# TransFusion: Cross-view Fusion with Transformer for 3D Human Pose Estimation

Haoyu Ma[1]
haoyum3@uci.edu
Liangjian Chen[2]
clj@fb.edu
Deying Kong[1]
deyingk@uci.edu
Zhe Wang[1]
zwang15@uci.edu
Xingwei Liu[1]
xingweil@uci.edu
Hao Tang[1]
htang6@uci.edu
Xiangyi Yan[1]
xiangyy4@uci.edu
Yusheng Xie[3]
yushx@amazon.com
Shih-Yao Lin[4]
mike.lin@ieee.org
Xiaohui Xie[1]
xhx@uci.edu

[1] University of California, Irvine
[2] Facebook
(work done outside of Facebook)
[3] Amazon
(work done outside of Amazon)
[4] R&D Center US Laboratory,
Sony Corporation of America
(work done outside of Sony)

**Abstract**

Estimating the 2D human poses in each view is typically the first step in calibrated multi-view 3D pose estimation. But the performance of 2D pose detectors suffers from challenging situations such as occlusions and oblique viewing angles. To address these challenges, previous works derive point-to-point correspondences between different views from epipolar geometry and utilize the correspondences to merge prediction heatmaps or feature representations. Instead of post-prediction merge/calibration, here we introduce a transformer framework for multi-view 3D pose estimation, aiming at directly improving individual 2D predictors by integrating information from different views. Inspired by previous multi-modal transformers, we design a unified transformer architecture, named TransFusion, to fuse cues from both current views and neighboring views. Moreover, we propose the concept of *epipolar field* to encode 3D positional information into the transformer model. The 3D position encoding guided by epipolar field provides an efficient way of encoding correspondences between pixels of different views. Experiments on Human 3.6M and Ski-Pose show that our method is more efficient and has consistent improvements compared to other fusion methods. Specifically, we achieve 25.8 mm

MPJPE on Human 3.6M with only 5M parameters on 256 × 256 resolution. Source code and trained model can be found at https://github.com/HowieMa/TransFusion-Pose.

## 1 Introduction

Estimating the 3D locations of human joints is a critical task for many AI applications such as augmented reality, virtual reality and medical diagnosis [9]. The estimation is often carried out in two common settings: One is estimating the 3D pose from monocular images [5, 6, 14, 32, 43, 50, 59, 66], and the other is estimating 3D poses from multiple cameras [7, 19, 34, 37, 49]. The former is challenging due to the ambiguity of depth estimation with only one view. The latter setting, the focus of this paper, usually obtains better 3D pose estimation performance since the multi-view settings can help resolve depth ambiguity. Most multi-view works follow a two-step pipeline that firstly estimates 2D poses in each view and then recovers 3D pose from them. However, it is still difficult to solve challenging cases such as occlusions in the first step, and the estimated 3D poses are often inaccurate as it depends on the results from the first step.

Researchers have sought to introduce the 3D information in the first step to improve the 2D pose detector, because the challenging cases in one view are potentially easier to solve in other views. Specifically, they usually fuse the features of the neighboring view (reference view) with epipolar constraints [19, 54, 60]. Although interpretable, fusing along the epipolar line only does not fully utilize the semantic information of the reference view as the information off the epipolar line is discarded. For example, it is difficult to associate the ankle with the leg from the epipolar line in the reference view of Figure 1, which could be an important cue as part of the structure information for pose estimation. On the other hand, fusing all locations of other views can address this drawback. In this paper, we propose the *Epipolar Field*, a more general form of the epipolar line. It assigns probabilities to all locations of the reference view and still keep the knowledge of epipolar constraints.

Recently, attention mechanisms and the transformers [46] achieve great progress in computer vision areas [2, 11, 29, 42, 47, 51, 63, 64]. The self-attention module [46] can capture long range dependencies and correspondences, which is difficult for the convolutional layer. Although promising, there are only a few works [29] that apply it to the 3D pose estimation tasks. To the best of our knowledge, none of the previous works have exploited the transformer architectures in the multi-view 3D pose estimation setting. Inspired by previous multi-modal transformers [22, 39, 41], we propose the *TransFusion*, a lightweight framework that can utilize all pixels from both the current view itself and reference view simultaneously. As an example in Figure 1, the attention layer actually relies on the whole leg to infer the location of the ankle. Moreover, we add the 3D geometry positional encoding based on the epipolar field to help the transformer explicitly capture the correspondence.

Our main contributions are summarized as follows:

- We are the first to apply the transformer architecture to multi-view 3D human pose estimation. We propose the TransFusion, a unified architecture to fuse cues from multiple views.
- We propose the *epipolar field*, a novel and more general form of epipolar line. It readily integrates with the transformer through our proposed geometry positional encoding to encode the 3D relationships among different views.
- Extensive experiments are conducted to demonstrate that our TransFusion outperforms previous fusion methods on both Human 3.6M and SkiPose datasets, but requires substantially fewer parameters.

