

# TransCenter: Transformers with Dense Queries for Multiple-Object Tracking

Yihong Xu<sup>\*1</sup> Yutong Ban<sup>\*2</sup> Guillaume Delorme<sup>1</sup>  
Chuang Gan<sup>3</sup> Daniela Rus<sup>2</sup> Xavier Alameda-Pineda<sup>†</sup>

<sup>1</sup>Inria, LJK, Univ. Grenoble Alpes, France <sup>2</sup>Distributed Robotics Lab, CSAIL, MIT, USA

<sup>3</sup>MIT-IBM Watson AI Lab

<sup>1</sup>{firstname.lastname}@inria.fr <sup>2,3</sup>{yban, ganchuang, rus}@csail.mit.edu

## Abstract

Transformer networks have proven extremely powerful for a wide variety of tasks since they were introduced. Computer vision is not an exception, as the use of transformers has become very popular in the vision community in recent years. Despite this wave, multiple-object tracking (MOT) exhibits for now some sort of incompatibility with transformers. We argue that the standard representation – bounding boxes – is not adapted to learning transformers for MOT. Inspired by recent research, we propose TransCenter, the first transformer-based architecture for tracking the centers of multiple targets. Methodologically, we propose the use of dense queries in a double-decoder network, to be able to robustly infer the heatmap of targets' centers and associate them through time. TransCenter outperforms the current state-of-the-art in multiple-object tracking, both in MOT17 and MOT20. Our ablation study demonstrates the advantage in the proposed architecture compared to more naive alternatives. The code will be made publicly available.

## 1. Introduction

The task of tracking multiple objects, usually understood as the simultaneous inference of the position and identity of various persons in a visual scene recorded by one or more cameras, became a core problem in computer vision in the past years. Undoubtedly, the various multiple-object tracking (MOT) challenges and associated datasets [40, 10], helped foster research on this topic and provided a standard way to evaluate and monitor the performance of the methods proposed by many research teams worldwide.

In line with recent progress in computer vision using transformers [54] for tasks such as pedestrian detec-

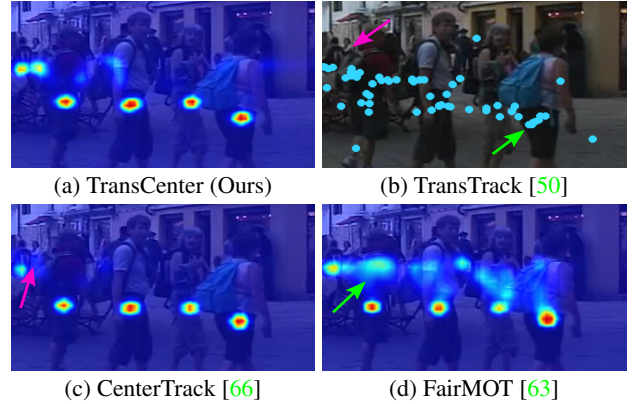


Figure 1: Results of state-of-the-art MOT methods: (a), (c) and (d) are center heatmaps of TransCenter (Ours), CenterTrack [66] and FairMOT [63] respectively, (b) shows the bounding boxes centers from the queries in TransTrack [50]. Previous transformer-based tracking methods [50, 39] use sparse queries, leading to miss detections (pink arrow), that are heavily overlapped, possibly leading to false detections (green arrow). Previous MOT center trackers [63, 66] suffer from the same problems because the centers are estimated independently of each other. TransCenter is designed to mitigate these two adverse effects by using dense (pixel-level) multi-scale queries to enable heatmap-based inference and exploiting the attention mechanisms to introduce co-dependency between center predictions.

tion [7, 67, 34], person re-identification [24] or image super resolution [60], we are interested in investigating the use of transformer-based architectures for multiple-object tracking, as recent evidence [50, 39] demonstrated the interest of exploring the use of such architectures for this task. However, we argue that the pedestrian representation used so far is not appropriate for learning transformer-based architectures for MOT. Indeed, TransTrack [50] and TrackFormer [39] use bounding boxes to represent pedestrians, which is very intuitive since bounding-box is a wide-spread

<sup>\*</sup>Equal contribution.

<sup>†</sup>Xavier Alameda-Pineda acknowledges funding from the ANR ML3RI project (ANR-19-CE33-0008-01) and the H2020 SPRING project (under GA #871245).

representation for MOT for instance in combination with probabilistic methods [45, 2] or deep convolutional architectures [3, 59, 44, 42, 55, 18, 62, 48]. One of the prominent drawbacks of using bounding boxes for tracking multiple objects manifests when dealing with very crowded scenes [10], where occlusions are very difficult to handle since ground-truth bounding boxes often overlap each other. This is problematic because these bounding boxes are used during training, not only to regress the position, width, and height of each person but also to discriminate the visual appearance associated to each track. In this context, overlapping bounding boxes mean training a visual appearance representation that combines the visual content of two or even more people [23, 22]. Certainly, jointly addressing the person tracking and segmentation tasks [39] can partially solve the occlusion problem. However, this requires to have extra annotations – segmentation masks – which are very tedious and costly to obtain. In addition, such annotations are not available in standard benchmark datasets [40, 10].

In this paper, we get inspiration from very recent research in MOT [66, 63] and choose to devise a transformer-based architecture that can be trained to track the center of each person, and name it TransCenter. Therefore, the main difference with respect to TransTrack [50] and TrackFormer [39], developed directly from object detection transformers [67] and [7] respectively, is that TransCenter is conceived to mitigate the occlusion problem inherent to bounding-box tracking without requiring extra ground-truth annotations such as segmentation masks. While this intuition is very straightforward, designing an efficient transformer-based architecture that implements this intuition is far from evident.

Indeed, the first challenge is to be able to infer dense representations (i.e. center heatmaps). To do so, we propose the use of dense (pixel-level) multi-scale queries. In addition to allowing heatmap-based MOT, the use of dense queries overcomes the limitations [7, 67] associated with querying the decoder with a small number of queries. Inspired by [50], TransCenter has two different decoders: one for person detection and another one for person tracking. Both decoders are given queries that depend on the current image, but they are extracted with different learnable layers. However, while the memory (i.e. the output of the transformer encoder) of the current frame is given to the detection decoder, the memory of the previous frame is given to the tracking decoder.

