Multiple View Geometry Transformers for 3D Human Pose Estimation

Ziwei Liao[1,*]   Jialiang Zhu[2,*]   Chunyu Wang[3,†]   Han Hu[3]   Steven L. Waslander[1]

[1] University of Toronto   [2] Southeast University   [3] Microsoft Research Asia

ziwei.liao@mail.utoronto.ca, steven.waslander@robotics.utias.utoronto.ca

jialiangzhu@seu.edu.cn

chnuwa@microsoft.com, hanhu@microsoft.com

Abstract

In this work, we aim to improve the 3D reasoning ability of Transformers in multi-view 3D human pose estimation. Recent works have focused on end-to-end learning-based transformer designs, which struggle to resolve geometric information accurately, particularly during occlusion. Instead, we propose a novel hybrid model, MVGFormer, which has a series of geometric and appearance modules organized in an iterative manner. The geometry modules are learning-free and handle all viewpoint-dependent 3D tasks geometrically which notably improves the model's generalization ability. The appearance modules are learnable and are dedicated to estimating 2D poses from image signals end-to-end which enables them to achieve accurate estimates even when occlusion occurs, leading to a model that is both accurate and generalizable to new cameras and geometries. We evaluate our approach for both in-domain and out-of-domain settings, where our model consistently outperforms state-of-the-art methods, and especially does so by a significant margin in the out-of-domain setting. We will release the code and models: https://github.com/XunshanMan/MVGFormer.

1. Introduction

Estimating 3D human pose from multiple cameras is an important task with diverse applications. Recently, state-of-the-art accuracy on benchmark datasets [15, 17] has been advanced significantly by the application of large neural networks [29, 31, 36], due to their capabilities to learn complex priors required to handle severe occlusion. However, our study shows that the learnable prior is a double-edged sword that may harm its generalization ability, especially in the multiple-view setting where the camera views may un-
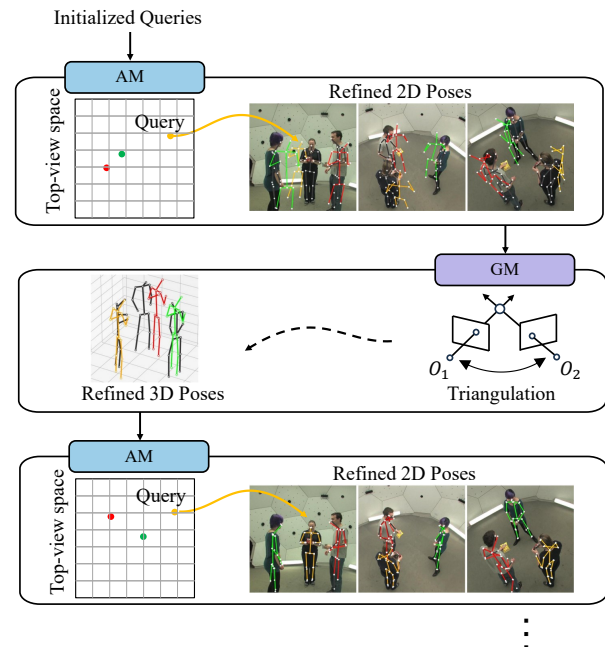
---

* Equal contribution
† Correspondance author

Figure 1. Overview of our MVGFormer which directly introduces multiple view geometry reasoning in Transformers. It is fully-differentiable and end-to-end trainable. The two types of modules (AM & GM) iteratively refine the 2D and 3D poses, respectively.

dergo large viewpoint variations in an environment unseen during training. This observation inspires us to consider the following question in our work:

What sub-tasks should be addressed geometrically in a learning-free style rather than by neural networks?

The core challenge of multiple view 3D human pose estimation is to establish correspondence across views and reason about the 3D world from 2D measurements. Some methods [4, 8, 9, 27, 37] address the task from a primarily geometric direction. They first estimate 2D poses in each view, group them into clusters, and then estimate a 3D pose for each cluster by triangulation. They generalize well to unseen settings because of explicit geometry modeling. However, they struggle to handle severe occlusion due to

the lack of effective priors leading to correspondence failures, when compared to learning-based neural networks.

Recent works [16, 29, 31, 35, 36] have emerged that rely on deep networks such as Transformers to map implicit image features to 3D poses bypassing explicit geometry modeling. Some methods such as [36] even map camera parameters to latent features for end-to-end training. The learned priors embedded in the networks enable them to achieve accurate estimates even when occlusion occurs. However, as will be shown in our experiments, learned approaches tend to obtain poor results on new scenes, especially when the testing camera configurations are not seen in training.

In this paper, we present a novel hybrid model called MVGFormer that combines Multiple View Geometry reasoning with Transformers. As shown in Fig. 1 and Fig. 2 (a), it takes as input a set of initialized queries that encode 3D poses, and iteratively refines them using two modules, the Appearance Module (AM) and Geometry Module (GM). In AM, we project a query onto the images to get a set of sampling points, and estimate a more accurate 2D pose from the sampled features (see Fig. 2 (b)). With the refined 2D poses in all views, GM predicts a refined 3D pose in a learning-free style through triangulation, which in turn is used to update the query (see Fig. 2 (c)). Thus, in the next round, AM can sample features at more accurate positions, leading to a refined 2D pose.

MVGFormer has four advantages. First, compared with a pure learning-based method that directly regresses 3D poses and lacks the geometric understanding of the 2D-3D relationship, we explicitly introduce the 3D geometry module, simplifying and constraining the learning task. This enhances the coordination between the appearance and geometry modules across multiple decoder layers. Second, decomposing the 2D and 3D tasks reduces the complexity of the problem. Third, a query, which represents a single 3D pose, geometrically guides the system to relate, fuse, and improve the 2D estimates in each view. Refining queries in a coarse-to-fine manner implicitly solves the challenging cross-view person grouping stage, by addressing the problem from a top-down perspective, making it robust to low-level errors from occlusion. Furthermore, during the end-to-end training, the geometric module does not suffer from quantization errors faced by voxel-based methods [31, 38].

