CrossFormer: Cross Spatio-Temporal Transformer for 3D Human Pose Estimation

Mohammed Hassanin[1], Abdelwahed Khamiss[2], Mohammed Bennamoun[3], Farid Boussaid[3], and Ibrahim Radwan[1]

[1] Canberra University
d.mfawzy@gmail.com
[2] CSIRO
[3] The University of Western Australia

Abstract. 3D human pose estimation can be handled by encoding the geometric dependencies between the body parts and enforcing the kinematic constraints. Recently, Transformer has been adopted to encode the long-range dependencies between the joints in the spatial and temporal domains. While they had shown excellence in long-range dependencies, studies have noted the need for improving the locality of vision Transformers. In this direction, we propose a novel pose estimation Transformer featuring rich representations of body joints critical for capturing subtle changes across frames (*i.e.*, inter-feature representation). Specifically, through two novel interaction modules; Cross-Joint Interaction and Cross-Frame Interaction, the model explicitly encodes the local and global dependencies between the body joints. The proposed architecture achieved state-of-the-art performance on two popular 3D human pose estimation datasets, Human3.6 and MPI-INF-3DHP. In particular, our proposed CrossFormer method boosts performance by 0.9% and 0.3%, compared to the closest counterpart, PoseFormer, using the detected 2D poses and ground-truth settings respectively.[4]

1 Introduction

Automatic reconstruction of the 3D human pose from 2D images is a fundamental problem in computer vision. 3D human pose estimation solution provides a geometric representation that is important to many applications including human-computer interaction [13], [21], action understanding [28], [14], healthcare [1], and motion analysis [2]. The recently developed solutions to this problem can be categorised into two main groups:**(1)** Two-stage approaches such as [50] and [4], where the input is firstly extracted using 2D human pose estimation (HPE) architectures (for instance, [7], [15], [39]), and **(2)** End-to-end methods [36], [34], where 3D reconstruction is inferred directly from input images or videos. Owing to the recent advances in the area of 2D pose detectors, the two-stage approaches currently outperform their end-to-end counterparts.

Despite research spanning decades, 3D human pose estimation remains very challenging. It is an ill-posed problem caused by the ambiguity and high degree of freedom in the input space [38]. Tackling the problem requires accounting the associated challenges. 2D to 3D pose regression is an under determined problem where many 3D poses

---

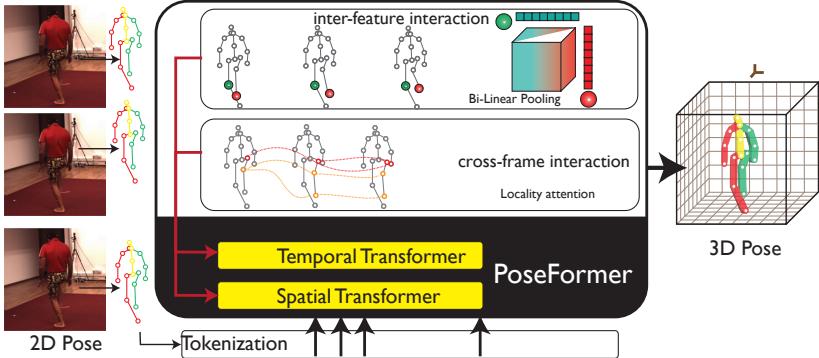[4] Codes and models will be publicly available on github.com

Fig. 1: **Conceptual Illustration of CrossFromer Architecture**. This augments the PoseFormer[51]'s architecture with the proposed cross-feature (joints) and cross-frame interaction modules.

might correspond to **almost identical** 2D projections. In this setting, even the slightest changes in joints positions and appearances can be informative. Given this, the problem calls for two natural requirements for successful regression. First, capturing rich per-joint feature representations to help mitigate the ambiguity and improve the accuracy (`R1:cross-feature interaction`). A potentially promising direction is lever-aging cross-joints features correlations at a detailed level. Second, leveraging information across the Spatio-Temporal steam by attending carefully to features most relevant to the preformed pose (`R2:cross-frame interaction`). In this work, we draw on the recent advances of Vision Transformer and design an explicit mechanism for meeting those requirements.

The recent advent of transformers [40] have progressed many visual recognition tasks. Transformers have been used to encode the long-range relationships between in-put tokens. As pose estimation is one of the fundamental computer vision problems, it has been approached by many Transformer-based architecture. Yet, one of most notable addition to this line of work is PoseFromer, [51]. Despite its great success, whose core is ViT [11], it inherits some of the ViT limitations pointed out in the literature [25], Namely, poor locality. This issue manifests itself in the fact that attention module at-tends to all tokens. While this design contributed to Transformer excellence in natural language domains, in vision applications it desirable attend to local information. As this limitation can be linked to R2, one can expect an improvement in PoseFormer by addressing it. More superficially, in this paper, we address the following question: *is it possible to improve Pose Transformers by improving locality and inter-feature repre-sentations?*.

To answer the above question, we propose to integrate locality and rich inter-features interaction (as in Fig. 1), while retaining the key advantages of the original PoseFormer[51] (*i.e.*, capacity to handle large number of tokens and Spatio-Temporal modelling). To this end, we design novel interaction modules to account for the above requirements as follows:

`R1)` To capture rich feature that highlights potentially feeble but effective details, we further integrate **Bi-linear Pooling** [48] in the locality attention module by modify-

ing the cross terms in the attention using outer product. Hence , expanding the attention to all channels (unlike the original inner product that merges information across channels dimension). Bi-linear Pooling was originally motivated by a similar goals of a fine-grained visual classification and has demonstrated success in many applications [47] from fine-grained recognition to semantic segmentation and video classification.