Overall, this paper has the following contributions:

- We propose the use of transformers for multiple-object center tracking and term this architecture TransCenter.
- To infer position heatmaps, we propose the use of dense multi-scale queries that are computed from the encoding of the current image using learnable layers.
- TransCenter sets a new state-of-the-art baseline among

online MOT tracking methods in MOT17 [40] (+10.1% multiple-object tracking accuracy, MOTA) as well as MOT20 [10] (+5% MOTA), leading both MOT competitions. Moreover, to our knowledge, TransCenter sets the first transformer-based state-of-the-art baseline in MOT20<sup>1</sup>, thanks to its ability to track in crowded scenes.

## 2. Related Works

### 2.1. Multiple-Object Tracking

In MOT literature, initial works [2, 45, 1] focus on how to find the optimal associations between detections and tracklets through probabilistic models while [41] first formulates the problem as an end-to-end learning task with recurrent neural networks. Moreover, [47] models the dynamics of objects by a recurrent network and further combines the dynamics with an interaction and an appearance branch. [59] proposes a framework to directly use the standard evaluation measures MOTA and MOTP as loss functions to backpropagate the errors for an end-to-end tracking system. [3] employs object detection methods for MOT by modeling the problem as a regression task. A person re-identification network [53, 3] can be added at the second stage to boost the performance. However, it is still not optimal to treat the person re-identification as a secondary task. [63] further proposes a framework that treats the person detection and re-identification task equally.

Moreover, traditional graphs are also used to model the positions of objects as nodes and the temporal connection of the objects as edges [25, 53, 51, 29, 52]. The performance of those methods is further boosted by the recent rise of Graph Neural Networks (GNNs): hand-designed graphs are replaced by learnable GNNs [56, 57, 55, 43, 6] to model the complex interaction of the objects.

In most of the methods above, bounding boxes are used as object representation for the network. However, it is not a satisfying solution because it creates ambiguity when objects occlude each other, or noisy background information is included. CenterTrack [66] and FairMOT [63] represent objects as heatmaps then reasons about all the objects jointly and associate heatmaps across frames.

### 2.2. Transformers in Vision

Transformer is first proposed by [54] for machine translation, and has shown its ability to handle long-term complex dependencies between sequences by using multi-head attention mechanism. With its great success in natural language processing, works in computer vision start to investigate transformers for various tasks, such as image recognition [14], person re-identification [24], realistic image gen-

<sup>1</sup>TrackFormer [39] is tested on MOT20S, which are sequences from MOT17 containing far less crowded scenes than MOT20.

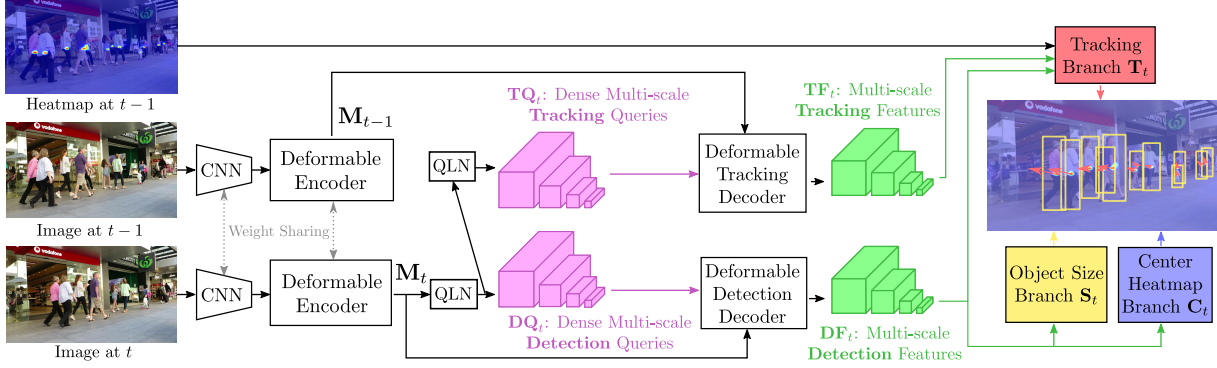


Figure 2: Overview of TransCenter. Images at  $t$  and  $t - 1$  are fed to a CNN backbone to produce multi-scale features, then processed by a deformable encoder to produce memory  $M_t$  and  $M_{t-1}$  respectively.  $M_t$  is used to compute dense multi-scale detection and tracking queries ( $DQ_t$  and  $TQ_t$ ) through two query learning networks (QLN).  $DQ_t$  and  $TQ_t$  are fed to the detection and tracking deformable decoders respectively, together with  $M_t$  and  $M_{t-1}$ . The outputs are multi-scale detection and tracking features ( $DF_t$  and  $TF_t$ ) and are used to estimate the center heatmap and object sizes. Both multi-scale features, together with the center heatmap at  $t - 1$  are used to estimate the displacement vector for each center.

eration [27], super resolution [60] and audio-visual learning [17, 16].

Object detection with Transformer (DETR) [7] can be seen as an exploration and correlation task. It is an encoder-decoder structure where the encoder extracts the image information and the decoder finds the best correlation between the object query and the encoded image features with an attention module. However, the attention calculation suffers from heavy computational and memory complexities w.r.t the input size: the feature map extracted from a ResNet [21] backbone is used as the input to alleviate the problem. Deformable DETR [67] tackles the issue by proposing a deformable attention inspired by [9], drastically speeding up convergence ( $10\times$ ) and reducing the complexity. This allows to capture finer details by using multi-scale features, yielding better detection performance.

Following the success in detection using transformers, two concurrent works directly apply transformers on MOT based on DETR framework. First, Trackformer [39] builds directly from DETR [7] and is trained to propagate the queries through time. Second, Transtrack [50] extends [67] to MOT by adding a decoder that processes the features at  $t - 1$  to refine previous detection positions. Importantly, both methods stay in the detection framework and use it for tracking, a strategy that have proven successful in previous works [59, 3]. However, recent literature [66, 63] also suggests that bounding boxes may not be the best representation for MOT, and this paper investigates the use of transformers for center tracking, thus introducing TransCenter.