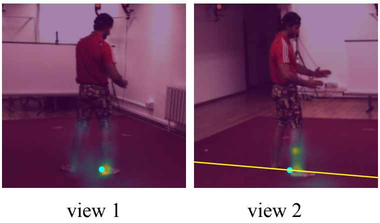


Figure 1: Comparison of epipolar line and attention module. Given the query pixel (cyan dot) in view 1 (current). The attention map on the view 2 (reference) indicates that the prediction relies on the image clues provided by the area of right shank, not just the corresponded right ankle. While previous methods based on epipolar line (yellow line) cannot capture this information.

# 2 Related Work

## 2.1 Multi-view 3D Pose Estimation

Multi-view 3D pose estimation usually follows a two-step process: (1) localize 2D joints with a 2D pose estimator on each view, and (2) lift the 2D joints from multi-view images to the 3D position via triangulation. To improve the performance of 2D pose detector, researchers typically resort to sophisticated architectures to capture both low-level and high-level representations [10, 33, 40, 52, 53] or use the structural information to model the spatial constraints [8, 24, 25, 26, 44]. However, the occlusion cases are still challenging, as monocular images do not provide evidence for occlusion joints localization.

An alternative approach, more explainable, is to make the 2D pose detector 3D-aware, *i.e.*, fusing the 2D feature heatmaps [7, 19, 34, 54, 60] from different views. Specifically, the Cross-view Fusion [34] directly learns a fixed attention weight to fuse all pairs of pixels given a pair of views. However, the learnable weight requires the multi-camera setup unchanged during the inference time, and the number of parameters is quadratic to the resolution of input images. The epipolar transformer [19] applies the non-local module [47] to obtain the weights and only fuse pixels along the epipolar line in other views. Thus it is easy to learn and flexible to use. However, sampling along the epipolar line discards off-epipolar line information and thus obtains limited information from the reference views. In the second step, researchers use graphical model with the structure of human [34] to improve the quality of triangulation or directly learn 3D pose via differentiable triangulation [21]. Our work still focuses on enhancing 2D pose by fully integrating information from different views.

## 2.2 Transformer

**Vision Transformer** Recently, several studies demonstrated that the transformer architectures [46] plays a significant role in a wide range of computer vision tasks, such as image classification [4, 11, 45], object detection [2, 64], and semantic segmentation [48, 55, 63]. Recently, some studies also explored applying the transformer on human pose estimation tasks [28, 29, 31, 56, 62]. More specifically, for 2D pose estimation, TransPose [56] aims to explain the spatial dependencies of the predicted keypoints with transformers, PRTR [28] and TF-Pose [31] attempt to directly regress the joint coordinates by transformer decoders. While for the 3D pose estimation, METRO [29] firstly applies transformer to reconstruct 3D human pose and mesh from a single image, and PoseFormer [62] builds a spatial-temporal transformers with the input of 2D joint sequences for 3D pose estimation in videos. However, previous works have hardly exploited the transformer architectures on the multi-view 3D pose estimation setting, which is however an important task in the pose estimation area.

**Multi-modal Transformer** Transformers with multi-modality inputs such as images and texts have also been fully exploited [22, 27, 39, 41, 57, 58, 65]. In general, these methods directly concatenate the embeddings from two sources together [22] and make the trans-

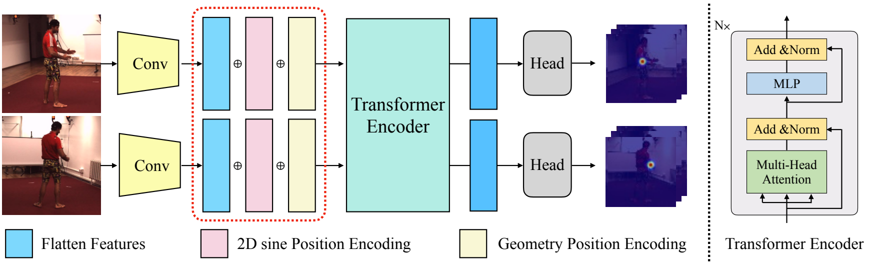


Figure 2: Overview of TransFusion.

former itself to learn the correspondence between two modalities from millions of image-text paris [36]. Thus, these methods are quite expensive and inefficient, and difficult to apply on limited datasets. Our method, however, directly provides the correspondence between two inputs and makes the transformer explicitly learn their relationships.