R2) There is a growing research that thrives for improving locality of Transformers using various approaches such as local attention [10] and regional attention maps [5]. Our work shares the same motivation, albeit using novel methodology. Inspired by non-local Neural Networks [43], [48], we opt for **locality attention** [5] to leverage the feature representations of the joints across frames. This generalises the vanilla self-attention [43], module [41], and can be interpreted as seeking a favourable middle ground between the locality-insensitive approaches (vanilla self-attention) and the purely local (stationary convolutional) approaches.

To summarise, we combine the locality and inter-feature interaction in a transformer-based approach for 3D pose estimation. Our architecture realises the needed requirements and provides two novel cross interaction modules to encode both the local and global dependencies. More specifically, a cross-joint interaction (CJI) module is plugged in the spatial encoder of the Transformer architecture to encode the kinematic constraints between the body parts within a frame. This module (see Sec. 3.2) is composed of depth-wise convolutions followed by group normalization and non-linearity layer (GELU). In addition to the cross-joint interaction module, we also propose a cross-frame interaction (CFI) module to handle interactions between the joints across frames. As opposed to PoseFormer [51], where the inner-product is used to compute the correlation between frames, the CFI module explicitly learns the correlations between the frames by using the outer-product between feature representations across the frames. This helps in turn model the fine-grained temporal dynamics of the body parts.

Experiments were performed on Human3.6 [19] and MPI-INF-3DHP [31] datasets. Reported results demonstrate the superiority of proposed method over the state-of-the-art. Moreover, qualitative comparisons show that our method is efficient in capturing hardly-visible body parts.

The main contributions of this paper are :

- A cross-joint interaction module CJI for spatial transformer architectures to encode the kinematic dependencies between body joints while taking into account the local connections of each joint.
- A cross-frame interaction module CFI for temporal transformer architectures to capture the explicit correlations between body joints across frames.
- State-of-the-art performance achieved on two popular benchmark datasets; Human3.6 and MPI-INF-3DHP.

---

[5] originally named "non-local operation" [43] after the "non-local" mean operation [3] and to set it apart for the local convolution. Here, we use "locality attention" to signify its role in our architecture and avoid confusion.

2   Related Work

3D Human pose estimation methods are commonly used as the second stage to 2D detection methods. First, the input image is passed to detection frameworks [15], [8] to infer 2D poses. Then, 2D poses are lifted to 3D using other methods [51], [37]. Martinez *et al.* [30] used a fully-connected residual network to predict the 3D poses. Fang *et al.* [12] lifted to 3D poses using a grammar model for body joints configuration. Several other methods used the temporal information to overcome the occlusion in the input images [24], [37]. Pavllo *et al.* [37] proposed a dilated temporal convolution approach to exploit the temporal information. Cai *et al.* [4] used a graph method to choose the center frame and then refine the final estimated 3D pose. Wang *et al.* [42] customized graph convolution network (GCN) in a U-shape as they involved motion modeling to learn the 3D poses. In [50], a variant of a non-local module is customized to include the semantics of the input images.

Recently, vision transformers advanced all the visual recognition tasks [40]. Following [11], transformer has been used to lift 2D poses to the corresponding 3D poses. In [27], Lin *et al.* used convolutions and transformer together without temporal information to predict 3D poses. In order to eliminate the redundancy in the sequence with temporal information, Li *et al.* [24] proposed a strided transformer network. In [51], spatial-temporal transformer is used for 3D HPE tasks. Using transformers in HPE showed significant improvement overall. However, a pre-training on a large dataset is required to learn more representative and effective representations for the sequence HPE data. Our proposed method is different from the previous methods in leveraging the cross-interaction between the joints of the body parts in the spatial and the temporal domains.

3   Method

This section presents the proposed architecture to estimate 3D human pose from 2D. Inspired by recently developed transformer approach, namely, Poseformer [51], we propose interaction modules inside the spatial and temporal encoders to make the transformer more efficient when lifting the 2D to 3D poses. The 2D input poses can be inferred from any 2D pose detection approach such as [7], [15]. The poses of the consecutive frames in an input video are concatenated to form the input to the proposed architecture. Suppose, $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ denotes the set of the 2D input frames, where $\mathbf{x}_i \in \mathbb{R}^{J \times 2}$ is composed of the 2D positions of the body joints for frame i, N is the total number of frames in the input video and J is the number of joints. The output of each frame is the 3D body joints, $\mathbf{y} \in \mathbb{R}^{J \times 3}$. The proposed architecture incorporates cross-interaction modules into a vanilla spatial and temporal transformer [11]. Incorporating these modules with the transformer helps to capture both long-range relationships and local interactions between the body joints and across frames in both the spatial and temporal domains, respectively.
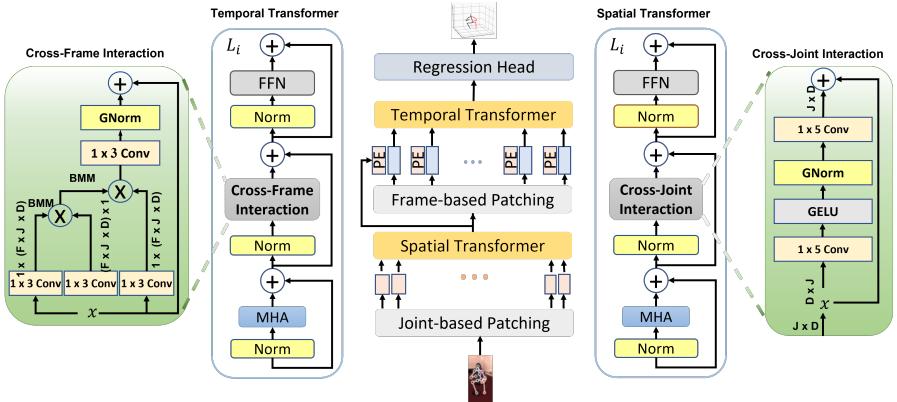
Fig. 2: The proposed architecture is composed of two main modules: a spatial transformer along with the Cross-Joint Interaction module (CJI), and a temporal transformer with the proposed Cross-Frame Interaction (CFI).