### 3. TransCenter for Multiple Object Tracking

We are motivated to investigate the use of transformers for multiple-object tracking. As described in the introduc-

tion, previous works in this direction attempted to learn to infer bounding boxes. We question this choice, and explore the use of an alternative representation very popular in the recent past: center heatmaps. However, differently from bounding boxes, heatmaps are *dense* rather than *sparse* representations. Consequently, while [50, 39] used sparse object queries, we introduce the use of dense multi-scale queries for transformers in computer vision. Indeed, up to our knowledge, we are the first to propose the use of a dense query feature map that scales with the input image size. To give a figure, in our experiments the decoders are queried with roughly 14k queries. One downside of using dense queries is the associated memory consumption. To mitigate this undesirable effect, we propose to use deformable decoders, inspired by deformable convolutions.

More precisely, we cast the multiple-object tracking problem into two separate subtasks: the detection of objects at time  $t$ , and the association with objects detected at  $t - 1$ . Different from previous studies following the same rationale [3, 59], TransCenter addresses these two tasks in parallel, by using a fully deformable dual decoder architecture. The output of the detection decoder is used to estimate the object center and size, while it is combined with the output of the tracking decoder to estimate the displacement of the object w.r.t. the previous image. An important consequence of combining center heatmaps with the use of a dual decoder architecture is that the object association through time depends not only on geometry features (e.g. IOU) but also on the visual features from the decoder.

#### 3.1. TransCenter in a Nutshell

The overall architecture of TransCenter can be seen in Figure 2. The RGB images at time  $t$  and  $t - 1$  are fed to a CNN backbone to produce multi-scale features and cap-

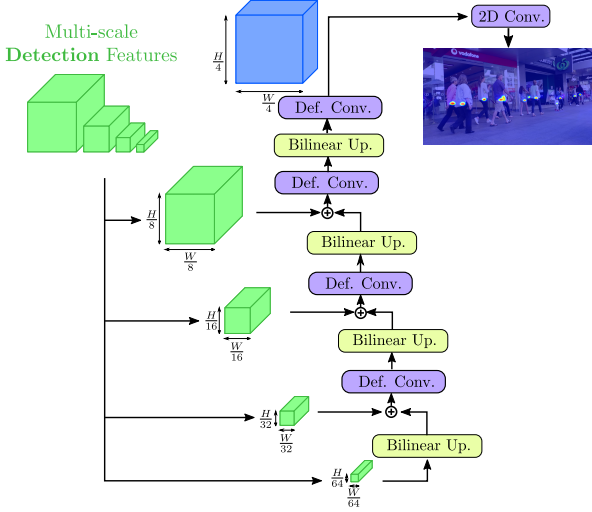


Figure 3: Overview of the center heatmap branch. The multi-scale detection features are upsampled and merged via a series of deformable convolutions, into the output center heatmap. A similar strategy is followed for the object size and the tracking branches.

ture finer details in the image as done in [67] and then to a deformable self-attention encoder, thus obtaining multi-scale memory feature maps associated to the two images,  $\mathbf{M}_t$  and  $\mathbf{M}_{t-1}$  respectively. Then,  $\mathbf{M}_t$  is given to a query learning network (QLN), which are fully connected layers operating pixel-wise, that outputs a feature map of dense multi-scale detection queries,  $\mathbf{DQ}_t$ . These go through another QLN to produce a feature map of dense multi-scale tracking queries,  $\mathbf{TQ}_t$ . A fully deformable dual decoder architecture is then used to process them: the deformable detection decoder compares the detection queries  $\mathbf{DQ}_t$  to the memory  $\mathbf{M}_t$  to output multi-scale detection features  $\mathbf{DF}_t$ , and the deformable tracking decoder does the same with the tracking queries  $\mathbf{TQ}_t$  and the memory  $\mathbf{M}_{t-1}$  to output multi-scale tracking features  $\mathbf{TF}_t$ . The detection multi-scale features are used to estimate the bounding box size  $\mathbf{S}_t$  and the center heatmap  $\mathbf{C}_t$ . Together with the tracking features and the center heatmap,  $\mathbf{C}_{t-1}$ , the detection features are also used to estimate the tracking displacement  $\mathbf{T}_t$ .

In the following we first explain the design of the dense multi-scale queries, then the architecture of the fully deformable dual decoder, the three main branches – center heatmap, object size, and tracking – and finally the training losses.

### 3.2. Dense Multi-scale Queries

Traditional transformer architectures output as many elements as queries fed to the decoder, and more importantly, these outputs correspond to the entities sought (e.g. pedestrian bounding boxes). When inferring center heatmaps, the probability of having a person’s center at a given pixel be-

comes one of these sought entities, thus requiring the transformer decoder to be fed with dense queries. Such queries are obtained from the multi-scale encoder’s memory, via a first query learning network (QLN), which is a feed-forward network operating pixel-wise, obtaining  $\mathbf{DQ}_t$ . We use two different queries for the dual decoder: a second QLN processes  $\mathbf{DQ}_t$  to obtain  $\mathbf{TQ}_t$ . They will be fed to the fully deformable dual decoder, see Sec. 3.3.

The fact that the dense query feature map resolution is proportional to the resolution of the input image has two prominent advantages. First, the queries can be multi-scale and exploit the multi-resolution structure of the encoder, allowing for very small targets to be captured by those queries. Second, dense queries also make the network more flexible since it is able to adapt to arbitrary image size. More generally, the use of QLN avoids the problem of manually sizing the queries and selecting beforehand the number of maximum detection, as it was done in previous transformer architectures (for computer vision).