We evaluate *MVGFormer* on three benchmark datasets in two settings: the commonly used in-domain setting and the less studied out-of-domain setting, where the method is evaluated on a dataset with viewpoints not seen during training. The latter is used to show the generalization ability of different models. We observe that our model is able to achieve accurate 2D estimates even when severe occlusion occurs (see Fig. 5 for examples) due to the effective multi-view regularization in GM during iterative refinement. This

lays the foundation for triangulation to obtain accurate 3D poses. Our method outperforms the state-of-the-art methods consistently on the three datasets. Further, it gets notably better results in the out-of-domain setting.

2. Related Work

Geometry modeling. Triangulation is commonly used in multi-view pose estimation [1, 7–9, 16, 27, 32, 37]. It takes as input the pre-detected 2D poses and computes a 3D pose by multi-view geometry [12]. Some work assumes only a single person in the scene [1, 7, 16]. When multiple persons exist, data associations across views need to be resolved. One group of works [8, 9, 27] follows a two-stage design, first solving associations with a hand-crafted affinity matrix, and then estimating 3D poses with triangulation. Though generalized to any cameras, they are vulnerable to inaccurate 2D pose estimates. A further set of methods require learning-based modules for data associations [32, 37], pose refinements [32], or depth regression [20], which face generalization challenges since the learning process depends on the training cameras. Most recently, Bartol et al. [1] investigate the generalization performance of triangulation, but only consider a single person in the scene. Video input has also been exploited for 3d pose estimation [9, 29, 37, 38] relying on temporal information to improve pose tracking over time. In this work, we focus on single-frame results, leaving the tracking problem as future work.

Volumetric Representation is another way to introduce geometric information into neural networks by projecting 2D image features into 3D [25, 26]. VoxelPose [31] and its variants [6, 29, 35, 38] are more robust to occlusion by fusing multi-view image features in a voxelized 3D representation. However, the high computational complexity and memory requirements of the dense 3D convolutions lead to a discretization tradeoff that frequently results in large quantization errors of 30mm [31] to fit on available hardware. VoxelPose is also susceptible to increased errors when applied to novel cameras or datasets, due to the learned dependence of visual patterns that arise in the voxels for specific camera angles observed during training. Performing extensive augmentations such as voxel rotations [16] is needed to help alleviate the problem, but the challenge persists.

Transformers for 3D Detections. DETR [5] is a Transformer framework for 2D image tasks and has been extended to 3D object detection from point clouds [11, 14, 24]. Some works [19, 23, 34, 39] construct volumetric BEV representations from multi-view images for 3D object detection in autonomous driving scenarios. However, they are not suitable for 3D human pose estimation because of the large quantization errors. PETR [21, 22] encodes the position information of 3D coordinates as features into DETR for multi-view 3D object detection. However, it ignores the
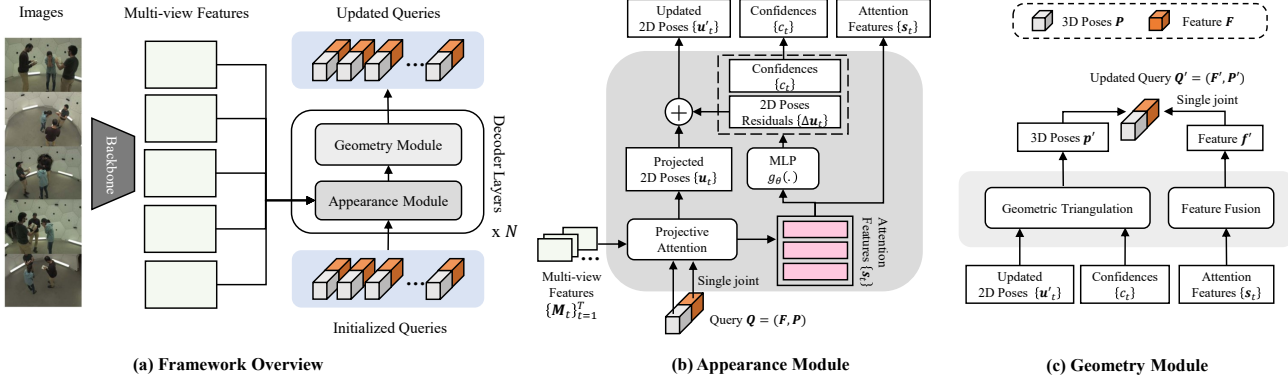
Figure 2. (a) MVGFormer estimates 3D human poses from multiple cameras with N decoder layers. (b) The appearance module updates 2D poses in each view mainly based on image features. (c) The geometry module explicitly estimates 3D poses based on the previously estimated 2D poses and the camera parameters. The two modules iteratively refine the initialized queries in a coarse-to-fine manner.

geometry constraints and requires a large amount of data to train the network to accurately learn the 2D-to-3D relationship. Thus, the approach results in overfitting to the training camera configurations, which limits its generalization performance. Most closely related to our work, MvP [36] extends DETR for multi-view 3D human pose estimation. Similar to PETR [21, 22], it introduces a RayConv operation to integrate the camera parameters into the image features. Although it achieves good in-domain performance when cameras are the same for training and testing, it does not generalize well to different camera arrangements. In our experiments, the accuracy almost decreases to zero when applying it to different camera arrangements or datasets. Taking our insight from these stark results on out-of-domain performance for existing learned methods such as MvP, we present a principled approach to leverage DETR in multi-view 3D tasks by enabling explicit multi-view geometry modeling within a course-to-fine refinement pipeline.