# 3 Methods

## 3.1 Overview

Figure 2 is an overview of TransFusion. It takes two images from different views as input, and predicts the heatmaps of joints in each view. The framework consists of three modules: a CNN backbone to extract low-level features; a transformer encoder to capture both correspondence between two views and long-range spatial correlations within single view images; a head to predict the heatmaps of joints. Specifically, given images $\mathbf{I}_i \in \mathbb{R}^{3\times H_I \times W_I}$ in each view, where $i \in \{1,2\}$ denotes view 1 and view 2, the backbone $\mathcal{F}(\cdot)$ firstly produces the low-level features $\mathbf{X}_i = \mathcal{F}(\mathbf{I}_i) \in \mathbb{R}^{d\times H\times W}$ of each image. Here $d$ is the number of channels. $H$ and $W$ are the height and width of the feature map, respectively. The feature $\mathbf{X}_i$ is flattened into a sequence vector $\mathbf{X}_i^{'} \in \mathbb{R}^{L\times d}$, where $L = H \times W$. Both 2D sine positional encoding $\mathbf{E_{2D}}$ and 3D geometry positional encoding $\mathbf{E_G}_i$ are added onto $\mathbf{X}_i^{'}$ to make the transformer aware of position information. $\mathbf{X}_1^{'}$ and $\mathbf{X}_2^{'}$ are concatenated together to build a uniform embedding $\mathbf{X} = [\mathbf{X}_1^{'} + \mathbf{E_{2D}} + \mathbf{E_G}_1, \mathbf{X}_2^{'} + \mathbf{E_{2D}} + \mathbf{E_G}_2] \in \mathbb{R}^{2L\times d}$. The embedding $\mathbf{X}$ then enters the standard transformer encoder $\mathcal{E}(\cdot)$. Finally, the output of the transformers $\tilde{\mathbf{X}} = \mathcal{E}(\mathbf{X})$ are split into $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$, which is embedding of each view, and a prediction head $\mathcal{H}(\cdot)$ takes $\tilde{\mathbf{X}}_i$ and predicts the joint heatmaps $\bar{\mathbf{H}}_i = \mathcal{H}(\tilde{\mathbf{X}}_i) \in \mathbb{R}^{k\times H_h \times W_h}$ for each view, where $k$ is the number of joints.

## 3.2 TransFusion

**Transformer Encoders** The transform encoder $\mathcal{E}(\cdot)$ consists of several layers of multi-head self-attention. Let $l = 2L$ for short, given the input sequence $\mathbf{X} \in \mathbb{R}^{l\times d}$, the self-attention layer first uses linear projections to obtain a set of queries ($\mathbf{Q} \in \mathbb{R}^{l\times d}$), keys ($\mathbf{K} \in \mathbb{R}^{l\times d}$) and values ($\mathbf{V} \in \mathbb{R}^{l\times d}$) from $\mathbf{X}$. The three linear projections are parameterized by three learnable matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d\times d}$. Following [2], the position encoding $\mathbf{E}$ is added into the input $\mathbf{X}$ for computing the query and key. The scaled dot-product attention [46] between $\mathbf{Q}$ and $\mathbf{K}$ is adopted to compute the attention weights, and aggregate the values:

$$\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{1}$$

Finally, a non-linear transformation (*i.e.*, multi layer perceptron, and the skip connection) is applied on $\mathbf{A}$ to calculate the output $\tilde{\mathbf{X}}$. As $\mathbf{X}$ is low-level features of all views, given one query pixel on the feature map, it can attend cues from the its own view and other views simultaneously through the entire network.

**Positional encoding** The attention layer would degenerate into a permutation-equivariant architecture without any position information. Thus, the positional encoding is necessary to make the transformer aware of position and order of input sequence. For each individual view, we follow the 2D sine positional encoding in the original transformers [11, 56], and we denote it as $\mathbf{E_{2D}}$. However, it only encodes position information from its own view, while the position information in the 3D space and that from the reference views cannot be encoded. Thus, another positional encoding $\mathbf{E_G}_i$ (See Section 3.3) is required to encode the 3D location information of each view $i$ in the 3D space.

## 3.3 Geometry Position Encoding (GPE)

To make the transformers 3D-aware, we introduce 3D camera information [1, 61] into the positional encoding and propose the *Geometry Positional Encoding*. Denote the world coordinate system as $\mathbf{O}_{\text{world}}$. The 3D location of view $i$ 's camera center in $\mathbf{O}_{\text{world}}$ is denoted as $\mathbf{C}_i$, and the 3D location of the $n$-th pixel ($n \in \{1, 2, ..., L\}$) of view $i$ in $\mathbf{O}_{\text{world}}$ is denoted as $\mathbf{P}_i^n$. $\mathbf{P}_i^n$ can be derived from the camera parameters of view $i$. As shown in Figure 3, the ray $\overrightarrow{\mathbf{C}_i\mathbf{P}_i^n}$ (gray line with arrow) indicates the direction of the pixel $p_i$ in the world. The unit vector $\widehat{\overrightarrow{\mathbf{C}_i\mathbf{P}_i^n}}$ is its direction vector, and can encode the relative 3D location of each pixel. Thus, we design GPE based on this unit vector, and we add one linear transformation to make it fit the input dimension $d$. The 3D geometry positional encoding for the $n$-th pixel in view $i$ is defined as:

$$\mathbf{E_G}_i^n = \mathbf{W}_e \widehat{\overrightarrow{\mathbf{C}_i\mathbf{P}_i^n}} \in \mathbb{R}^d \tag{2}$$

Where $\mathbf{W}_e \in \mathbb{R}^{d\times 3}$ is a learnable transformation matrix. With $\mathbf{E_G}_i$, the transformer can be aware of the 3D location of each view.