### 3.3. Fully Deformable Dual Decoder

To successfully find object trajectories, a MOT method should not only detect the objects but also associate them across frames. To do so, TransCenter proposes to use a fully deformable dual decoder. More precisely, two fully deformable decoders deal in parallel with the two subtasks: detection and tracking. While the detection decoder correlates  $\mathbf{DQ}_t$  and  $\mathbf{M}_t$  with the attention modules to detect objects in the image  $I_t$ , the tracking decoder correlates  $\mathbf{TQ}_t$  and  $\mathbf{M}_{t-1}$  to associate the detected objects to their position in the previous image  $I_{t-1}$ . Specifically, the detection decoder searches for objects in multi-scale  $\mathbf{M}_t$  with the attention correlated to the multi-scale  $\mathbf{DQ}_t$  and then outputs the multi-scale detection features  $\mathbf{DF}_t$ , used to find the object centers and box sizes. Differently, the deformable tracking decoder finds the objects in  $\mathbf{M}_{t-1}$  and associates them with the objects at  $t$ . To do this, the multi-head deformable attention in the tracking decoder performs a temporal cross-correlation between the multi-scale  $\mathbf{TQ}_t$  and  $\mathbf{M}_{t-1}$  and outputs the multi-scale tracking features  $\mathbf{TF}_t$ , containing the temporal information that is used in the tracking branch to estimate the displacements from time  $t$  back to  $t - 1$ .

Both the detection and tracking decoders input a dense query feature map so as to output dense information as well. However, the use of the multi-head attention modules used in traditional transformers [54] in TransCenter implies a memory and complexity growth that is quadratic with the input image size  $O(H^2W^2)$ . Of course this is undesirable and would limit the scalability and usability of the method, especially when processing multi-scale features. Naturally, we resort to deformable multi-head attention, thus leading to a fully deformable dual decoder architecture.

### 3.4. The Center, the Size and the Tracking Branches

The output of the two fully deformable decoders are two sets of multi-scale features, referred to as the detection  $\mathbf{DF}_t$  and tracking features  $\mathbf{TF}_t$ . More precisely, these multi-scale features contain four feature maps at different resolutions, namely  $1/64, 1/32, 1/16$  and  $1/8$  of the input image resolution. For the center heatmap and the object size branches, the feature maps at different resolutions are combined using deformable convolutions and bilinear interpolation, following the architecture shown in Figure 3, into a feature maps of  $1/4$  of the input resolution, and finally into  $\mathbf{C}_t \in [0, 1]^{H/4 \times W/4}$  and  $\mathbf{S}_t \in \mathbb{R}^{H/4 \times W/4 \times 2}$  (the two channels of  $\mathbf{S}_t$  encode the width and the height). Regarding the tracking branch, the two multi-scale features follow the same up-scaling as in the two other branches (but with different parameters), obtaining two feature maps at resolution  $1/4$ . These two feature maps are concatenated to the previous center heatmap  $\mathbf{C}_{t-1}$  downscaled to the resolution of the feature maps. As in the other branches, a block of convolutional layers computes the final output, i.e. the displacement of the objects  $\mathbf{T}_t \in \mathbb{R}^{H/4 \times W/4 \times 2}$  where the two channels encode the horizontal and vertical displacements respectively.

### 3.5. Training TransCenter

Training TransCenter is achieved by jointly learning a classification task for the object center heatmap and a regression task for the object size and tracking displacements, covering the branches of TransCenter. For the sake of clarity, in this section we will drop the time index  $t$ .

**Center Focal Loss** In order to train the center branch, we need first to build the ground-truth heatmap response  $\mathbf{C}^* \in [0, 1]^{H/4 \times W/4}$ . As done in [66], we construct  $\mathbf{C}^*$  by considering the maximum response of a set of Gaussian kernels centered at each of the  $K > 0$  ground-truth object centers. More formally, for every pixel position  $(x, y)$  the ground-truth heatmap response is computed as:

$$\mathbf{C}_{xy}^* = \max_{k=1, \dots, K} G((x, y), (x_k, y_k); \sigma), \quad (1)$$

where  $(x_k, y_k)$  is the ground-truth object center, and  $G(\cdot, \cdot; \sigma)$  is the Gaussian kernel with spread  $\sigma$ . In our case,  $\sigma$  is proportional to the object's size, as described in [30]. Given the ground-truth  $\mathbf{C}^*$  and the inferred  $\mathbf{C}$  center heatmaps, the center focal loss,  $L_C$  is formulated as:

$$L_C = \frac{1}{K} \sum_{xy} \begin{cases} (1 - \mathbf{C}_{xy})^\alpha \log(\mathbf{C}_{xy}) & \mathbf{C}_{xy}^* = 1, \\ (1 - \mathbf{C}_{xy}^*)^\beta (\mathbf{C}_{xy})^\alpha \log(1 - \mathbf{C}_{xy}) & \text{otherwise.} \end{cases} \quad (2)$$

where the scaling factors are  $\alpha = 2$  and  $\beta = 4$ , see [63].

**Sparse Regression Loss** The values of  $\mathbf{S}$  and  $\mathbf{T}$  are supervised only on the locations where object centers are present,

i.e.  $\mathbf{C}_{xy}^* = 1$  using a  $L_1$  loss:

$$L_S = \frac{1}{K} \sum_{xy} \begin{cases} \|\mathbf{S}_{xy} - \mathbf{S}_{xy}^*\|_1 & \mathbf{C}_{xy}^* = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The formulation of  $L_T$  is analogous to  $L_S$  but using the tracking output and ground-truth, instead of the object size. To complete the sparsity of  $L_S, L_T$ , we add an extra  $L_1$  regression loss, denoted as  $L_R$  with the bounding boxes computed from  $\mathbf{S}_t$  and ground-truth centers. The impact of this additional loss is marginal as shown in Section 4.4.

