D poses. We insert ContextPose between the encoder and decoder networks as shown in Figure 2. To reduce GPU memory cost, we decrease the number of layers in the 3D network from five to two. The modification slightly improves the results of the baseline. For the ContextPose network, M is set to be 3. We jointly train the 2D and 3D networks for 30 epochs with the Adam [17] optimizer. The learning rates are set to be 0.0001 and 0.001 for the 2D and 3D networks, respectively. To prevent from over-fitting to the human appearance in the H36M dataset, we fix the 2D network and train the 3D network for 20 epochs before end-to-end training.

4.3. Comparison to the State-of-the-arts

Results on the H36M Dataset Table 2 shows the results of the state-of-the-art methods on the H36M dataset. Our approach outperforms the state-of-the-art methods by a notable margin under both protocols. This includes methods that explore temporal information in videos (labeled by * in the table). In particular, our method outperforms PSM [28], FCN [21], GNN [40], and LCN [10] by an even larger margin which validates the effectiveness of our context modeling strategy. We discover in our experiment that PSM [28] gets very poor results in the monocular setting. To investigate the reasons, we project the estimated 3D poses back to 2D images and find that, for most cases, the projections perfectly match the 2D people although their 3D estimates are very different from the GT poses. We show an example in Figure 3. This is mainly because PSM alone has limited capability to resolve ambiguity. Note that a 3D pose estimate may be inaccurate even when its limb lengths are correct. In contrast, the deep learning-based methods such as GNN [10, 21, 40] have strong capability to reduce ambiguity because they can fit a large amount of data. We will discuss in more details on why our approach gets more accurate estimates than PSM and GNN in the subsequent ablative study.

Results on the 3DHP Dataset Table 3 shows the results of different methods on the 3DHP dataset. Our approach achieves significantly better PCK and AUC scores than other methods including FCN, LCN, and PSM for almost all scenes. The result suggests that ContextPose has strong generalization performance which we think is due to the leverage of limb length priors in deep networks. FCN [21] gets a low accuracy because the dense connections degrade the generalization capability which has already been discussed in [10]. LCN [10] gets better results by fusing features of contextual joints but it is still worse than ours. The result validates the importance of combining deep networks and limb length priors.

Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Zhou *et al.* [41] ICCV'17	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.2	66.1	51.4	63.2	55.3	64.9
Martinez *et al.* (**FCN**) [21] ICCV'17	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Pavlakos *et al.* [25] CVPR'18	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Yang *et al.* [39] CVPR'18	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Zhao *et al.* (**GNN**) [40] CVPR'19	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	**55.0**	67.8	61.0	42.1	60.6	45.3	57.6
Qiu *et al.* (**PSM**) [28] ICCV'19	223.1	231.8	273.0	237.3	248.1	243.9	209.0	279.7	280.9	296.3	241.9	234.0	230.8	217.8	220.4	244.8
Iskakov *et al.* [14] ICCV'19	41.9	49.2	46.9	47.6	50.7	57.9	41.2	50.9	57.3	74.9	48.6	44.3	**41.3**	52.8	42.7	49.9
Wang *et al.* [36] ICCV'19	44.7	48.9	47.0	49.0	56.4	67.7	48.7	47.0	63.0	78.1	51.1	50.1	54.5	40.1	43.0	52.6
Ci *et al.* (**LCN**) [10] ICCV'19	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Pavllo* *et al.* [27] CVPR'19	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8*
Cai* *et al.* [6] ICCV'19	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	53.5	39.2	41.2	50.6*
Xu* *et al.* [38] CVPR'20	40.6	47.1	45.7	46.6	50.7	63.1	45.0	47.7	56.3	63.9	49.4	46.5	51.9	38.1	42.3	49.2*
Ours	**36.3**	**42.8**	**39.5**	**40.0**	**43.9**	**48.8**	**36.7**	**44.0**	**51.0**	63.1	**44.3**	**40.6**	44.4	**34.9**	**36.7**	**43.4**
Protocol #2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Martinez *et al.* (**FCN**) [21] ICCV'17	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Pavlakos *et al.* [25] CVPR'18	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Yang *et al.* [39] CVPR'18	**26.9**	**30.9**	36.3	39.9	43.9	47.4	28.8	**29.4**	**36.9**	58.4	41.5	**30.5**	**29.5**	42.5	32.2	37.7
Qiu *et al.* (**PSM**) [28] ICCV'19	117.0	123.2	128.0	121.7	126.1	128.7	105.3	130.1	145.1	170.2	125.1	114.5	128.9	115.3	117.1	126.7
Wang *et al.* [36] ICCV'19	33.6	38.1	37.6	38.5	43.4	48.8	36.0	35.7	51.1	63.1	41.0	38.6	40.9	30.3	34.1	40.7
Ci *et al.* (**LCN**) [10] ICCV'19	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
Pavllo* *et al.* [27] CVPR'19	36.0	38.7	38.0	41.7	40.1	45.9	37.1	35.4	46.8	53.4	41.4	36.9	43.1	30.3	34.8	40.0*
Cai* *et al.* [6] ICCV'19	36.8	38.7	38.2	41.7	40.7	46.8	37.9	35.6	47.6	**51.7**	41.3	36.8	42.7	31.0	34.7	40.2*
Xu* *et al.* [38] CVPR'20	33.6	37.4	37.0	37.6	39.2	46.4	34.3	35.4	45.1	52.1	40.1	35.5	42.1	29.8	35.3	38.9*
Ours	30.5	34.9	**32.0**	**32.2**	**35.0**	**37.8**	**28.6**	32.6	40.8	52.0	**35.0**	31.9	35.6	**26.6**	**28.5**	**34.6**