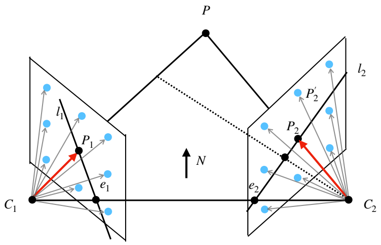


Figure 3: Illustration of Geometry Positional Encoding. For each pixel, its geometry positional encoding is calculated from the unit vector $\hat{\mathbf{u}}_\mathbf{i}^\mathbf{n}$ of ray $\mathbf{C}_i\mathbf{P}_i$ in the world coordinate $\mathbf{O}_{\text{world}}$.

## 3.4 Epipolar Field

Although GPE impose the 3D space information into transformers, it does not explicitly encode the relationship between two views. As a result, given a pixel in the current view, it is still difficult to attend the corresponding regions when performing global attention between features of two views. We further impose the *Epipolar Constraints* [1] into GPE: Given one pixel $p_1$ in view 1, its correspondence pixel $p_2$ in view 2 must be on the epipolar line $l_2$ (Figure 3). However, the epipolar line does not model the relationship with pixels outside $l_2$. Instead, pixels close to the line and pixels away from the line should be treated differently.

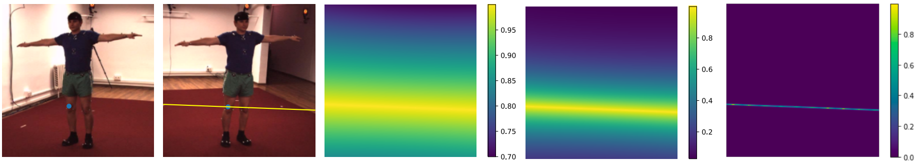
Thus, we propose the *Epipolar Field* to model the relationship among all pixels in the reference view. In detail, given $\mathbf{P}_1^n$, the 3D location of $n$-th pixel $p_1^n$ in view 1, we calculate the normal vector $\mathbf{N}_{\mathbf{P}_1^n\mathbf{C}_1\mathbf{C}_2}$ of plane $\mathbf{P}_1^n\mathbf{C}_1\mathbf{C}_2$ by:

$$\mathbf{N}_{\mathbf{P}_1^n\mathbf{C}_1\mathbf{C}_2} = \overrightarrow{\mathbf{C}_1\mathbf{C}_2} \times \overrightarrow{\mathbf{C}_1\mathbf{P}_1^n} \tag{3}$$

Given $\mathbf{P}_2^m$, the 3D location of $m$-th pixel $p_2^m$ in view 2, we use the angle $\theta$ between the normal vector $\mathbf{N}_{\mathbf{P}_1^n\mathbf{C}_1\mathbf{C}_2}$ and ray $\overrightarrow{\mathbf{C}_2\mathbf{P}_2^m}$ to model the relationship between $p_1^n$ and $p_2^m$, and use the cosine of $\theta$ to calculate the correspondence score:

$$\mathbf{S}(p_1^n, p_2^m) = 1 - \mid \cos\theta \mid = 1 - \mid \mathbf{N}_{\mathbf{P}_1^n\mathbf{C}_1\mathbf{C}_2} \cdot \overrightarrow{\mathbf{C}_2\mathbf{P}_2^m} \mid \tag{4}$$

The absolute $|\cdot|$ is added to limit the score in $[0,1]$. With Eq. 4, if $p_2^m$ falls in the epipolar line $l_2$, the score $\mathbf{S}(p_1^n, p_2^m)$ will be 1. Otherwise, the far $p_2^m$ is from $l_2$, the closer the score would be 0. We further add a soft factor $\gamma$ to control the sharpness, thus the epipolar field is $\mathbf{S}'(p_1^n, p_2^m) = (\mathbf{S}(p_1^n, p_2^m))^\gamma$. Figure 4 gives a visualization of the epipolar field. Comparing with the epipolar line, the epipolar field model relationships with all pixels in the reference view. We can also reduce it to the epipolar line with a very large $\gamma$. Thus, epipolar field can be considered as a more general form of the epipolar line.



Query Pixel | Epipolar Line | Epipolar Field ($\gamma = 1$) | Epipolar Field ($\gamma = 10$) | Epipolar Field ($\gamma = 1000$)

Figure 4: Illustration of Epipolar Field. The epipolar field can reflect the distance from the epipolar line to the off-line pixels. By adjusting the soft factor $\gamma$, it can also reduce to the standard epipolar line.

We then use the epipolar field to guide the learning of $\mathbf{W}_e$ to help the $\mathbf{E_G}_i^n$ encode correspondence between two views. In detail, we let the dot product of $\mathbf{E_G}_1^n$ and $\mathbf{E_G}_2^m$ match $\mathbf{S}'(p_1^n, p_2^m)$ with the mean square error loss during the training process:

$$L_{\text{pos}} = \frac{1}{L^2}\sum_n^L\sum_m^L(\mathbf{E_G}_1^n(\mathbf{E_G}_2^m)^T - \mathbf{S}'(p_1^n, p_2^m))^2 \tag{5}$$

Therefore, a high attention score will be achieved along the epipolar line when calculating the cross-view attention maps between $\mathbf{X}_1^{'} + \mathbf{E_{2D}} + \mathbf{E_G}_1$ and $\mathbf{X}_2^{'} + \mathbf{E_{2D}} + \mathbf{E_G}_2$, which makes the transformer easy to attend corresponding regions. Moreover, with this soft design, semantic information from offline pixels are still kept, rather than discarded like [19].