3. Methodology

A schematic illustration is provided in Figure 2. The input is a set of images $\{\mathbf{I}_t\}_{t=1}^{T}$ collected from T cameras. The images are passed to a backbone network to extract the multi-view feature maps $\{\mathbf{M}_t\}_{t=1}^{T}$. We initialize K compositional queries $\{\mathbf{Q}_k\}_{k=1}^{K}$ (Section 3.1), which go through a series of decoder layers to be iteratively refined to estimate the 3D poses and features (Section 3.2) for each person in the scene. We present the training objectives in Section 3.3.

3.1. Compositional Query

We use a compositional query to represent an individual person in the scene. Each query $\mathbf{Q}_k$ consists of an appearance term $\mathbf{F}_k \in \mathbb{R}^{J \times L}$ and a geometry term $\mathbf{P}_k \in \mathbb{R}^{J \times 3}$ where J and L denote the number of joints and the feature dimension, respectively. Note that each body joint has a separate feature vector in both terms. The two terms form

the compositional query: $\mathbf{Q}_k = (\mathbf{F}_k, \mathbf{P}_k)$.

Appearance Term We adopt a hierarchical query embedding scheme proposed in [36] to reduce the number of learnable parameters. In particular, we learn K instance embeddings $\{\mathbf{h}_k\}_{k=1}^{K} \subset \mathbb{R}^L$ and J joint embeddings $\{\mathbf{g}_j\}_{j=1}^{J} \subset \mathbb{R}^L$. The appearance term of joint j of the kth query is obtained by $\mathbf{f}_k^j = \mathbf{h}_k + \mathbf{g}_j$. The appearance terms of all joints are denoted as: $\mathbf{F}_k = [\mathbf{f}_k^1, \mathbf{f}_k^2, ..., \mathbf{f}_k^J]^\mathsf{T}$. The embeddings are randomly initialized from the standard normal distribution during training.

Geometry Term The geometry term $\mathbf{p}_k^j \in \mathbb{R}^3$ of joint j of the kth query directly stores its 3D position. We leverage a simple initialization method to ensure the best generalization. We place a T-pose at K uniformly sampled human centers on the ground plane and use the resulting 3D positions of the body joint to initialize $\mathbf{p}_k^j$, as in Figure 3 and Figure 4. The geometry terms can also be more accurately initialized by the 3D pose predictions obtained by other methods, like VoxelPose of which we will show examples in experiments. The geometry terms of all joints of the kth query are denoted as: $\mathbf{P}_k = [\mathbf{p}_k^1, \mathbf{p}_k^2, ..., \mathbf{p}_k^J]^\mathsf{T}$.

3.2. Transformer Decoders

As shown in Figure 2 (a), the model has multiple transformer decoder layers, with each having two successive modules that focus on the 2D and 3D tasks, respectively. We formally introduce the two modules in the following subsections. For the sake of clarity, we slightly abuse the notation and use $\mathbf{f}$ and $\mathbf{p}$ to represent the appearance and geometry terms of a single joint, respectively, and omit the subscripts k and j.

Appearance Module (AM) For each query, AM is responsible for refining the corresponding 2D poses in all views, respectively. Figure 2 (b) shows the architecture. For each query, the projective attention module extracts the attention features in each view for refining its 2D position. In par-
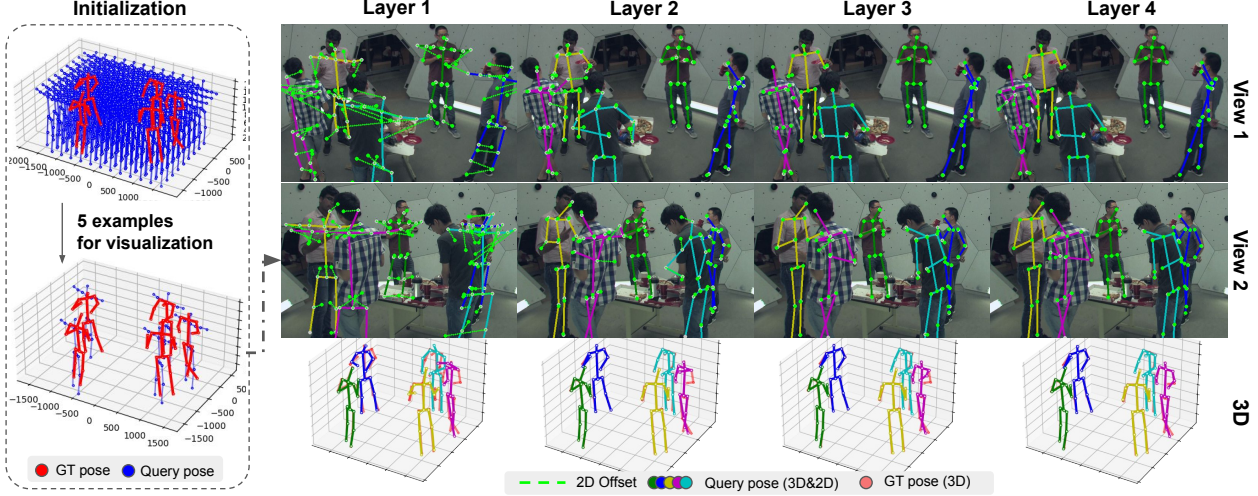
Figure 3. Iterative query refinement through decoder layers. (1) Initialization. We uniformly sample N T-poses on the ground to initialize queries. We visualize the details of 5 queries after filtering for clarity. (2) For each query, each joint is projected into 2D images to aggregate features with attention, and is updated with a learnt offset (green dashed line). Then, triangulation is used to estimate a 3D pose based on updated 2D poses in each view. The process is repeated in the four layers. Note that GT poses are NOT used during inference and are for visualization only. We recommend viewing electrically in color.