3.1 Transformers

Despite the great success achieved by transformers in computer vision tasks [11], they focus mostly on global dependencies between frames in the input sequence (as observed by [51]). Motivated by this, we augment the proposed transformer with modules to capture more joints-related context information within-frame and corss-frames. Below we formulate our problem and review vision Transformer.

The input corresponds to the 2D poses in subsequent frames $\{\mathbf{x}_i \in \mathbb{R}^{J \times 2} | i = 1, ..., N\}$. The initial layer of the transformer embeds the high dimensional features of each patch and their positional information. This step is called *patch embedding*, and is achieved by the following projection operation:

$$Z_0 = [\mathbf{x}_1 E; \mathbf{x}_2 E; ..., \mathbf{x}_P E],$$
$$E \in \mathbb{R}^{(J \times 2) \times D}, Z_0 \in \mathbb{R}^{P \times D} \tag{1}$$

where D is the embedding dimension, P is the number of patches that is equivalent to the number of input body joints in the case of spatial processing and to the number of frames in the case of temporal processing.

The output of the patch embedding step, Z_0, is then passed into the most important step of the transformer, *i.e.* the *self-attention*. It implicitly correlates the input patches in the form of attention scores. This step is mainly based on mapping three learnable weight matrices $Q, K, V \in \mathbb{R}^{P \times D}$ to attend for the output features.

The self-attention operation is computed as a scaled dot-product between these matrices as follows:

$$A(Q, K, V) = \text{Softmax}(\frac{Q.K^T}{\sqrt{P}}).V \tag{2}$$

This attention operation is applied through a multi-head attention (MHA), which combines various representations with different positions in parallel. The multi-head atten-

tion operation is simply achieved by concatenating all heads as follows:

$$MHA = Concat(A_j(.)).W, \quad j \in 1, ..., H \tag{3}$$

where W is a learnable weight matrix and H is the number of heads.

The self-attention module is then combined with other layers such as layer normalisation [44] and multi-layer perceptron (MLP). The steps in a transformer layer (l), which are following the patch embedding step can be listed, in general, as follows:

$$
\begin{aligned}
Z_l &= MHA(LN(Z_{l-1})) + Z_{l-1}, \\
Z_l &= MLP(LN(Z_l)) + Z_l, \\
Z_l &= LN(Z_l), \\
where &\quad l = 1, 2, ..., L
\end{aligned}
\tag{4}
$$

where $LN(.)$ represents the layer normalization and L is the layer indicator of the transformer. Both the spatial and temporal parts of the transformer consist of identical layers. However, the output feature space for the spatial encoders $\in \mathbb{R}^{J \times D}$ and for the temporal encoders is in $\mathbb{R}^{F \times D}$, where F is number of frames in the input patch. The final layer is another linear projection step that maps the output space to $\mathbb{R}^{J \times 3}$ for each frame.

3.2 Spatial Interaction

The spatial encoders of the transformer learn the relationships between the body joints within the frame. The input is represented as $\mathbf{x} \in \mathbb{R}^{J \times 2}$, where each joint is considered as an independent patch, and the output is the feature representation of each joint with respect to the other joints. Inspired by [51], the 2D coordinates of each joint are firstly transformed using a linear operation. The output of this step, $Z_0 \in \mathbb{R}^{P \times D}$ is then passed forward to compute the self-attention scores. This encodes the dependencies between the different joints. However, these operations disregard the low-score relationships. This is due to the non-local nature of the transformer operations listed in Eq. 4.

In order to tackle this issue, we propose the Cross-Joints Interaction (CJI) module that we integrate inside the spatial encoders with an aim to achieve two characteristics; **1)** getting the transformer to consider the locality nature of the human body parts and their local interactions the same as encoding their non-local interactions (*i.e.* long-range dependencies), and **2)** explicitly encoding the interaction between the joints of the body parts across the channels, which enriches the representation of the joints with low attention scores. Achieving these two characteristics improves the scores of the Multi-Head Attention for the 3D human pose estimation task.