In summary, the overall loss is formulated as the weighted sum of all the losses, the weights are chosen according to the numeric scale of each loss:

$$L = L_C + \lambda_S L_S + \lambda_T L_T + \lambda_R L_R \quad (4)$$

## 4. Experimental Evaluation

### 4.1. Implementation Details

**Inference with TransCenter** Once the method is trained, we detect objects by filtering the output center heatmap  $\mathbf{C}_t$ . Since the datasets are annotated with bounding boxes, we need to convert our estimates into this representation. In detail, we apply a threshold  $\tau = 0.5$  to the heatmap, thus producing a list of center positions  $\{\mathbf{c}_{t,k}\}_{k=1}^{K_t}$ . We extract the object size  $\mathbf{s}_{t,k}$  associated to each position  $\mathbf{c}_{t,k}$  in  $\mathbf{S}_t$ . The set of detections produced by TransCenter is directly  $\mathbf{D}_t = \{\mathbf{c}_{t,k}, \mathbf{s}_{t,k}\}_{k=1}^{K_t}$ . Once the detection step is performed, we can estimate the position of the object in the previous image extracting the estimated displacement  $\mathbf{t}_{t,k}$  from the tracking branch output  $\mathbf{T}_t$  and the center position  $\mathbf{c}_{t,k}$ . Indeed, we can construct a set of detections *tracked back to the previous image*  $\tilde{\mathbf{D}}_{t-1} = \{\mathbf{c}_{t,k} + \mathbf{t}_{t,k}, \mathbf{s}_{t,k}\}_{k=1}^{K_t}$ . Finally we use the Hungarian algorithm to match the detections at the previous time step  $\mathbf{D}_{t-1}$  with the tracked-back detection  $\tilde{\mathbf{D}}_{t-1}$  to associate the tracks through time. The birth and death processes are naturally integrated in TransCenter: Detections not associated to previous detections give birth to new tracks, while unmatched previous detections are put to sleep for at most  $T = 60$  frames before being discarded. New tracks are compared to sleeping tracks by means of an external re-identification network from [3] trained only on MOT17 [40], whose impact is ablated in the experiments.

**Network and Training Parameters** The input images are resized to  $640 \times 1088$ . Both the encoder and the decoder have six layers with hidden dimension  $h = 256$  with eight attention heads. The query learning networks consist of two fully connected layers with ReLU activation. Our CNN backbone is ResNet-50 [21]. TransCenter is trained with loss weights  $\lambda_S = 0.1$ ,  $\lambda_R = 0.5$  and  $\lambda_T = 1.0$  by the AdamW optimizer [37] with learning rate  $2e-5$  for the

Table 1: Results on MOT17 [40]. The left and right halves of the table correspond to public and private detections respectively. The cell background color encodes the amount of extra-training data: green for none, orange for one extra dataset, red for five extra datasets. Methods with \* are not associated to a publication. The best result within the same training conditions (background color) is underlined. The best result among published methods is in bold. Best seen in color.

Method	Public Detections									Private Detections								
	Data	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	Data	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓
TransCenter (Ours)	CH	<b>71.9</b>	<b>81.4</b>	<b>62.3</b>	<b>38.0</b>	<b>22.7</b>	<b>17,378</b>	<b>137,008</b>	4,046	CH	<b>73.2</b>	<b>81.1</b>	62.2	<b>40.8</b>	<b>18.5</b>	23,112	<b>123,738</b>	4,614
*TrackFormer [39]	CH	61.8		59.8	35.4	<u>21.1</u>	35,226	177,270	2,982									
*UnsupTrack [28]	PT	61.7	78.3	58.1	27.2	32.4	16,872	197,632	1,864									
MOTDT17 [8]	RE	50.9	76.6	52.7	17.5	35.7	24,069	250,768	<b>2,474</b>									
*TransTrack [50]																		
CenterTrack [66]	NO	61.5	78.9	59.6	26.4	31.9	14,076	200,672	2,583	CH	65.8	78.8	56.9	32.2	21.8	24,000	163,683	5,355
FUFET [48]	NO	62.0		59.5	27.8	31.5	15,114	19,6672	2,621	CH	67.8	78.4	<b>64.7</b>	34.6	24.6	<b>18,489</b>	160,332	<b>3,039</b>
MLT [62]										(5D1)	<b>76.2</b>	81.1	68.0	<b>51.1</b>	<b>13.6</b>	32,796	<b>98,475</b>	3,237
*CSTrack [33]										(5D1)	75.3	<b>81.7</b>	<b>75.5</b>	49.3	19.5	<b>27,879</b>	109,836	<b>1,719</b>
*FairMOT [63]										5D1	74.9	80.9	72.6	41.5	17.5	<b>23,847</b>	114,303	3,567
*GSDT [55]										5D1	73.7	81.3	72.3	43.2	17.3	<b>27,507</b>	117,477	3,303
GSM-Tracker [36]	NO	56.4	77.9	57.8	22.2	34.5	14,379	230,174	<b>1,485</b>	5D2	66.2	79.9	<b>68.7</b>	40.8	18.3	43,368	144,261	3,318
Tracker++ [3]	NO	56.3	78.8	55.1	21.1	35.3	<b>8,866</b>	235,449	1,987									
TrctrD17 [59]	NO	53.7	77.2	53.8	19.4	36.6	11,731	247,447	1,947									
Tracker [3]	NO	53.5	78.0	52.3	19.5	36.6	12,201	248,047	2,072									
*MAT [20]	NO	67.1	<b>80.8</b>	<b>69.2</b>	<b>38.9</b>	26.4	22,756	161,547	1,279									
ChainedTracker [44]										NO	66.6	78.2	57.4	32.2	24.2	22,284	160,491	5,529
TubeTK [42]										NO	63.0	78.3	58.6	31.2	<b>19.9</b>	27,060	177,483	<b>4,137</b>
TransCenter (Ours)	NO	<b>68.8</b>	<b>79.9</b>	<b>61.4</b>	<b>36.8</b>	<b>23.9</b>	22,860	<b>149,188</b>	4,102	NO	<b>70.0</b>	<b>79.6</b>	<b>62.1</b>	<b>38.9</b>	20.4	<b>28,119</b>	<b>1,36,722</b>	4,647

CNN backbone and 2e−4 for the rest of the network. The training lasts 50 epochs, applying learning rate decay of 1/10 at the 40th epoch. The entire network is pre-trained on the pedestrian class of COCO [35] and then fine-tuned on the respective MOT dataset [40, 10]. Overall, with 2 RTX Titan GPUs and batch size 2, it takes around 1h30 and 1h per epoch of MOT20 and MOT17 respectively. We also present the results fine-tuning with extra data, namely the CrowdHuman dataset [49]. See the results and discussion for details.