## 3.5 Implementation Details

**CNN backbone** We follow [56] and apply a very shallow CNN architecture as the CNN backbone $\mathcal{F}(\cdot)$, which is the initial part of the ResNet-50 [18]. Specifically, the number of parameters of the shallow CNN is 1.4 M, which is just 5.5% of the original Simple Baseline with ResNet-50 (25.6M). The output feature map has size $H = H_I/8$, $W = W_I/8$. Thus, the fine-grained local feature information can still be kept.

**TransFusion** Following [2, 56], we set the dimension of the feature embedding $d$ to 256, the number of heads to 8, the number of encoder layers $N$ to 3. Due to the limitation of resource, we only consider the fusion of 2 neighborhood views, although out framework can be easily extended to more than 2 views.

**Prediction head** Given $\tilde{\mathbf{X}}_i$, we first reshape it back to $\tilde{\mathbf{X}}_i' \in \mathbb{R}^{d \times H \times W}$. The prediction head $\mathcal{H}(\cdot)$ applies one deconvolution layer and one $1 \times 1$ convolution layer to predict the heatmap of keypoints. By default, the height and width of heatmaps $\mathbf{H}_i$ are $H_h = H_I/4$ and $W_h = W_I/4$.

**Loss function** The groundtruth heatmap $\mathbf{H}_i \in \mathbb{R}^{k \times H_h \times W_h}$ of 2D keypoints is defined as a 2D a Gaussian centering around each keypoint [52]. We apply the Mean Square Error (MSE) loss to calculate the difference between the output heatmaps $\bar{\mathbf{H}}_i$ and $\mathbf{H}_i$. By combining the Equation 5, we train the network end-to-end with loss function $L = \frac{1}{HW} \parallel \bar{\mathbf{H}}_i - \mathbf{H}_i \parallel_F^2 + L_{\text{pos}}$.

# 4 Experiments

## 4.1 Experimental Settings

**Dataset** We conduct extensive experiments on two public multi-view 3D human pose estimation datasets, Human 3.6M [3, 20] and Ski-Pose [12, 13, 15, 16, 17, 35, 38]. (1) The Human 3.6M contains joint annotations of video frames captured by four calibrated cameras in a room. We adopt the same training and test split as in [19, 21, 34], where subjects 1, 5, 6, 7, 8 are used for training, and 9, 11 are for testing. Note that 3D annotations of some scenes of the 'S9' are damaged [21], we exclude these scenes from the evaluation as in [19, 21]. (2) The Ski-Pose dataset aims to help analyze skiers's giant slalom runs with 6 calibrated cameras. It provides six camera views as well as corresponding 3D pose. In detail, 8,481 frames are used for training and $1,716$ are used for testing. We resize all images to $256 \times 256$ in all experiments.

**Training** As the training of transformers requires huge datasets [11, 46], while the scenes of multi-view pose datasets are quite limited, making it difficult to train the transformer from scratch. By convention [19, 21], we use the MS-COCO [30] pretrained TransPose [56] to initialize our network and fine tune it on the multi-view human pose datasets. Following the settings in [19], we apply Adam optimizer [23] and train the model for 20 epochs. The learning rate is initialized with 0.001 and decays at 10-th and 15-th epoch with ratio 0.1.

**Evaluation metrics** The performance of 2D pose estimation is evaluated by Joint Detection Rate (JDR), which measures the percentage of the successfully detected keypoints. A keypoints is detected if the distance between the predicted location and the ground truth is within a predefined threshold. The threshold is set to half of the head size for human pose estimation. Given the estimated 2D joints of each view, following [19, 34], direct triangulation is used for estimating the 3D poses with respect to the global coordinates. The 3D pose estimation accuracy is measured by Mean Per Joint Position Error (MPJPE) between the groundtruth 3D pose and the estimated 3D pose.

## 4.2 Results on Human 3.6M

We compare with two state-of-the-art methods, the crossview fusion [34] and the epipolar transformers [19]. For fair comparison, we use the SimpleBaseline-ResNet50 pretrained on COCO [53] as initialization and then finetuned with their official codes [19, 34].

**Quantitative results** The results of both 2D and 3D pose estimation are shown in Table 1. We also shown the number of parameters of each model, the MACs (multiply-add operations). Besides, we also report the inference time to obtain the 3D pose from 4 views on a single 2080Ti GPU of all multiview methods. For both 2D and 3D pose estimation, TransFusion consistently outperforms or achieves comparable performance with epipolar transformers [19] and cross-view fusion [34]. Note that JDR is a relative loose metric, with a wider threshold which tolerates small errors, so the improvement on 2D is not very obvious. However, on the 3D metric, which directly computes the distance, our improvement is much more significant. Moreover, as in Table 2, our method can achieve significant improvement on sophisticated poses sequences such as "Phone" and "Smoke", which usually encounters heave occlusions for certain views. This result suggests that fusing features from the entire images of other views, instead of just features along the epipolar line [19], can bring more benefits. Besides, comparing to the single view TransPose [56], our Transfusion can achieve 4.7 mm gain on 3D. Thus, the improvement is not only from the TransPose architecture, but from the fusion with other views. Moreover, our method is lightweight and efficient. It only requires 2.1% (5M / 235M) of the parameters of cross-view fusion [34]. Benefit from the parallel computing of transformers architectures, it further reduces the inference time, while the operation of sampling along epipolar lines [19] is time-consuming.