Cross-Joints Interaction (CJI) Module
This module is inserted between the MHA layer and MLP for each block. The CJI module consists of two depth-wise convolutions with kernel size 5, group normalization and non-linearity GELU. Also, the residual connection is added to the output of the module to avoid overfitting. The operations within the CJI module are performed using the

sequence of the following operations on outpout of the patch embedding step Z_0:

$$Z = CONV(GN(GELU(CONV(Z)))) + Z \tag{5}$$

where $GELU$ refers to the non-linear layer in [16], $CONV$ is the standard convolution layer with kernel 5 and GN indicates the group normalization used in [44]. Since the focus of the CJI module is on the cross-interaction between the joints, the output of the MHA part in Eq. 4 has been transposed. That is, it becomes $Z_0 \in \mathbb{R}^{D \times P}$. The spatial encoders for a transformer layer l can then be represented by the following list of operations:

$$\begin{aligned}
Z_l &= MHA(LN(Z_{l-1})) + Z_{l-1}, \\
Z_l &= CONV(GN(GELU(CONV(Z_l)))) + Z_l \\
Z_l &= MLP(LN(Z_l)) + Z_l \\
Z_l &= LN(Z_l),
\end{aligned} \tag{6}$$

3.3   Temporal Interaction

In contrast to the spatial encoders, which encode the long-range dependencies between the joints within each frame, the temporal counterparts aim at learning rich representations across frames. These encoders are stacked on top of the spatial ones. Their input is firstly flattened from $\mathbb{R}^{J \times D}$ to $\mathbb{R}^{1 \times (J \times D)}$ leading to $\mathbb{R}^{F \times (J \times D)}$ for all frames in a patch. Similar to the vanilla transformers [40], the temporal positional information are added to the input space. Apart from that, the remaining settings including self-attention modules and MLP blocks are just as in the case of spatial encoders. The input size of the transformer is maintained all over the transformers layers, which is $\mathbb{R}^{F \times (J \times D)}$.

Cross-Frames Interaction (CFI) Module

The whole attention in temporal encoders is based on the dependencies between the channels (*i.e.*, $\mathbb{R}^{f \times (J \times D)}$), where the explicit interaction between the frames is disregarded. Depending on the scores produced by SoftMax calculation, joints with low scores might be ignored in the process. For example, partially visible or occluded joint won't be properly represented and reflected in the 3D space. In order to resolve this problem, we propose a Cross-Frame Interaction Module, CFI, to explicitly encode the relationships between the same joint across frames using Bi-Linear Pooling operations [48] instead of the traditional SoftMax. This helps in learning the correlations between the channels explicitly as in [18] and reflects the the kinematic constraints on the output space.

Briefly, the Bi-Linear Pooling learns pairwise feature correlations using the outer product. Each element of the correlation matrix $C_{ij} = \sum_F Z_i Z_j$ is a Bi-Linear product of the corresponding embedded features of frames i and j and then it is sum-pooled, where $Z_i \in \mathbb{R}^{J \times D}$ is the input feature of frame i. More precisely, the input is transformed by combining the positional information with the frames where $Z \in \mathbb{R}^{F \times (J \times D)}$ and then using convolutions we extract K, Q, and V such that:

$$K = ZW_k, \quad Q = ZW_q, \quad V = ZW_v \tag{7}$$

The bilinear matrix multiplication is performed between matrices Q, K and V as follows:

$$
\begin{aligned}
C &= K \otimes Q \quad \in \mathbb{R}^{F \times F} \\
Z &= C \otimes V \quad \in \mathbb{R}^{D \times F}
\end{aligned}
\tag{8}
$$

where $\otimes$ refers to the bilinear pooling operation. Then, the output is added to the input after performing convolution and group normalization. Compared to self-attention modules in Equation 4, CFI module uses a bilinear pooling to learn pairwise interactions between the same joint across different frames. This highlights the discriminativeness of each frame which leads to rich representation. For example, one frame will focus on the top joints of the body and another on the lower part, while CFI will focus on combining both parts. Finally, CFI is merged with temporal transformer between MHA layer and MLP blocks. The updated sequence of the temporal transformer operations for a layer l can be listed as follows:

$$
\begin{aligned}
Z_l &= MHA(LN(Z_{l-1})) + Z_{l-1}, \\
K &= Z_l W_k, Q = Z_l W_q, V = Z_l W_v \\
Z_l &= GN(CONV(((K \otimes Q) \otimes V))) + Z_l \\
Z_l &= MLP(LN(Z_l)) + Z_l \\
Z_l &= LN(Z_l),
\end{aligned}
\tag{9}
$$

The sequence of transformer encoders is combined in a compact form, which enables an end-to-end training. Moreover, following the vanilla transformers [51] on using the three embedding matrices with the input *i.e.*, Q, K, and V allows CJI and CFI to serve as generalised modules, which can be plugged in many other transformer architectures for other various visual recognition tasks.

3.4   Regression Head

The spatial and temporal transformers are stacked together as in Fig. 2, where their input is passed to the spatial encoders and then to the temporal ones. The output of the temporal transformer is $\mathbb{R}^{F \times (J \times D)}$, which requires to be reduced to $\mathbb{R}^{1 \times (J \times 3)}$. First, 1D convolution is applied as a weighted average for the frames to transform to $\mathbb{R}^{1 \times (J \times D)}$. Then, a linear layer is used to learn the 3D geometries from the D dimension followed by the normlisation layer. The final output is the estimated 3D position for each joint $\mathbb{R}^{1 \times (J \times 3)}$.

3.5   Loss function

Following the recent work in [37], the MPJPE loss function is employed to optimise the parameters of the whole architecture:

$$\mathcal{L} = \frac{1}{J} \sum_{k=1}^{J} \|g_k - p_k\|_2, \tag{10}$$

where g_k represents the ground-truth 3D joint position of joint k and p_k is the 3D output of the proposed architecture of the k-th joint.