Table 2: The MPJPE (mm) of the state-of-the-art methods on the H36M dataset under protocol #1 and protocol #2, respectively. * means the method uses temporal information in videos.

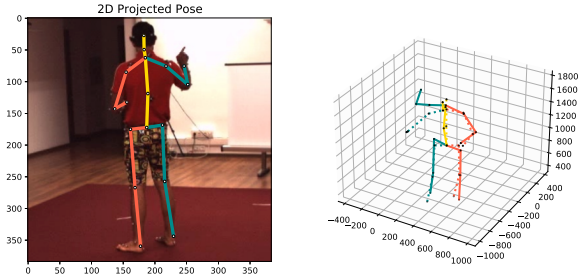

Figure 3: Visualization of a 3D pose estimated by PSM [28]. The left figure shows the projection of the estimated 3D pose. The right figure shows the estimated (solid lines) and GT (dashed lines) 3D poses. The estimated 3D pose has correct 2D projection but it is very different from GT 3D pose. It means PSM suffers from severe ambiguity when it is used in the monocular setting.

Method	GS (PCK)	noGS (PCK)	Outdoor (PCK)	ALL (PCK) ↑	ALL (AUC) ↑
Trained on: H36M+MPII [2]					
Zhou *et al.* [41]	71.1	64.7	72.7	69.2	32.5
Yang *et al.* [39]	-	-	-	69.0	32.0
Wang *et al.* [36]	-	-	-	71.9	35.8
Trained on: H36M+MPII+LSP [15]					
Pavlakos *et al.* [25]	76.5	63.1	77.5	71.9	35.3
Trained on: H36M					
Martinez *et al.* (**FCN**) [21]	49.8	42.5	31.2	42.5	17.0
Qiu *et al.* (**PSM**) [28]	26.4	22.6	19.6	23.3	8.0
Ci *et al.* (**LCN**) [10]	74.8	70.8	**77.3**	74.0	36.7
Baseline	75.2	73.3	62.2	71.3	35.0
Ours	**82.6**	**80.5**	**77.3**	**80.5**	**42.7**

Table 3: The results of the state-of-the-art methods on the 3DHP dataset. GS represents the green screen background scene. The results of [21] are taken from [20].

4.4. Ablation Study

Effect of ContextPose We first compare our approach to the baseline w/o ContextPose. The results on the H36M dataset are shown in Table 4. We can see that ContextPose notably decreases MPJPE of the baseline from 54.38mm to 50.24mm on the challenging subject S9. MPLLE decreases by nearly 6% meaning that the limb lengths of the estimated 3D poses are more accurate than the baseline. The improvement for S11 in terms of MPJPE is marginal because the baseline is already very accurate. However, we can see that there is still clear improvement in terms of limb lengths and angles. The result of the baseline is different from the number in Table 2 because we use a smaller 3D network in Table 4 to reduce memory usage as stated in Section 4.2.