| Method | Params | MACs | Inference Time (s) | JDR (%) ↑ | MPJPE (mm) ↓ |
|---|---|---|---|---|---|
| Single view - Simple Baseline[53] | 34M | 51.7G | - | 98.5 | 30.2 |
| Single view - TransPose [56] | 5M | 43.6G | - | 98.6 | 30.5 |
| Crossview Fusion [34] | 235M | 55.1G | 0.048 | **99.4** | 27.8 |
| Epipolar Transformer [19] | 34M | 51.7G | 0.086 | 98.6 | 27.1 |
| TransFusion | **5M** | 50.2G | **0.032** | **99.4** | **25.8** |

Table 1: 2D and 3D pose estimation accuracy comparison on Human3.6M. The metric of 2D pose is JDR (%), and the metric of 3D pose is MPJPE (mm). All networks are pretrained on COCO [30] and then finetuned on Human 3.6M [20]. All images are resized to 256 × 256.

| Method | Dir | Disc | Eat | Greet | Phone | Pose | Purch | Sit | SitD | Smoke | Photo | Wait | WalkD | Walk | WalkT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Crossview Fusion[34] | 24.0 | 28.8 | 25.6 | 24.5 | 28.3 | 24.4 | 26.9 | **30.7** | 34.4 | 29.0 | 32.6 | 25.1 | 24.3 | 30.8 | 24.9 |
| Epipolar transformers [19] | **23.2** | 27.1 | 23.4 | 22.4 | 32.4 | **21.4** | **22.6** | 37.3 | 35.4 | 29.0 | 27.7 | 24.2 | **21.2** | 26.6 | **22.3** |
| TransFusion | 24.4 | **26.4** | **23.4** | **21.1** | **25.2** | 23.2 | 24.7 | 33.8 | **29.8** | **26.4** | **26.8** | **24.2** | 23.2 | **26.1** | 23.3 |

Table 2: The MPJPE of each pose sequence on Human 3.6M.

**Visualization of Attention maps** Given the query pixel in one view, we further visualize the attention maps on both views. We show our results in Figure 5. It is observed that on the view itself, typically the attention map is around the joints. If the query joint is occluded, it may resort to joints on the other side of the symmetry [56]. On the neighboring view, the network usually not just attends the corresponding keypoint, but attends the whole limbs, which cannot be located by the epipolar line. Previous methods based on epipoar line [19] actually miss this important clue.

**Qualitative results** We also present examples of predicted 2D keypoints on the image and 3D pose in the space, and compare our methods with baseline methods [34]. As in Figure 6, even if the entire arms (green line) are occluded, our method still predicts the 2D keypoints correctly by fusing information from the reference view, and further gives a better 3D pose.

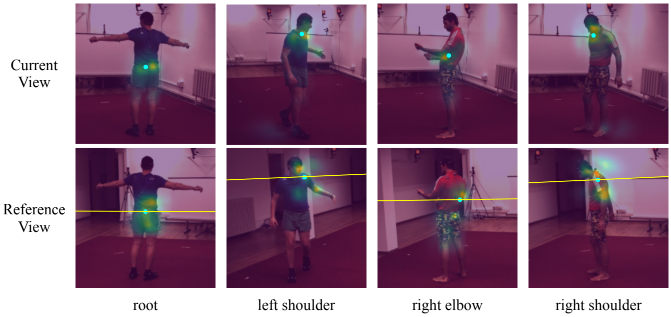


Figure 5: Visualization of attention maps on Human 3.6M test set. The cyan dots are groundtruth. Given the query pixel, the first rows are attention maps of current views, and the second rows are attention maps of reference views. We also visualize the epipolar line (yellow) for comparison.

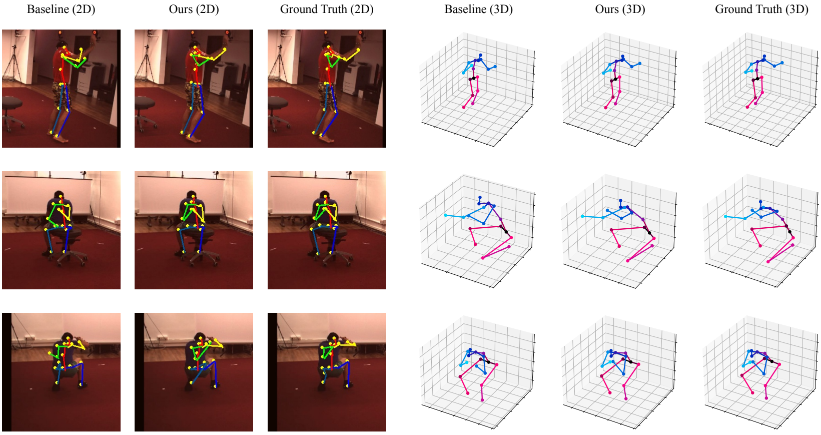


Figure 6: Visualization of predictions for Human 3.6M. Both skeletons of 2D keypoints on the image and 3D pose in the space are presented.