Table 1: Comparison between our proposed method and the state-of-the art approaches for 3D human pose estimation. Mean Per Joint Position Error(MPJPE) is used to measure the mean error between the estimated 3D pose and the ground truth 3D pose on Human3.6M under Protocols 1&2 where 2D pose detection is used as input. The top shows results of Protocol 1 (MPJPE), whereas the bottom part shows the results of Protocol 2 (P-MPJPE). f refers to the number of frames used in each method, $*$ denotes that the input 2D pose detection method used is the cascaded pyramid network (CPN), and † refers to a transformer-based model. (Red: best; Blue: second best)

Protocol 1		Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
Debral et al. [9]	ECCV'18	44.8	50.4	44.7	49.0	52.9	61.4	43.5	45.5	63.1	87.3	51.7	48.5	52.2	37.6	41.9	52.1
Cai et al. [4] ($f = 7$)	ICCV'19	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Pavllo et al. [37] ($f = 243$)*	CVPR'19	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Lin et al. [26]($f = 50$)	BMVC'19	42.5	44.8	42.6	44.2	48.5	57.1	52.6	41.4	56.5	64.5	47.4	43.0	48.1	33.0	35.1	46.6
Yeh et al. [46]	NIPS'19	44.8	46.1	43.3	46.4	49.0	55.2	44.6	44.0	58.3	62.7	47.1	43.9	48.6	32.7	33.3	46.7
Liu et al. [29] ($f = 243$)*	CVPR'20	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
SRNet [49] *	ECCV'20	46.6	47.1	43.9	41.6	45.8	49.6	46.5	40.0	53.4	61.1	46.1	42.6	43.1	31.5	32.6	44.8
UGCN [42] (f = 96)	ECCV'20	41.3	43.9	44.0	42.2	48.0	57.1	42.2	43.2	57.3	61.3	47.0	43.5	47.0	32.6	31.8	45.6
METRO [27] (f = 1) †	CVPR'21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.0
PoseFormer (no PT) [51] (81)	ICCV'21	43.0	46.5	41.4	44.1	48.1	53.2	43.7	43.6	54.9	62.3	47.1	44.9	47.7	32.8	33.5	45.7
Chen et al. [6] (f = 81)*	TCSVT'21	42.1	43.8	41.0	43.8	46.1	53.5	42.4	43.1	53.9	60.5	45.7	42.1	46.2	32.3	33.8	44.6
PoseFormer (PT)[51] (81)	ICCV'21	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
CrossFormer (81)		40.7	44.1	40.8	41.5	45.8	52.8	41.2	40.8	55.3	61.9	44.9	41.8	44.6	29.2	31.1	43.7
Protocol 2		Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Average
Pavlakos et al. [36]	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
Hossain et al. []	ECCV'18	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Cai et al. [4] (f = 7)	ICCV'19	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	32.3	39.0
Lin et al. [26] (f = 50)	BMVC'19	32.5	35.3	34.3	36.2	37.8	43.0	33.0	32.2	45.7	51.8	38.4	32.8	37.5	25.8	28.9	36.8
Pavllo et al. [37] (f = 243)*	CVPR'19	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Liu et al. [29] (f = 243)*	CVPR'20	32.3	35.2	33.3	35.8	35.9	41.5	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	35.6
UGCN [42] (f = 96)	ECCV'20	32.9	35.2	35.6	34.4	36.4	42.7	31.2	32.5	45.6	50.2	37.3	32.8	36.3	26.0	23.9	35.5
Chen et al. [6] (f = 81)*	TCSVT'21	33.1	35.3	33.4	35.9	36.1	41.7	32.8	33.3	42.6	49.4	37.0	32.7	36.5	25.5	27.9	35.6
PoseFormer (no PT) [51] (f=81)	ICCV'21	33.5	35.6	33.5	35.6	36.1	40.4	32.8	32.5	43.5	49.3	35.4	33.2	36.3	25.3	26.6	35.3
PoseFormer [51](f=81)	ICCV'21	32.5	34.8	32.6	34.6	35.3	39.5	32.1	32.0	42.8	48.5	34.8	32.4	35.3	24.5	26.0	34.6
CrossFormer (f=81)		31.4	34.6	32.6	33.7	34.3	39.7	31.6	31.0	44.3	49.3	35.9	31.3	34.4	23.4	25.5	34.3

4 Experiments

In this section, we provide empirical experiments to show the significance of our proposed method. First, we describe the used datasets, the evaluation criteria and protocols. Then, further experiments are conducted along with ablation studies. Finally, we provide the comparisons with state-of-the-art methods.

4.1 Datasets and Evaluation Protocols

Datasets Our experiments are evaluated on the most popular datasets for HPE tasks, Human3.6 [19] and MPI-INF-3DHP [31].
Human3.6 dataset is the most popular dataset and the largest one for HPE. It consists of 3.6 million images in the form of video frames. It includes seven subjects and a total of 15 actions including "walking", "sitting" and "posing". Each video is captured from 4 different views. 3D annotations are provided by an accurate marker-based motion capture. The subjects are split for training and testing as in [37], where S1, S5, S6, S7and S8 are used for training and S1 and S11 for testing. One model is used to train all the frames for the various actions. All the videos are recorded in indoor scenes.
MPI-INF-3DHP dataset contains 8 actions from 14 different views which result to