Method	GA	PA	S9			S11		
			MPJPE ↓	MPLLE ↓	MPLAE ↓	MPJPE ↓	MPLLE ↓	MPLAE ↓
Baseline	✗	✗	54.38	15.03	0.1600	35.12	10.16	0.1250
Ours w/o PA	✓	✗	52.00	14.58	0.1517	35.16	9.78	0.1240
Ours w/o GA	✗	✓	52.46	14.16	0.1524	34.98	9.67	0.1224
Ours	✓	✓	**50.24**	**14.13**	**0.1509**	**34.10**	**9.50**	**0.1217**

Table 4: Ablative study on the global attention and pairwise attention in ContextPose. We show the MPJPE (mm), MPLLE (mm) and MPLAE (radian) on each test subject separately. ContextPose achieves large improvement on the more challenging subject of S9.

We plot the MPLLE of the baseline and our method for each sample in H36M dataset in Figure 4. We can see that ContextPose gets smaller errors than baseline for about 80% of the test data. In particular, the improvement is larger for hard cases where the baseline gets large errors (see the left side of the figure). It indicates that ContextPose reduces the chance of getting absurd poses by exploring context. There are few cases where ContextPose gets worse results. This usually happens when multiple body joints are occluded which makes estimating global attention a very challenging task.

Table 3 shows the results on the 3DHP dataset. We can see that using ContextPose significantly improves the PCK of the baseline from 71.3% to 80.5%. The result represents that ContextPose is very important to improve the generalization performance of the 3D pose estimator. This is a big advantage for actual deployment. In fact, we can see that our approach even outperforms the methods which use even more training data.

Effect of GA and PA We report results when we add one of the two modules (GA and PA) to the baseline in Table 4. Adding only the GA module makes little difference on the ultimate results measured by MPJPE, MPLLE, and MPLAE. In contrast, if we add the PA module, the results are improved by a notable margin which validates the importance of pairwise compatibility in context modeling.

4.5. Qualitative Results

Figure 5 shows some 3D poses estimated by ContextPose. The last four columns show the predicted weights (*i.e.* the product of the GA and PA) for some random joints. In the first case of (a), the approach pays more attention to the features around the right knee when estimating the right ankle. Similarly, in the third case of (b), it focuses on features from right elbow when estimating right wrist. We show two failure cases in row (d) and (e). In particular, in (e) our estimate has correct limb lengths but inaccurate limb angles for the left leg. In addition, the projection of the 3D pose is also reasonable. This is a common error for monocular 3D pose estimation because it has severe ambiguity.

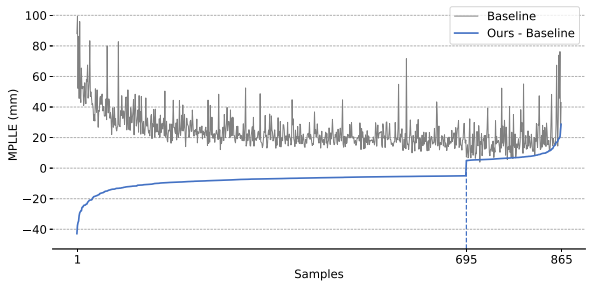

Figure 4: MPLLE (mm) of individual samples. The gray line shows the errors of the baseline. The blue line represents the error difference between ContextPose and baseline (below zero means our method gets smaller error).

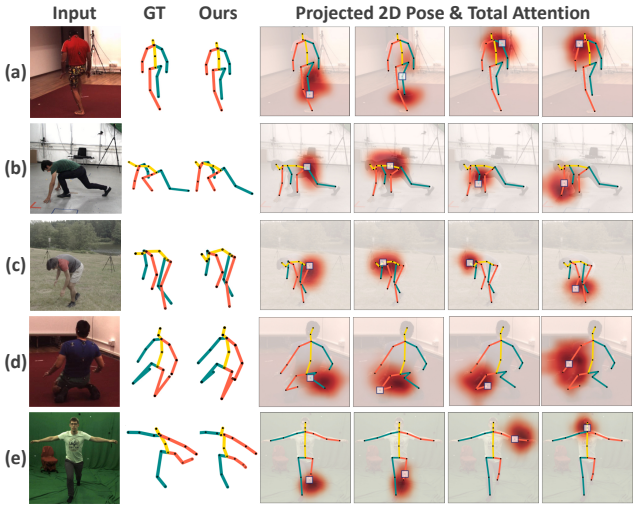

Figure 5: Example 3D pose estimates. The last four columns show the projected 2D poses and the weights in linear combination for some random joints (highlighted by small blue boxes). Row (d) and (e) show two failure cases.