## 4.3 Ablation Studies

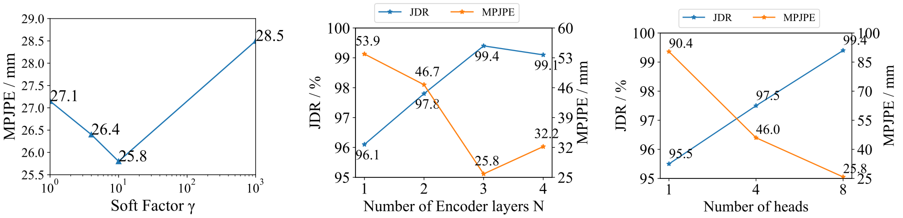
**Geometry Positional Encoding** We conduct ablation studies on the GPE to verify its significance. In detail, we consider 3 settings: 1) training without 3D geometry positional encoding, 2) applying a learnable 3D positional encoding, *i.e.*, directly learn $\mathbf{E}_{\mathbf{G}_i}$ from scratch 3) training the 3D GPE without epipolar field constraints $L_{\text{pos}}$. Table 3 presents the results. Without the 3D location information, the performance of 1) and 2) are even worse than the single view TransPose, we hypothesize that the 2D sine PE makes the transformer easy to attend the same pixel location of all views, and the learned 3D PE is easy to overfit the training examples. Without $L_{\text{pos}}$, the error will also increase. Thus the guide from the epipolar field is favorable, as it imposes correspondence for cross-view attention.

| Method | 2D Pose / JDR (%) ↑ | 3D Pose / MPJPE (mm) ↓ |
|---|---|---|
| TransFusion - without 3D positional encoding | 98.5 | 35.9 |
| TransFusion - learnable 3D positional encoding | 96.0 | 57.3 |
| TransFusion - GPE without $L_{\text{pos}}$ | 99.3 | 26.8 |
| TransFusion | **99.4** | **25.8** |

Table 3: Ablation studies on different types of 3D positional encoding

**Soft Factor $\gamma$** We also try different values of the soft vector $\gamma$, results are shown in Figure 7(a). With a small $\gamma$, the epipolar field assign all locations with relative high probabilities, the performance are slightly worse (1.3 mm drop). While with a huge $\gamma = 1000$, the epipolar field reduces to the hard epipolar line, and the performance drops 2.7 mm. Thus, we verify the effectiveness of our epipolar field compared with hard-coded epipolar line.

**Transformer architecture** We study how performance scales with the size of the transformer. As in Figure 7(b), with the number of layers $N$ increasing, the performance improves significantly, as the learning ability of transformer is more powerful with more parameters. But when $N > 3$, it tends to saturate or degenerate. We hypothesize that the transformer is easy to overfit when the size is too huge. Meanwhile, as in Figure 7(c), with the number of heads increases, the performance also improves gradually, as more heads can help attend different features [46]. In summary, our choice with $N = 3$ and 8 heas are reasonable.



(a) Soft Factor (b) Number of encoder layers (c) Number of heads

Figure 7: Ablation studies on the soft factor $\gamma$ of the epipolar field, the number of transformer encoder layers $N$, and the number of transformer heads.

| Method | 2D Pose / JDR (%) ↑ | 3D Pose / MPJPE (mm) ↓ |
|---|---|---|
| Single view - Simple Baseline [53] | 94.5 | 39.6 |
| Epipolar Transformer [19] | 94.9 | 34.2 |
| TransFusion | **96.0** | **31.6** |

Table 4: 2D and 3D pose estimation accuracy comparison on Ski-Pose.

### 4.4 Results on Ski-Pose Dataset

We further apply TransFusion on the Ski-Pose dataset to verify its generalization ability. Results are presented in Table 4. In the settings with six cameras, the Crossview Fusion is too huge (537M) to train on the 2080Ti GPU. Similar to Human 3.6M, TransFusion still outperform or achieve comparable performance with other fusion methods, while it is much lightweight. Thus, our method is also effective in outdoor multi-view settings.

## 5 Conclusion

In this paper, we apply the transformer to the multi-view 3D human pose estimation for the first time. Inspired by multi-modal transformers, we propose the TransFusion network, a lightweight architecture to integrate cues from both self views and reference views. Furthermore, we propose the epipolar field, and apply it to the 3D positional encoding to encode correspondence between two views explicitly. Experimental results shows that our method outperform previous fusion methods but with a more light weighted network. In the future we plan to apply our TransFusion to regress the 3D locations with multi-view inputs in an end-to-end way to further improve 3D predictions.

# References

[1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[3] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[4] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021.

[5] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Hui Tang, Yufan Xue, Xiaohui Xie, Yen-Yu Lin, and Wei Fan. Generating realistic training images based on tonality-alignment generative adversarial networks for hand pose estimation. *arXiv preprint arXiv:1811.09916*, 2018.