input   GT   Poseformer   CrossFormer   input   GT   Poseformer   CrossFormer
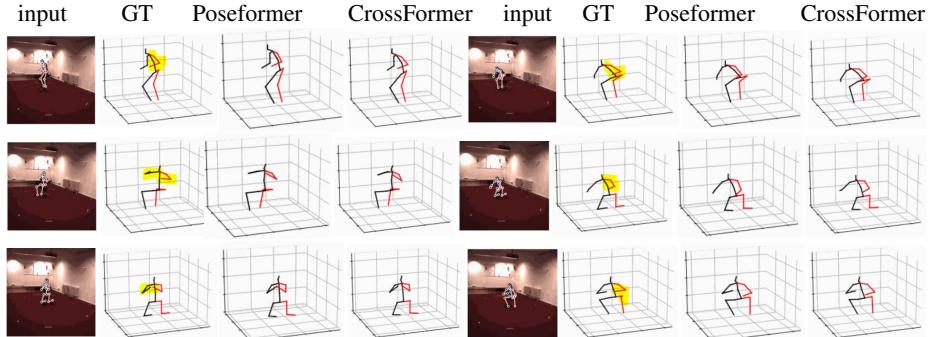

Fig. 3: Visual qualitative comparison of the proposed method (CrossFormer), the ground-truth and the state-of-the art approach (PoseFormer) [51]. The experiments are conducted on Human3.6M test set S11 with the "SittingDown" action. The blue arrows on the ground-truth highlight the locations where our method clearly behaves better.

in diverse poses. It contains indoor and outdoor complex scenes and thus it is more challenging for HPE tasks. The test set includes 6 various subjects. Without stated, the settings are following [36]. The scenarios of test set are: studio with a Green Screen (GS), studio without Green Screen (noGS) and outdoor scene (Outdoor).

Evaluation Protocols:

Human3.6 dataset evaluation for 3D pose estimation performance relative to the 3D ground-truth follow [37] using the most common metrics, *i.e.* MPJPE and P-MPJPE. MPJPE (Mean Per Joint Position Error) refers to the average of Euclidean distance in millimeters between the predicted 3D human-body joints and the ground-truth ones. It is also denoted by Protocol 1. For P-MPJPE, it calculates the Euclidean distance between the 3D predicted pose after rigid alignment and the ground-truth. It is referred to Protocol 2 as it is more robust to prediction failure of the joint individuals.

For MPI-INF-3DHP, Protocol 1 is used, Area Under Curve (AUC), and Percentage of Correct Keypoint (PCK) within the 150mm as defined in [31]

Table 2: Comparison between the estimated 3D pose of the proposed method and the ground truth 3D pose on Human3.6M dataset using the Mean Per Joint Position Error under Protocol 1 (MPJPE). All methods use the ground truth 2D pose as input. (Red: best; Blue: second best)

Protocol #1		Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.* [30]	ICCV'17	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Lee *et al.* [22]	ECCV'18	32.1	36.6	34.3	37.8	44.5	49.9	40.9	36.2	44.1	45.6	35.3	35.9	30.3	37.6	35.5	38.4
Pavllo *et al.* [37]	CVPR'19	35.2	40.2	32.7	35.7	38.2	45.5	40.6	36.1	48.8	47.3	37.8	39.7	38.7	27.8	29.5	37.8
Cai *et al.* [4] f = 243	ICCV'19	32.9	38.7	32.9	37.0	37.3	44.8	38.7	36.1	41.0	45.6	36.8	37.7	37.7	29.5	31.6	37.2
Xu *et al.* [45]	CVPR'21	35.8	38.1	31.0	35.3	35.8	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	30.7	35.8
Liu *et al.* [29] (f=243)	CVPR'20	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Chen *et al.* [6] (f=243)	TCSVT'21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
SRNet [49]	ECCV'20	34.8	32.1	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
PoseFormer [51] (f=81)	ICCV '21	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
CrossFormer (f=81)		26.0	30.0	26.8	26.2	28.0	31.0	30.4	29.6	35.4	37.1	28.4	27.3	26.7	20.5	19.9	28.3

4.2   Implementation Details

We used Pytorch [35] to implement our proposed method. Two Tesla A100 GPU 40 GB each were used to run the experiments. Adam optimizer [20] is chosen to train the model for 100 epochs, decaying with 10%. Another decaying schedule is used with initial learning rate of 0.0001 and an exponential decaying factor of 0.99 after each epoch, whereas the batch size is 512. We follow [37] for the selection of the 2D pose detector, which is the cascaded pyramid network (CPN) [8] on Human3.6 dataset. For MPI-INF-3DHP dataset, 2D ground-truth poses are used as in [37]. Both of the spatial and temporal transformers have 4 layers, and the multi-head attention has 8 heads. The dimension of the features is 32 for spatial transformer and 544 for the temporal one. The receptive fields are 9, 27 and 81. Horizontal flip augmentation is used for the training and testing stages.