5. Conclusion

We first introduce a general formula for context modeling in 3D pose estimation which allows comparing PSM and GNN side by side. Based on the formula, we present ContextPose that combines their advantages which allows enforcing limb length constraints in deep networks. So it can be trained end-to-end on large data. The approach outperforms the state-of-the-art methods on two benchmarks, and more importantly, shows better generalization performance on unseen datasets.

Acknowledgement

This work was supported in part by National Key R&D Program of China (2018YFB1403900), NSFC-61625201 and NSFC-62061136001.

References

[1] Sikandar Amin, Mykhaylo Andriluka, Marcus Rohrbach, and Bernt Schiele. Multi-view pictorial structures for 3d human pose estimation. In *BMVC*, volume 1. Citeseer, 2013. 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, June 2014. 6, 7

[3] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, pages 1669–1676, 2014. 2

[4] Irving Biederman, Robert J Mezzanotte, and Jan C Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982. 1

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, pages 561–578. Springer, 2016. 1

[6] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, pages 2272–2281, 2019. 7

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017. 1

[8] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, pages 1736–1744, 2014. 2

[9] Kiam Choo and David J Fleet. People tracking using hybrid monte carlo filtering. In *ICCV*, volume 2, pages 321–328. IEEE, 2001. 1

[10] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, pages 2262–2271, October 2019. 1, 2, 3, 6, 7

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, 2016. 2

[12] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, pages 2282–2292, 2019. 1

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

[14] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *ICCV*, pages 7718–7727, October 2019. 2, 4, 5, 6, 7

[15] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010. doi:10.5244/C.24.12. 7

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 1

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[18] Ilya Kostrikov and Juergen Gall. Depth sweep regression forests for estimating 3D human pose from images. In *BMVC*, volume 1, page 5, 2014. 2

[19] Mun Wai Lee and Isaac Cohen. Human upper body pose estimation in static images. In *ECCV*, pages 126–138. Springer, 2004. 1

[20] Chenxu Luo, Xiao Chu, and Alan Yuille. Orinet: A fully convolutional network for 3d human pose estimation. page 92, 2018. 7

[21] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, pages 2640–2649, 2017. 1, 2, 3, 6, 7

[22] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, pages 506–516. IEEE, 2017. 6

[23] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *TOG*, 36(4):1–14, 2017. 1

[24] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016. 1

[25] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, pages 7307–7316, 2018. 7

[26] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3D human pose annotations. In *CVPR*, pages 1253–1262, 2017. 2, 3

[27] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pages 7753–7762, 2019. 7

[28] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *ICCV*, pages 4342–4351, 2019. 2, 3, 6, 7

[29] Cristian Sminchisescu and Bill Triggs. Covariance scaled sampling for monocular 3d body tracking. In *CVPR*, volume 1, pages I–I. IEEE, 2001. 1

[30] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, June 2019. 1

[31] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 529–545, 2018. 1, 4, 5

[32] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, pages 1799–1807, 2014. 1

[33] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *ECCV*, pages 197–212, 2020. 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 2, 4

[35] Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L Yuille, and Wen Gao. Robust estimation of 3d human poses from a single image. In *CVPR*, pages 2361–2368, 2014. 1

[36] J Wang, S Huang, X Wang, and D Tao. Not all parts are created equal: 3d human pose estimation by modeling bi-directional dependencies of body parts. In *ICCV*, pages 7771–7780, 2019. 7

[37] Xiaolin K Wei and Jinxiang Chai. Modeling 3d human poses from uncalibrated monocular images. In *ICCV*, pages 1873–1880. IEEE, 2009. 1

[38] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *CVPR*, pages 899–908, June 2020. 7

[39] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, pages 5255–5264, June 2018. 7

[40] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pages 3425–3435, 2019. 2, 3, 6, 7

[41] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, pages 398–407, 2017. 7

[42] Xiaowei Zhou, Spyridon Leonardos, Xiaoyan Hu, and Kostas Daniilidis. 3d shape estimation from 2d landmarks: A convex relaxation approach. In *CVPR*, pages 4447–4455, 2015. 1