[6] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, Wei Fan, and Xiaohui Xie. Dggan: Depth-image guided generative adversarial networks for disentangling rgb and depth images in 3d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 411–419, 2020.

[7] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 836–845, 2021.

[8] Yifei Chen, Haoyu Ma, Deying Kong, Xiangyi Yan, Jianbao Wu, Wei Fan, and Xiaohui Xie. Nonparametric structure regularization machine for 2d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 381–390, 2020.

[9] Yifei Chen, Haoyu Ma, Jiangyuan Wang, Jianbao Wu, Xian Wu, and Xiaohui Xie. Pd-net: Quantitative motor function evaluation for parkinson's disease via automated hand gesture analysis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2683–2691, 2021.

[10] 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.

[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. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Benedikt Fasel, Jörg Spörri, Matthias Gilgien, Geo Boffi, Julien Chardonnens, Erich Müller, and Kamiar Aminian. Three-dimensional body and centre of mass kinematics in alpine ski racing using differential gnss and inertial sensors. *Remote Sensing*, 8(8): 671, 2016.

[13] Benedikt Fasel, Jörg Spörri, Julien Chardonnens, Josef Kröll, Erich Müller, and Kamiar Aminian. Joint inertial sensor orientation drift reduction for highly dynamic movements. *IEEE journal of biomedical and health informatics*, 22(1):77–86, 2017.

[14] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[15] Matthias Gilgien, Jörg Spörri, Julien Chardonnens, Josef Kröll, and Erich Müller. Determination of external forces in alpine skiing using a differential global navigation satellite system. *Sensors*, 13(8):9821–9835, 2013.

[16] Matthias Gilgien, Jörg Spörri, Philippe Limpach, Alain Geiger, and Erich Müller. The effect of different global navigation satellite system methods on positioning accuracy in elite alpine skiing. *Sensors*, 14(10):18433–18453, 2014.

[17] Matthias Gilgien, Jörg Spörri, Julien Chardonnens, Josef Kröll, Philippe Limpach, and Erich Müller. Determination of the centre of mass kinematics in alpine skiing using differential global navigation satellite systems. *Journal of sports sciences*, 33(9):960–969, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7779–7788, 2020.

[20] 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, jul 2014.

[21] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019.

[22] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

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

[24] Deying Kong, Yifei Chen, Haoyu Ma, Xiangyi Yan, and Xiaohui Xie. Adaptive graphical model network for 2d handpose estimation. *arXiv preprint arXiv:1909.08205*, 2019.

[25] Deying Kong, Haoyu Ma, Yifei Chen, and Xiaohui Xie. Rotation-invariant mixed graphical model network for 2d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1546–1555, 2020.

[26] Deying Kong, Haoyu Ma, and Xiaohui Xie. Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. *arXiv preprint arXiv:2009.12473*, 2020.

[27] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11336–11344, 2020.

[28] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. *arXiv preprint arXiv:2104.06976*, 2021.

[29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. *arXiv preprint arXiv:2012.09760*, 2020.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[31] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021.

[32] 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.

[33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[34] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4342–4351, 2019.

[35] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8437–8446, 2018.

[36] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[37] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[38] Jörg Spörri. Reasearch dedicated to sports injury prevention-the'sequence of prevention'on the example of alpine ski racing. *Habilitation with Venia Docendi in Biomechanics*, 1(2):7, 2016.

[39] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[40] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[41] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[42] Hao Tang, Xingwei Liu, Kun Han, Xiaohui Xie, Xuming Chen, Huang Qian, Yong Liu, Shanlin Sun, and Narisu Bai. Spatial context-aware self-attention model for multi-organ segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 939–949, 2021.

[43] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.

[44] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27:1799–1807, 2014.

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[47] 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.

[48] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020.

[49] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. *arXiv preprint arXiv:1905.07718*, 2019.

[50] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.

[51] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless C Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[52] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[53] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[54] Rongchang Xie, Chunyu Wang, and Yizhou Wang. Metafuse: A pre-trained fusion model for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13686–13695, 2020.

[55] Xiangyi Yan, Hao Tang, Shanlin Sun, Haoyu Ma, Deying Kong, and Xiaohui Xie. After-unet: Axial fusion transformer unet for medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[56] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020.

[57] Chenyu You, Nuo Chen, and Yuexian Zou. MRD-Net: Multi-Modal Residual Knowledge Distillation for Spoken Question Answering. In *IJCAI*, 2021.

[58] Chenyu You, Nuo Chen, and Yuexian Zou. Self-supervised contrastive cross-modality representation learning for spoken question answering. *arXiv preprint arXiv:2109.03381*, 2021.

[59] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019.

[60] Zhe Zhang, Chunyu Wang, Weichao Qiu, Wenhu Qin, and Wenjun Zeng. Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, 129(3):703–718, 2021.

[61] Yunhan Zhao, Shu Kong, and Charless Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15759–15768, 2021.

[62] 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.

[63] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[64] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[65] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain. *arXiv preprint arXiv:2103.16110*, 2021.

[66] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 813–822, 2019.