4.3   Comparison with the state-of-the art

Human3.6 In this part, we compare the proposed method with the state-of-the art methods on Human3.6 dataset. 15 actions have been selected from two subjects, S9 and S11, for the evaluation. In order to guarantee fair comparisons, the input is taken from CPN in the form of 2D keypoints for training and testing. Table 1 shows the comparison of the SOTA methods with the proposed method (81 frames). Overall, our method achieves the state-of-the art on Human3.6 on all the metrics and it outperforms the state-of-the art (Chen *et al.* [6]) with a considerable margin of 0.9%, 1.3% for Protocols 1 and 2, respectively. It is worth noting that the across-joint modules in the spatial and temporal cases are crucial to infer the body-joints dependencies. Comparing the proposed method with PoseFormer (with no pre-training used) shows the significance of the across-joint correlation modules. Our method outperforms with a large margin of 2% the SOTA. In terms of accuracy, we achieve 1% better than the second best accuracy. Additionally, the proposed method achieves the best performance amongst all the compared methods in protocol 2 in Table 1 (bottom). In some selected difficult poses such as walk together, walk, smoke, where the poses change very quickly, the proposed method showed a significant improvement ranging from 1.1% to 2.5% over the baseline. This highlights the ability of our method to encode the long-range interactions between the body-joints. Considering the pre-trained baseline, the proposed method achieves better performance for all the actions. These results show the importance of plugging the across-joints modules in the transformers.

Further experiments on Human3.6 using ground-truth 2D poses as input have also been performed. This shows the power of the proposed method where there is no noise in the input as in the previous case. Table 2 shows the comparisons of our method and the baselines. Overall, the proposed method achieved the best performance amongst the baselines. It achieved 28.3% MPJPE, whereas the second best approach achieves 31.3 with gain of 3%. The proposed method outperforms the baselines in all the actions with a considerable improvement range from 2.4% as the minimum difference, and 4.8% for the largest.

MPI-INF-3DHP We further compare the proposed methods (CrossFormer) to previous ones on MPP-INF-3DHP using 9 frames. This is important because it illustrates

the ability of the proposed method to train with fewer training samples in an outdoor settings. As Table 3 shows, our method obtains the best performance amongst the compared ones *w.r.t.* the metrics.

Table 3: Comparison between the proposed method (CrossFormer) and previous SOTA methods on MPI-INF-3DHP. The metrics of the comparison are the Percentage of Correct Keypoints (PCK) and Area Under the Curve (AUC). The best scores are marked in bold

		PCK ↑	AUC ↑	MPJPE ↓
Mehta *et al.* [32]	3DV'17	75.7	39.3	117.6
Mehta *et al.* [33]	ACM ToG'17	76.6	40.4	124.7
Pavllo *et al.* [37] ($f = 81$)	CVPR'19	86.0	51.9	84.0
Lin *et al.* [26] ($f = 25$)	BMVC'19	83.6	51.4	79.8
Li *et al.* [23]	CVPR'20	81.2	46.1	99.7
Chen *et al.* [6]	TCSVT'21	87.9	54.0	78.8
PoseFormer [51] (f = 9)	ICCV'21	88.6	56.4	77.1
CrossFormer (f=9)		**89.1**	**57.5**	**76.3**

Computational Complexity Analysis: Table 4 shows the comparison with different methods of complexity analysis. For the number of parameter analysis, it is relatively bigger than Poseformer in the three settings, it increases slightly. However, this increase is negligible in favor of accuracy gains. Apart from Poseformer, the number of parameters is still competitive to the other methods. It is also noticeable that the increase of frames does not translate in ab increase in the total number of parameters (only by hundreds). Regarding the FLOPs, the proposed method is not the best in comparison to the comparable methods. However, it only exhibits a slight increase over Poseformer. For frames per second (FPS), our method shows a slightly lower number compared to PoseFormer.
       Qualitative Results. In order to show the superiority of the proposed method qualitatively, we compare it with the ground-truth and Poseformer (the state-of-the art approach [51]). The evaluation is conducted on Human3.6 dataset S11 test set on "Setting-Down" action. From Fig. 3, it is clear that the proposed method is considerably better than Poseformer. We use blue arrows to define the locations where our method behaves better. While our method shows some failures, it is still overall better than Poseformer.

4.4   Ablation Study

In order to check the impact of the the proposed method individually, we perform empirical experiments on Human3.6 dataset using protocol 1. Also, one experiment is conducted on the optimal hyper-parameters selection.

The impact of cross-joints modules In this ablation study, we investigate the contribution of Cross-Joint Interaction (CJI) on the spatial transformer and on the whole network. We also study the impact of CJI with spatial embedding since it considers inductive bias implicitly and without. Regarding Cross-Frame Interaction (CFI) module,

Table 4: Comparison between the proposed method and a set of previous methods in terms of the comparison are computational complexity, number of the parameters, MPJPE, and Frames Per Second (FPS). The experiments are conducted on Human3.6M under Protocol 1 with the detected 2D pose as input.

	f	Parameters (M)	FLOPs (M)	MPJPE	FPS
Hossain and Little [17]	-	16.95	33.88	58.3	-
Chen *et al.* [6]	27	31.88	61.7	45.3	410
Chen *et al.* [6]	81	45.53	88.9	44.6	315
Chen *et al.* [6]	243	59.18	116	44.1	264
PoseFormer [51]	9	9.58	150	49.9	320
PoseFormer [51]	27	9.59	452	47.0	297
PoseFormer [51]	81	9.60	1358	44.3	269
CrossFormer	9	9.93	163	48.5	284
CrossFormer	27	9.93	515	46.5	266
CrossFormer	81	9.93	1739	43.8	241

we verify its impact on the network independently with temporal embedding and without. For a fair comparison, we use the optimal parameter settings, including 4 blocks for both transformers. The dimension of the keypoints is unified to 32 for the spatial and 544 for the temporal. Table 5 shows the results of various settings between CJI and CFI. The results illustrate that using both of the cross-joint modules improve the total performance significantly (from 49.9 to 48.5). Table 5 (row 4) discusses using CJI module along with both embeddings and without CFI. It achieved better performance than Poseformer (from 49.9 to 49.3). This explains the need of both cross-joint modules to add locality to the transformers. Similarly, using CFI only along with the embeddings achieved better performance compared to PoseFormer. However, it achieved lower performance than CJI. It is clear that using cross-joint modules without the spatial information embeddings shows no difference in terms of performance, which proves our claim of using CJI and CFI modules to add to the locality of the self-attention modules. However, CJI is more independent than CFI to positional information as it shows the same accuracy without the positional information, whereas CFI obtains 0.04% less.