## 4.2. Protocol

**Datasets and Detections** We use the standard split of the MOT17 [40] and MOT20 [10] datasets and the evaluation is obtained by submitting the results to the MOTChallenge website. The MOT17 test set contains 2,355 trajectories distributed in 17,757 frames. MOT20 test set contains 1,501 trajectories within only 4,479 frames, which leads to a much more challenging setting. We evaluate TransCenter both under public and private detections. When using public detections, we limit the maximum number of birth candidates at each frame to be the number of public detections per frame, as in [66, 39]. The selected birth candidates are those closest to the public detections with IOU larger than 0. When using private detections, there are no constraints, and the detections depend only on the network capacity, the use of external detectors, and more importantly, the use of extra training data. For this reason, we regroup the results by the use of extra training datasets as detailed in the following.

**Extra Training Data** To fairly compare with the state-of-the-art methods, we clearly denote the extra data used

to train each method (including several pre-prints listed in the MOTChallenge leaderboard, which are marked with \* in our result tables):<sup>2</sup> CH for CrowdHuman [49], PT for PathTrack [38], RE for the combination of Market1501 [64], CUHK01 and CUHK03 [31] person re-identification datasets, 5D1 for the use 5 extra datasets (CrowdHuman [49], Caltech Pedestrian [13, 12], CityPersons [61], CUHK-SYS [58], and PRW [65]), 5D2 is the same as 5D1 replacing CrowdHuman by ETH [15], (5D1) uses the tracking/detection results of FairMOT [63] (trained with in 5D1 setting), and NO for using no extra dataset.

**Metrics** Standard MOT metrics such as MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision) [4] are used: MOTA is mostly used since it reflects the average tracking performance including the number of FPs (False positives, predicted bounding boxes not enclosing any object), FNs (False negatives, missing ground-truth objects) and IDS [32] (Identities of predicted trajectories switch through time). MOTP evaluates the quality of bounding boxes from successfully tracked objects. Moreover, we also evaluate on IDF1 [46] (the ratio of correctly identified detections over the average number of ground-truth objects and predicted tracks), MT (the ratio of ground-truth trajectories that are covered by a track hypothesis more than 80% of their life span), and ML (less than 20% of their life span).

<sup>2</sup>COCO [35] and ImageNet [11] are not considered as extra data according to the MOTchallenge [40, 10].

Table 2: Results on MOT20 [10]. The table is structured following the same principles as Table 1. Methods with \* are not associated to a publication. The best result within the same training conditions (background color) is underlined. The best result among published methods is in bold. Best seen in color.

Method	Public Detections										Private Detections									
	Data	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓		Data	MOTA ↑	MOTP ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDS ↓	
TransCenter (Ours)	CH	<b>58.6</b>	<b>79.8</b>	<b>46.7</b>	<b>35.5</b>	<b>18.7</b>	<b>33,691</b>	<b>175,841</b>	<b>4,850</b>		CH	<b>58.3</b>	<b>79.7</b>	<b>46.8</b>	<b>35.7</b>	<b>18.6</b>	<b>35,959</b>	<b>174,893</b>	<b>4,947</b>	
*UnsupTrack [28]	PT	53.6	<u>80.1</u>	<u>50.6</u>	30.3	25.0	6,439	231,298	2,178											
*GSDT [55]											5d2	<u>67.1</u>	<u>79.1</u>	67.5	53.1	13.2	31,913	135,409	3,131	
*CSTrack [33]											5d1	66.6	78.8	<u>68.6</u>	50.4	15.5	25,404	144,358	3,196	
*FairMOT [63]											5d1	61.8	78.6	67.3	<u>68.8</u>	<u>7.6</u>	103,440	<u>88,901</u>	5,243	
*GNNMatch [43]	NO	54.5	79.4	49.0	32.8	25.5	9,522	223,611	2,038											
Tractor++ [3]	NO	52.6	<b>79.9</b>	<b>52.7</b>	29.4	26.7	<b>6,930</b>	236,680	<b>1,648</b>											
SORT [5]	NO	42.7	78.5	45.1	16.7	26.2	27,521	264,694	4,470											
MLT [62]											NO	48.9	78.0	<b>54.6</b>	30.9	22.1	45,660	216,803	<b>2,187</b>	
TransCenter (Ours)	NO	<b>57.5</b>	79.4	47.1	<b>35.6</b>	<b>18.0</b>	40,443	<b>174,850</b>	4,840		NO	<b>57.1</b>	<b>79.4</b>	46.7	<b>35.7</b>	<b>18.0</b>	<b>42,871</b>	<b>173,911</b>	4,940	

### 4.3. Results and Discussion

**MOT17** Table 1 presents the results obtained on the MOT17 [40] dataset. The first global remark is that most state-of-the-art methods do not evaluate under both public and private detections, and under different extra-training data settings, while we do. Secondly, TransCenter systematically outperforms all other methods, in terms of MOTA, under similar training data conditions, both for public and private detections. Indeed, the increase of MOTA w.r.t. the best performing published method is of 21% (10.1% taking unpublished methods into account) and 6.8% for public detections under extra and no-extra training data, and of 5.4% and 3.4% for private detections. If we consider only published methods, the superiority of TransCenter is remarkable in most of the metrics. We can also observe that TransCenter trained with no extra-training data outperforms, not only the methods trained with no extra data but also the methods trained with one extra dataset (in terms of MOTA for both public and private detections). In the same line, TransCenter trained on CH performances better than two of the methods trained with five extra datasets. Overall, these results confirm our hypothesis that heatmaps representation combined with the proposed TransCenter architecture is a better option for MOT using transformers.

**MOT20** Table 2 reports the results obtained in MOT20. In public detections, TransCenter leads the competition both in extra (+5% MOTA) and no-extra (+3% MOTA) training data. Another remarkable achievement of TransCenter is the significant decrease of FP when compared to the existing methods (−50  $k$  and beyond). Very importantly, to the best of our knowledge, our study is the first to report the results on MOT20 of a transformer-based architecture, demonstrating the tracking capacity of TransCenter even in a densely crowded scenario. For the sake of completeness, we provide the results on MOT20 for private detections and set a new baseline for future research for methods trained under CH and no extra data.