ticular, we project $\mathbf{p}$ to view t and obtain the 2D position $\mathbf{u}_t$: $\mathbf{u}_t = \mathbf{\Pi}_t[\mathbf{p}; 1]$, where $\mathbf{\Pi}_t \in \mathbb{R}^{3 \times 4}$ is the known projection matrix of camera t and $[\mathbf{p}; 1]$ is the homogeneous form. Then, following the deformable attention [41], we extract an attention feature vector $\mathbf{s}_t$ from the feature map $\mathbf{M}_t$ by predicting a set of sampling locations around $\mathbf{u}_t$. Then we learn an MLP $g_\theta(.)$ to predict a 2D residual (offset) $\Delta\mathbf{u}_t$ and a confidence score c_t from $\mathbf{s}_t$:

$$[\Delta\mathbf{u}_t, c_t] = g_\theta(\mathbf{s}_t), \qquad (1)$$

where c_t denotes the estimation confidence, used by the geometry module to reduce the impact of inaccurate 2D poses on the triangulation. We supervise c_t indirectly by minimizing the errors of the estimated 3D poses since the whole process is differentiable. We supervise $\Delta\mathbf{u}_t$ using the projected groundtruth 2D poses. We get the refined 2D position by adding the projected position and the residual:

$$\mathbf{u}'_t = \mathbf{u}_t + \Delta\mathbf{u}_t \qquad (2)$$

Besides estimating the residuals, we also use the attention feature $\mathbf{s}_t$ to update the query feature vector $\mathbf{f}$ as will be stated in the subsequent section. The updated query feature can better reflect the characteristics of the corresponding people in the images.

Geometry Module (GM) This module aims to estimate a more accurate 3D pose from the refined 2D poses. Figure 2 (c) shows the architecture. With the 2D positions in T views $\{\mathbf{u}'_t\}_{t=1}^T$, and the scores $\{c_t\}_{t=1}^T$ predicted in the previous module, we compute a new 3D position $\mathbf{p}'$ by differentiable algebraic triangulation [16]:

$$\mathbf{p}' = \text{Triangulate}(\{\mathbf{u}'_t\}_{t=1}^T, \{c_t\}_{t=1}^T, \{\mathbf{\Pi}_t\}_{t=1}^T), \qquad (3)$$

The scores are used to reduce the impact of inaccurate 2D poses. This module can be regarded as a denoising function because triangulation, which ensembles the multi-view 2D poses, can obtain a relatively accurate 3D pose even when the 2D poses in a small number of views are inaccurate. Thus, projecting the 3D pose back to each view usually gives a more accurate 2D pose, from which the appearance module can attract more accurate and relevant features.

Since the number of queries is larger than the actual number of people, we train an MLP-based classifier $f_\beta(.)$ to predict a score for each query based on the appearance term to remove the "empty" ones. It is set to be one for the positive queries and zero for the negative ones during training. We compute an overall score for each pose by averaging the scores of all joints. In both training and testing, we filter out the queries whose scores are smaller than ϵ layer by layer, as shown in Figure 4. This helps spend computation only on the relevant regions in the space.

Finally, we update each query for the next decoder layer. The geometry term is directly updated by the estimated 3D joint position $\mathbf{p}'$. The appearance term is updated by fusing the attention features in all views $\{\mathbf{s_t}\}_{t=1}^T$, which contains more accurate information describing the target person. We first compute a mean vector from $\{\mathbf{s_t}\}_{t=1}^T$. Then, we use an MLP $f_\alpha(.)$ to estimate a feature residual and add it to the origin appearance term $\mathbf{f}$. Finally, we feed it to another MLP $f_\gamma(.)$ to obtain $\mathbf{f}'$:

$$\mathbf{f}' = f_\gamma(\mathbf{f} + f_\alpha(\text{Mean}(\{\mathbf{s_t}\}_{t=1}^T))) \qquad (4)$$

It is worth noting that fusing multi-view features makes the approach more robust to occlusion in individual views,
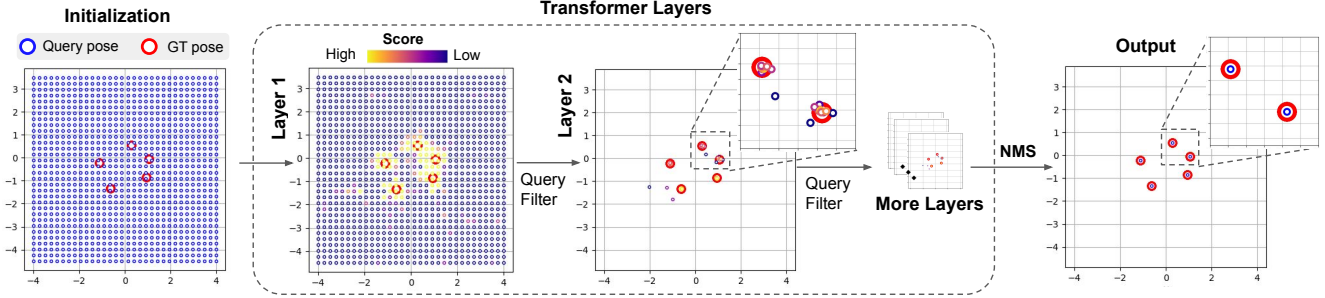
Figure 4. An example of the query distributions on the CMU Panoptic dataset with 5 persons. We show the center points of each pose from the top view. We uniformly sample $N = 1024 = 64 \times 64$ points on the ground and then initialize a T-pose for each. For each decoder layer, we filter queries with scores below a threshold. Finally, we use NMS to keep only one query with the highest score from a cluster of queries attached to the same person. GT poses are not used during inference and are for visualization only.

which improves the performance of our method.