Another ablation study (a table is included in the supplementary) is conducted to verify the importance of the spatial and temporal transformers. This shows the various settings of the proposed network and provides comparisons to Poseformer.

5   Conclusion

Two interaction modules have been proposed to resolve the issues of using the spatial and temporal transformers for 3D human pose estimation. The first module, cross-joint interaction (CJI), has been presented to resolve the locality issue of the spatial transformers, while the second module, cross-frame interaction (CFI), has been developed to encode the dependencies of the joints across the subsequent frames. Both of two modules have been incorporated into transformer architecture, CrossFormer, and validated on popular 3D pose datasets. The proposed method has achieved new SOTA results

Table 5: Quantitative Comparison between Poseformer and the proposed method on the impact of different components on the total performance. The experiments are performed on Human3.6M (Protocol 1) using CPN 2D pose as input. Various settings are used for cross-joint modules (CJI: Cross-Joint Interactions; CFI: Cross-Frames Interactions; E_s: spatial embedding; E_t: temporal embedding)

frames	CJI	CFI	E_s	E_t	CrossFormer
9	✗	✗	✓	✓	49.90
9	✓	✗	✗	✗	49.30
9	✓	✗	✓	✗	49.30
9	✓	✗	✓	✓	49.30
9	✓	✓	✗	✓	48.50
9	✗	✓	✗	✗	49.83
9	✗	✓	✗	✓	49.80
9	✗	✓	✓	✓	49.80
9	✓	✓	✓	✓	48.50

for both datasets. In the future, we will test on other visual recognition applications to ensure its generalisation to different visual tasks.

References

1. Md Atiqur Rahman Ahad, Anindya Das Antar, and Omar Shahid. Vision-based action understanding for assistive healthcare: A short review. In *CVPR Workshops*, pages 1–11, 2019.
2. David Ahmedt-Aristizabal, Kien Nguyen, Simon Denman, M Saquib Sarfraz, Sridha Sridharan, Sasha Dionisio, and Clinton Fookes. Vision-based mouth motion analysis in epilepsy: A 3d perspective. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1625–1629. IEEE, 2019.
3. Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.
4. 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019.
5. Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. *arXiv preprint arXiv:2106.02689*, 2021.
6. Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
7. Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.
8. Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.
9. Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–683, 2018.

10. Yifeng Ding, Zhanyu Ma, Shaoguo Wen, Jiyang Xie, Dongliang Chang, Zhongwei Si, Ming Wu, and Haibin Ling. Ap-cnn: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification. *IEEE Transactions on Image Processing*, 30:2826–2836, 2021.

11. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

12. Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

13. Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *ACM Computing Surveys (CSUR)*, 54(3):1–35, 2021.

14. Mohammed Hassanin, Ibrahim Radwan, Salman Khan, and Murat Tahtali. Learning discriminative representations for multi-label image recognition. *Journal of Visual Communication and Image Representation*, page 103448, 2022.

15. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

16. Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

17. Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018.

18. Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

19. Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

20. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

21. Branislav Kisacanin, Vladimir Pavlovic, and Thomas S Huang. *Real-time vision for human-computer interaction*. Springer Science & Business Media, 2005.

22. Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.

23. Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6173–6183, 2020.

24. Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *arXiv preprint arXiv:2103.14304*, 2021.

25. Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

26. Jiahao Lin and Gim Hee Lee. Trajectory space factorization for deep video-based 3d human pose estimation. In *BVMC*, 2019.

27. Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

28. Bangli Liu, Haibin Cai, Zhaojie Ju, and Honghai Liu. Rgb-d sensing based human action and interaction analysis: A survey. *Pattern Recognition*, 94:1–12, 2019.

29. Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2020.

30. Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

31. 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 *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017.

32. 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 *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017.

33. 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. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

34. Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 752–768. Springer, 2020.

35. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

36. Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

37. Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

38. Ibrahim Radwan, Abhinav Dhall, and Roland Goecke. Monocular image 3d human pose estimation under self-occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1888–1895, 2013.

39. Ibrahim Radwan, Nour Moustafa, Byron Keating, Kim-Kwang Raymond Choo, and Roland Goecke. Hierarchical adversarial network for human pose estimation. *IEEE Access*, 7:103619–103628, 2019.

40. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

41. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

42. Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020.

43. Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

44. Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
45. Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16105–16114, 2021.
46. Raymond Yeh, Yuan-Ting Hu, and Alexander Schwing. Chirality nets for human pose regression. *Advances in Neural Information Processing Systems*, 32:8163–8173, 2019.
47. Tan Yu, Xiaoyun Li, and Ping Li. Fast and compact bilinear pooling by shifted random maclaurin. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3243–3251, 2021.
48. Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6511–6520, 2018.
49. Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.
50. Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.
51. Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *arXiv preprint arXiv:2103.10455*, 2021.