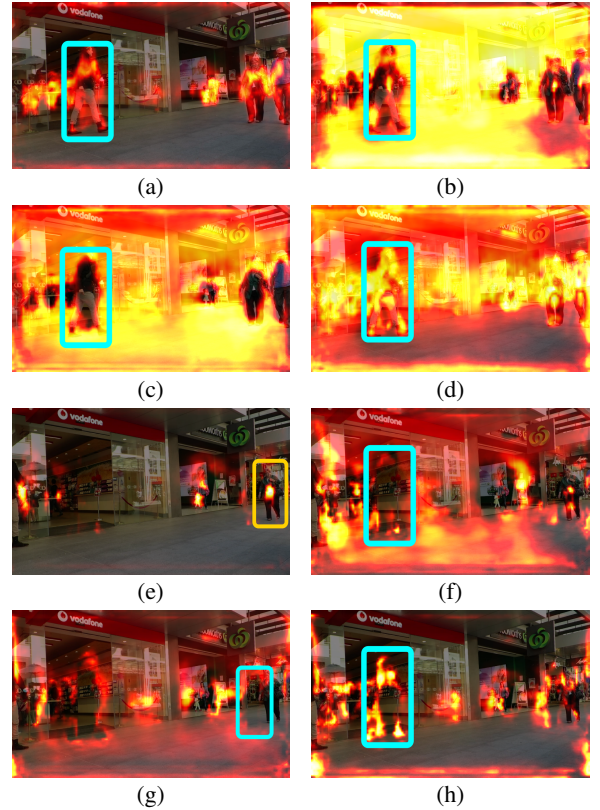


Figure 4: Visualization of the attention from the detection (a)-(d) decoder and the tracking (e)-(h) decoder in the current and previous image at  $t - 50$  (for better visualization), respectively. The brighter the higher the attention weights.

**Attention Visualization** We show in Fig. 4 the attention from different attention heads of both detection and tracking decoders. We can see that for the detection attention, different heads focus on different areas of  $I_t$ : (a) the people; (b), (c) the background; (d) both the background and the people. For the tracking attention, interestingly we observe that the object information at  $t$  does correlate to the previous image: in (f)-(h), the tracking decoder tries to look for objects at  $t - 1$  in the surrounding of the positions of the

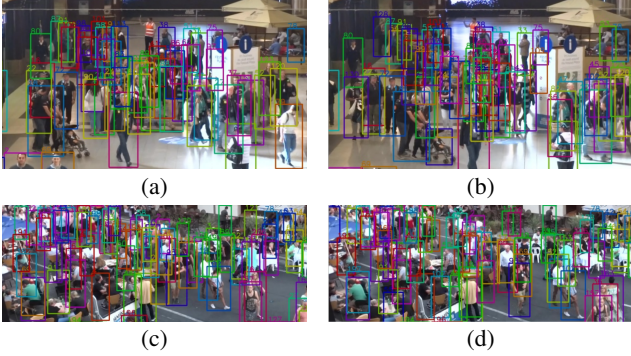


Figure 5: Tracking results visualization of TransCenter on MOT20 test set, in the Private Detection setting.

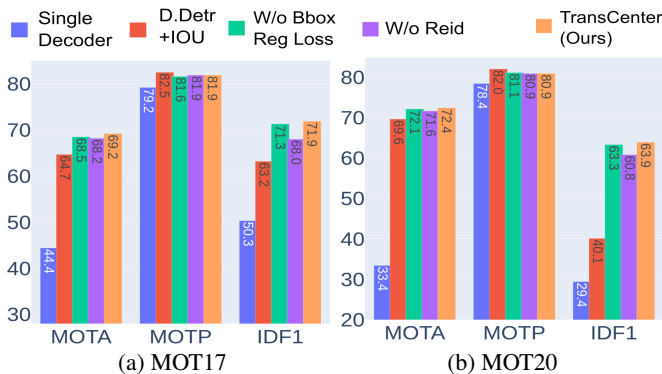


Figure 6: MOTA, MOTP and IDF1 ablation studies on MOT17 and MOT20.

objects at  $t$ . In addition, it also focuses in the objects in the previous image, as shown within the orange box in (e).

**Qualitative Results** We report in Fig. 5 qualitative results on the MOT20 test set, to assess the ability of TransCenter to detect and track targets in the context of crowded scenes and highly overlapping bounding boxes. Fig 5(a) and 5(b) are extracted from MOT20-07, Fig 5(c) and 5(d) MOT20-08. We observe that TransCenter manages to keep high recall, even in the context of drastic mutual-occlusions and reliably associate detections across time.

To summarize, TransCenter exhibits outstanding results on both MOT17 and MOT20 datasets for both public and private detections, and for both with or without extra training data, which indicates that multiple-object center tracking using transformers is a promising research direction.

#### 4.4. Ablation Study

In this section, we experimentally demonstrate different configurations of our TransCenter. For the ablation, we further divide the training sets into train-validation split, we take the first 50% of frames (2,664 and 4,468 frames for MOT17 and MOT20, respectively) as training data and test on the last 25% (1,332 and 2,234 frames for MOT17 and

Table 3: Ablation studies on MOT17 [40] and MOT20 [10].

Method	MOT17			MOT20		
	FP ↓	FN ↓	IDS ↓	FP ↓	FN ↓	IDS ↓
Single decoder	305	12,991	1,782	1,302	160,254	43,482
D.Detr+IOU	1,291	7,774	507	6,921	78,648	7,978
W/out $L_R$ Loss	1,279	7,090	184	5,210	79,103	1,589
W/out Reid	1,202	6,951	467	7,107	76,137	4,286
TransCenter (ours)	1,202	6,951	203	7,127	76,157	1,549

MOT20, respectively). The rest 25% frames in the middle of the sequences are thrown to prevent over-fitting.

**Single Decoder Is Not Enough** We study the possibility of using one single decoder and one set of dense multi-scale queries to perform tracking. Using a single decoder leads to very poor results, as shown in Figure 6 (Single Decoder). This is because the network switches its attention between image  $t$  and image  $t - 1$  during training and eventually fails to track objects correctly at  $t$  and  $t - 1$  (low MOTA). More details can be found in supplementary materials. Using a single decoder for sure brings the memory efficiency, which is not so crucial in TransCenter, thanks to the deformable modules [67]. The overall memory consumption is therefore affordable for a normal GPU setting (see details in Sec. 4.1).