3.3. Training Objectives

The whole model is end-to-end differentiable. We use an anchor-based matching strategy [40] to assign the queries to the ground truth poses. In particular, the initialized geometry terms of all queries $\{\mathbf{P}_k^{(0)}\}_{k=1}^K$ are regarded as the anchors. The ground-truth poses with Z persons are denoted as $\{\mathbf{H}_z\}_{z=1}^Z \subset \mathbb{R}^{J \times 3}$. For each $\mathbf{H}_z$, we assign it to W nearest queries $\{\mathbf{P}_{z(w)}\}_{w=1}^W$ based on their geometry terms where $z(w)$ is the index of the wth matched query. The unassigned queries are the negative anchors. The assignment is fixed during training. Our learning objective has two terms. The first is a pose loss supervising the 2D and 3D poses estimated from the positive queries:

$$
L_{pose}(\{\mathbf{P}_k\}, \mathbf{H}_z) = \sum_{w=1}^W (\underbrace{L_1(\mathbf{P}_{z(w)}, \mathbf{H}_z)}_{\text{3D pose loss}}
$$
$$
+ \sum_{t=1}^T \underbrace{L_1(\hat{\mathbf{U}}_{z(w),t}, \mathbf{U}_{z,t}))}_{\text{2D pose loss}} \quad (5)
$$

where $\hat{\mathbf{U}}_{z(w),t}$ is the predicted 2D pose of the $z(w)^{\text{th}}$ query in the tth view and $\mathbf{U}_{z,t}$ is the corresponding groundtruth 2D pose. The second is a cross-entropy loss for the classifier f_β applied to both positive and negative queries. We enforce the above losses at every decoder layer.

4. Experiments

4.1. Benchmarks and Implementation Details

Datasets, baselines, and metrics. We evaluate our method in two settings: in-domain and out-of-domain on three common benchmarks, CMU Panoptic [17], Shelf [2], and Campus [2]. We compare our method to three state-of-the-art baselines: two end-to-end learning methods: a

Transformer-based MvP [36], and a volumetric-based VoxelPose [31]; and a geometric method using triangulation Dong et al. [8]. We use their public official codes. Besides, we also list the published results of other works in the in-domain setting. Following [31, 36], we use metrics of Average Precision (AP), Recall, and Mean Per Joint Position Error (MPJPE) for the CMU Panoptic dataset, and Percentage of Correct Parts (PCP) for the Campus and Shelf datasets. Following the common practice, we use the camera poses provided by the datasets for a fair comparison of all the models. We discuss practical methods to estimate the camera poses in Appendix A5.

Implementation details. Following previous work [31, 36], we use ResNet-50 as the backbone for extracting image features. It is pre-trained for 2D pose estimation [33] on the COCO dataset. Thereafter, the backbone is fixed in the subsequent training steps. The rest modules are jointly trained for 40 epochs with a learning rate of $4e^{-4}$. We use four decoder layers with independent parameters for each. We filter out the queries whose scores are smaller than $\epsilon = 0.1$ at each layer. Finally, we use NMS to remove the redundant queries to obtain the final set of estimates, as shown in Figure 4. For the sampling space of the 1024 initialized queries, we use the given studio space for the CMU Panoptic dataset and use the maximum range of all the persons in the training set for the Campus and Shelf dataset. We present more architectural details in Appendix A1.

4.2. Generalization Performance

We present three types of generalization experiments with increasingly larger domain gaps including cross-cameras, cross-arrangements, and cross-datasets. We follow Bartol et al. [1] to consider five camera arrangements with different camera numbers and camera poses on the CMU Panoptic dataset, and denote them as CMU0, CMU1, CMU2, CMU3, and CMU4, respectively. We show more details in Appendix A3. We train all the models on the camera arrangement of CMU0 with 5 cameras. For in-domain ex-

Generalization	Change camera number									
Inference Datasets	CMU0 (3)		CMU0 (4)		CMU0 (6)		CMU0 (7)		Average	
Metrics	AP25	mAP	AP25	mAP	AP25	mAP	AP25	mAP	AP25	mAP
Dong et al. [8]	0.0	37.7	0.0	46.6	0.1	41.8	0.2	40.1	13.3	41.6
VoxelPose [31]	13.7	73.0	65.4	93.1	84.5	96.8	81.5	96.3	69.1	89.8
MvP [36]	12.3	57.1	59.8	84.1	0.0	0.3	0.0	0.2	21.6	35.4
Ours	**44.6**	**83.4**	**83.6**	**96.0**	**94.7**	**98.5**	**95.1**	**98.6**	**83.3**	**94.1**

Table 1. Generalization experiments on **changing camera number**. All models are trained on CMU0 with 5 cameras, and tested on *Inference Datasets*. CMU0 (K) means there are a total of K cameras after removing or adding cameras.

Generalization	Change camera arrangements									
Inference Datasets	CMU1		CMU2		CMU3		CMU4		Average	
Metrics	AP25	mAP	AP25	mAP	AP25	mAP	AP25	mAP	AP25	mAP
Dong et al. [8]	0.1	39.9	0.0	29.2	0.0	29.2	0.0	32.8	0.0	32.8
VoxelPose [31]	28.8	86.9	26.9	69.6	21.0	69.6	40.9	87.9	29.4	78.5
MvP [36]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ours	**86.8**	**96.1**	**67.9**	**90.3**	**52.5**	**78.8**	**91.5**	**97.2**	**74.7**	**90.6**

Table 2. Generalization experiments on **changing camera arrangements**. All models are trained on CMU0, and tested on *Inference Datasets*. The datasets CMU1-4 have different numbers and poses of cameras compared with CMU0.

periments, we also test on CMU0. For generalization experiments, the models are tested on other arrangements and datasets without finetuning.

Cross-cameras We remove or add a few cameras to CMU0. The results are in Table 1. "CMU0(K)" means the test camera number is K. When removing cameras, all methods witness accuracy loss partly due to increased occlusion (see Figure 3 for some examples). However, the accuracies of VoxelPose and MvP drop more significantly than ours due to their weak generalization ability.

When adding new cameras, we observe that ours is the only method that can benefit from the unseen cameras in training. For VoxelPose, the more unseen cameras we add, the worse the result becomes. The accuracy of MvP even decreases to zero because the rays of the new cameras were not seen during training, making them hard to be correctly handled by the network. MvP tends to only "memorize" the geometry for trained cameras but fails to generalize to unseen new cameras. Though the pure geometric method Dong et al. [8] in theory has no generalization problem, it fails to accurately reconstruct poses on challenging CMU Panoptic dataset. Note that only qualitative results on CMU are given in the original paper. Ours achieves an obvious improvement over the baselines.

Cross-arrangements Then we evaluate in a more challenging setting by completely changing the camera arrangements, *e.g.* having different camera numbers and poses from those used in training. The results are shown in Table 2.

Our method obtains notably better results than the baselines. VoxelPose sacrifices accuracy a lot for new cameras. MvP fails to work on unseen cameras. The pure geometric method Dong et al. [8] fails to accurately reconstruct the poses due to severe occlusion. Ours can be directly ap-

plied to CMU1 and CMU4 with high accuracy. The accuracy on the CMU2 and CMU3 datasets is relatively low due to the small number of cameras and the missing of the other side of the view, as shown in Appendix A3. In summary, we achieve much higher accuracy on all arrangements compared with all the baselines. Due to the difficulty of including all possible camera arrangements in training, the practical values of the other learning-based baselines are limited.

Cross-datasets We train the models on CMU0 and test them on the Shelf and Campus datasets. The results are in Table 3. When testing without finetuning, ours shows much better generalization performance than the baselines, where MvP still fails to generalize to new cameras. It is worth noting that ours achieves a PCP of 87.9% on the Shelf. Though much better than the learning baselines, the accuracy on Campus is relatively low due to the significant appearance domain gap and the larger outdoor space, which we will discuss more in Appendix A7. Finetuning further improves the performance a lot. Ours achieves a state-of-the-art performance than both the learning and geometric baselines. Note that finetuning needs to collect training data in a new setting which is costly in time and resources, and sometimes not available. Ours with generalization capacity shows a potential trend to train once and deploy anywhere. The pure geometric method Dong et al. [8] gets reasonable results on the Shelf and Campus dataset. But as discussed in Table 1 and 2, the performance drops when occlusion occurs.

4.3. In-domain Performance

We evaluate the in-domain performance of the models on three benchmarks, CMU Panoptic [17], Shelf [2], and Campus [2] in Table 4. We use the same train and test splits following [31, 36]. The same camera arrangements are used

Generalization		Change scenes and datasets							
Inference Datasets		Shelf				Campus			
Training Setting	Methods	A1	A2	A3	Average	A1	A2	A3	Average
N/A	Dong et al. [8]	98.8	94.1	**97.8**	96.9	97.6	93.3	98.0	96.3
w/o Finetune	VoxelPose [31]	77.9	63.2	68.1	69.8	16.1	12.4	5.1	11.2
	MvP [36]	5.7	16.0	4.6	8.7	0.0	0.0	0.0	0.0
	Ours	**89.9**	**85.4**	**88.6**	**87.9**	**40.2**	**61.0**	**73.1**	**58.1**
w/ Finetune	VoxelPose [31]	99.3	94.1	97.6	97.0	97.6	93.8	**98.8**	96.7
	MvP [36]	99.3	95.1	**97.8**	97.4	98.2	94.1	97.4	96.6
	Ours	**99.5**	**96.8**	**97.8**	**98.0**	**99.2**	**94.4**	96.5	96.7

Table 3. Generalization experiments on **changing scenes and datasets**. All learning-based models are trained on CMU0 from the CMU Panoptic dataset, and tested on *Inference Datasets* with or without finetunning. Metric: PCP.

Methods	CMU Panoptic		Shelf	Campus
	AP25 ↑	MPJPE ↓	PCP ↑	PCP ↑
Belagiannis et al. [3]	-	-	77.5	84.5
Ershadi et al. [10]	-	-	88.0	90.6
VoxelTrack [38]	-	-	97.1	96.7
Graph3D [32]	-	15.8	97.7	-
Lin et al. [20]	92.1	16.8	97.9	97.0
Faster VoxelPose [35]	85.2	18.3	97.6	96.2
Chen et al. [6]	83.8	17.6	97.3	96.3
Perez-Yus et al. [27]	-	-	96.5	96.7
Dong et al. [8]	-	-	96.9	96.3
VoxelPose [31]	84.0	17.7	97.0	96.7
MvP [36]	92.3	15.8	97.4	96.6
Ours	**92.3**	16.0	**98.0**	96.7

Table 4. In-domain experiments with the same training and testing settings. We consider the published image-based methods for a fair comparison. "-" indicates missing data in the published paper.

for training and testing in this experiment. It demonstrates that the generalization ability of our method is achieved without sacrificing its fitting ability. Instead, it achieves comparable even better results than the state-of-the-art. We show qualitative results in Figure 5 and Appendix A2.

The results of the baselines are obtained from the papers (our re-implementation gets similar results). Our method outperforms VoxelPose [31] by a margin in terms of AP25, which suggests that the end-to-end framework can achieve precise estimation results by largely decreasing the quantization error. Besides, it performs comparably to or better than MvP [36], which is also an end-to-end approach, on all metrics. However, as discussed in Section 4.2, MvP has the risk of overfitting to the training cameras, making the results less meaningful. Since the Shelf and Campus datasets are small, we finetuned the model trained on CMU0 following the previous work [31, 36]. Our method achieves comparable results to the state-of-the-art methods on both datasets. We have compared the generalization performance of [31, 36] and show a much better performance in Sec 4.2.

4.4. Ablation Study

Number of Queries This parameter is mainly determined by the size of the motion capture space. The guideline is to make sure each GT pose has a sufficiently large overlap with at least one query so that the model can extract appropriate features for estimating the 2D poses in the first layer. As shown in Table 5a, when the number of queries increases from 128 to 1024, the performance keeps growing and reaches the maximum performance at 1024. When we further increase the number of queries to 1280, the performance begins to slightly drop because it poses a challenge for the training. We didn't try the query numbers larger than 1280 because of the GPU memory limitation in training.

Number of Decoder Layers The 3D poses are iteratively refined through the decoder layers as shown in Figure 3. The improvement is significant in the first two layers and becomes marginal from the third layer. The observation agrees with the numerical results shown in Table 5b. When increasing the number of layers, the accuracy continuously improves and reaches the best with 4 layers, which validates that the proposed iterative refinement is effective. When we keep adding more layers, the performance saturates. It is worth noting that if we only use one layer, AP25 drops significantly to 6%, showing that multiple decoder layers and iterative refinement are essential to obtain precise estimates.

Query Assignment During training, we select W queries whose initialized geometry terms are nearest to the ground truth poses. Then, we enforce pose losses to train those queries to output the matched groundtruth poses. As shown in Table 5c, when setting W to 5, the performance of AP25 improves from 86.0% to 92.3%. Compared with the one-to-one matching strategy, e.g., Hungarian loss in DETR [5], the multiple-to-one matching strategy can augment the training data and help the network converge better. The performance begins to drop when selecting more than 5 queries since some queries far away are also matched which is difficult to learn. In particular, we tried the Hungarian loss but it did not converge in our experiments.

Appearance Term We evaluate three strategies for appear-

Queries	AP25	AP100	MPJPE
128	77.9	93.4	26.2
256	86.8	97.4	20.1
512	90.7	98.4	18.0
1024	**92.3**	**99.3**	**16.0**
1280	92.0	99.6	16.0

(a) Query number

Layers	AP25	AP100	MPJPE
1	5.2	96.4	47.2
2	79.2	97.9	19.9
3	88.6	98.9	17.6
4	**92.3**	**99.3**	**16.0**
5	91.0	99.4	16.3

(b) Layer number

KNN	AP25	AP100	MPJPE
1	86.0	98.5	19.1
2	87.0	99.0	17.9
3	90.5	99.2	17.4
4	91.4	99.2	17.0
5	**92.3**	**99.3**	**16.0**
6	90.8	99.1	16.7

(c) Query assignment

Feature	AP25	AP100	MPJPE
Joint-level	90.9	99.2	17.0
Person-level	**92.3**	**99.3**	**16.0**
Human-level	91.0	99.4	16.1

(d) Appearance term

Methods	AP25	AP100	MPJPE
MLP	**92.3**	**99.3**	**16.0**
Attention	90.4	99.1	16.8
Mean	36.2	54.0	71.8

(e) Feature fusion

NMS	AP25	AP100	MPJPE
w/o	20.0	20.8	16.2
w/	**92.3**	**99.3**	**16.0**

(f) NMS

Table 5. Ablation studies. The settings with the best AP25 have been highlighted in bold.
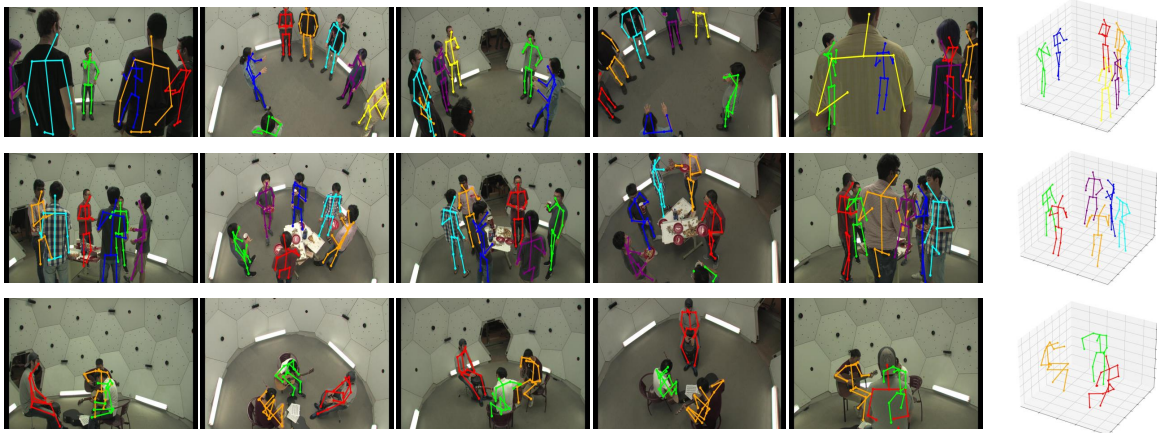

Figure 5. Sample estimation results on the CMU Panoptic dataset.

ance term embedding: the joint-level strategy where each joint in each query has independent parameters; the human-level embedding where all queries share the same parameters; and our adopted hierarchical embedding strategy. The results are shown in Table 5d. The human-level and person-level strategies achieve better results than the joint-level. Compared with a human-level feature, person-level is more flexible and each query can learn a feature related to its initialized positions and achieves the best performance.

Feature Fusion In each decoder layer, we need to update the queries by fusing the features from multiple views. We ablated several fusion methods in Table 5e. We notice that MLP achieves the best results, even a little better than attention. We think that attention has better capacity but also is harder to learn compared with MLP.

Non-maximum Suppression As we use the KNN-based instead of the one-to-one matching strategy during training, multiple queries may have large confidence scores for a single GT pose. Non-maximum suppression is important to remove the redundant estimates, as shown in Figure 4. Without NMS, we can see that AP25 and AP100 drop significantly in Table 5f. When using NMS to filter out those redundant estimates, the accuracy is boosted to 92.3%.

Inference Speed We use a single V100 GPU for inference. Ours needs 0.21s for each inference, which is faster than VoxelPose, which needs 0.29s. Ours lies inside a 19% range with MvP which needs 0.17s. A more efficient query sampling strategy remains in future work.

We provide more ablation studies in the Appendix.

5. Conclusion

We propose a hybrid method for multi-view 3D human pose estimation, combining a classical geometry module with a learned appearance module in a coarse-to-fine iterative network. Differing from the previous methods, we explicitly decouple 2D appearance and 3D geometry in the Transformer architecture to achieve strong generalization performance and yet maintain the robustness benefits of end-to-end learning. The 2D module directly estimates and refines the 2D poses from the images, while the 3D module recovers the 3D poses using triangulation and is immune to quantization errors. This approach results in an end-to-end trained model that generalizes well across different cameras and datasets. Our framework is generalizable and can be applied to other keypoint estimation tasks such as shape, hand, and face estimation.

References

[1] Kristijan Bartol, David Bojanić, Tomislav Petković, and Tomislav Pribanić. Generalizable human pose triangulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11028–11037, 2022. 2, 5, 1

[2] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1669–1676, 2014. 5, 6

[3] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures revisited: Multiple human pose estimation. *TPAMI*, 38(10):1929–1942, 2015. 7

[4] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-person 3d pose estimation and tracking in sports. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2, 7

[6] Yuxing Chen, Renshu Gu, Ouhan Huang, and Gangyong Jia. Vtp: Volumetric transformer for multi-view multi-person 3d pose estimation. *arXiv preprint arXiv:2205.12602*, 2022. 2, 7

[7] Zhuo Chen, Xu Zhao, and Xiaoyue Wan. Structural triangulation: A closed-form solution to constrained 3d human pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 695–711. Springer, 2022. 2, 4

[8] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7792–7801, 2019. 1, 2, 5, 6, 7

[9] Junting Dong, Qi Fang, Wen Jiang, Yurou Yang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation and tracking from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6981–6992, 2021. 1, 2, 4

[10] Sara Ershadi-Nasab, Erfan Noury, Shohreh Kasaei, and Esmaeil Sanaei. Multiple human 3d pose estimation from multiview images. *Multimedia Tools and Applications*, 77(12): 15573–15601, 2018. 7

[11] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[13] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J Black. Populating 3d scenes by learning human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14708–14718, 2021. 4

[14] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8417–8427, 2022. 2

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *T-PAMI*, 36(7):1325–1339, 2014. 1

[16] 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. 2, 4

[17] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 5, 6

[18] Sang-Eun Lee, Keisuke Shibata, Soma Nonaka, Shohei Nobuhara, and Ko Nishino. Extrinsic camera calibration from a moving person. *IEEE Robotics and Automation Letters*, 7(4):10344–10351, 2022. 3

[19] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022. 2

[20] Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11886–11895, 2021. 2, 7

[21] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 531–548. Springer, 2022. 2, 3

[22] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 2, 3

[23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 2

[24] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceed-*

ings of the IEEE/CVF International Conference on Computer Vision, pages 2906–2917, 2021. 2

[25] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, pages 1263–1272. IEEE, 2017. 2

[26] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, pages 7307–7316, 2018. 2

[27] Alejandro Perez-Yus and Antonio Agudo. Matching and recovering 3d people from multiple views. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3622–3631, 2022. 1, 2, 7

[28] Jens Puwein, Luca Ballan, Remo Ziegler, and Marc Pollefeys. Joint camera pose estimation and 3d human pose estimation in a multi-camera setup. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, pages 473–487. Springer, 2015. 3

[29] N Dinesh Reddy, Laurent Guigues, Leonid Pishchulin, Jayan Eledath, and Srinivasa G Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15190–15200, 2021. 1, 2, 4

[30] Joaquim Salvi, Xavier Armangué, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern recognition*, 35(7):1617–1635, 2002. 3

[31] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 197–212. Springer, 2020. 1, 2, 5, 6, 7, 3

[32] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11148–11157, 2021. 2, 7

[33] 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. 5

[34] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *arXiv preprint arXiv:2211.10439*, 2022. 2

[35] Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster voxelpose: Real-time 3d human pose estimation by orthographic projection. In *European Conference on Computer Vision*, pages 142–159. Springer, 2022. 2, 7

[36] Jianfeng Zhang, Yujun Cai, Shuicheng Yan, Jiashi Feng, et al. Direct multi-view multi-person 3d pose estimation. *Advances in Neural Information Processing Systems*, 34: 13153–13164, 2021. 1, 2, 3, 5, 6, 7

[37] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4d association graph for realtime multi-person

motion capture using multiple video cameras. In *CVPR*, pages 1324–1333, 2020. 1, 2, 4

[38] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenyu Liu, and Wenjun Zeng. Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2613–2626, 2022. 2, 7, 4

[39] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 2

[40] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 5

[41] 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. 4