**Lost Person Re-identification** We use an external Re-ID network to recover the identities which are temporally suspended by the tracker. The Re-ID network is the one in [3], pre-trained on MOT17 [40] training set. Similarly, a light-weight optical flow estimation network LiteFlowNet [26] pre-trained on Kitti [19] is used to recover the lost identities. This process helps us to reduce IDS, but the overall tracking performance does not come from these external networks since FP, FN is not improved by them. see Tab. 3, we even observe a performance drop of FP and FN since the external networks were not finetuned on MOT20.

**Beyond Detection** We also ablate the D.Detr [67]+IOU matching, which is to use bounding box object detection and handcrafted geometry IOU matching method to perform tracking. From Figure 6, we observe that bounding box object detector can better enclose correctly detected objects (i.e. higher MOTP). However, due to the fact that it lacks the prior information from the past, which leads to a higher IDS and FNs.

**Without  $L_R$**  We evaluate the impact of the additional bounding box regression loss  $L_R$  that completes the sparse object size loss, as discussed in Section 3.5. We observe a slight performance drop (-0.7% MOTA for MOT17 and -0.3% for MOT20), indicating that the two sparse regression losses and the dense center heatmap focal loss are sufficient to train TransCenter.

## 5. Conclusion

In this paper, we introduce TransCenter, a novel transformer-based architectures for multiple-object tracking. TransCenter proposed the use of dense multi-scale queries in combination with a fully deformable dual decoder, able to output dense representations for the objects' center, size and temporal displacement. The deformable decoder allows processing thousands of queries while keeping the overall memory usage within reasonable boundaries. Under the same training conditions, TransCenter outperforms all its competitors in MOT17 and MOT20, and even exhibits comparable performance to some methods trained with much more data.

## References

- [1] Nathanael L Baisa. Online multi-object visual tracking using a gm-phd filter with deep appearance learning. In *22th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2019. 2
- [2] Yutong Ban, Sileye Ba, Xavier Alameda-Pineda, and Radu Horaud. Tracking multiple persons based on a variational bayesian model. In *European Conference on Computer Vision (ECCV)*, pages 52–67. Springer, 2016. 2
- [3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *IEEE International Conference on Computer Vision (ICCV)*, pages 941–951, 2019. 2, 3, 5, 6, 7, 8
- [4] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 6
- [5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Uppcroft. Simple online and realtime tracking. In *IEEE international Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016. 7
- [6] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2
- [7] 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 (ECCV)*, pages 213–229. Springer, 2020. 1, 2, 3
- [8] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. 6
- [9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 3
- [10] Patrick Dendorfer, Hamid Rezaatofghi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003[cs]*, March 2020. 1, 2, 6, 7, 8, 12
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. 6
- [12] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009. 6
- [13] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 2012. 6
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, October, 2020. 2
- [15] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008. 6
- [16] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. *European Conference on Computer Vision (ECCV)*, pages 758–775. Springer, 2020. 3
- [17] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10478–10487, 2020. 3
- [18] Chuang Gan, Hang Zhao, Peihao Chen, David Cox, and Antonio Torralba. Self-supervised moving vehicle tracking with stereo sound. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7053–7062, 2019. 2
- [19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 8
- [20] Shoudong Han, Piao Huang, Hongwei Wang, En Yu, Donghaisheng Liu, Xiaofeng Pan, and Jun Zhao. Mat: Motion-aware multi-object tracking. *arXiv preprint arXiv:2009.04794*, September, 2020. 6
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016. 3, 5
- [22] Lingxiao He and Wu Liu. Guided saliency feature learning for person re-identification in crowded scenes. In *European Conference on Computer Vision (ECCV)*, pages 357–373. Springer, 2020. 2
- [23] Lingxiao He, Yinggang Wang, Wu Liu, He Zhao, Zhenan Sun, and Jiashi Feng. Foreground-aware pyramid reconstruction for alignment-free occluded person re-identification. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

- [24] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *arXiv preprint arXiv:2102.04378*, February, 2021. 1, 2
- [25] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning (ICML)*, pages 4364–4375. PMLR, 2020. 2
- [26] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Lite-flownet: A lightweight convolutional neural network for optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8981–8989, 2018. 8
- [27] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, February, 2021. 3
- [28] Shyamgopal Karthik, Ameya Prabhu, and Vineet Gandhi. Simple unsupervised multi-object tracking. *arXiv preprint arXiv:2006.02609*, June, 2020. 6, 7
- [29] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, July, 2016. 2
- [30] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision (ECCV)*, pages 734–750, Springer, 2018. 5
- [31] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deep-reid: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 152–159, 2014. 6
- [32] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2953–2960, 2009. 6
- [33] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv preprint arXiv:2010.12138*, October, 2020. 6, 7
- [34] Matthieu Lin, Chuming Li, Xingyuan Bu, Ming Sun, Chen Lin, Junjie Yan, Wanli Ouyang, and Zhidong Deng. Detr for pedestrian detection. *arXiv preprint arXiv:2012.06785*, December, 2020. 1
- [35] 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 (ECCV)*, pages 740–755. Springer, 2014. 6
- [36] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 530–536. International Joint Conferences on Artificial Intelligence Organization, July, 2020. Main track. 6
- [37] Ilya Loshchilov, and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, May, 2019. 5
- [38] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool. Pathtrack: Fast trajectory annotation with path supervision. In *IEEE International Conference on Computer Vision (ICCV)*, pages 290–299, October, 2017. 6
- [39] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, January, 2021. 1, 2, 3, 6
- [40] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831 [cs]*, March, 2016. 1, 2, 5, 6, 7, 8, 11, 12, 15, 16
- [41] Anton Milan, Seyed Hamid Reza Tofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI Conference on Artificial Intelligence*, volume 31, 2017. 2
- [42] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 6
- [43] Ioannis Papakis, Abhijit Sarkar, and Anuj Karpatne. Gcnmatch: Graph convolutional neural networks for multi-object tracking via sinkhorn normalization. *arXiv preprint arXiv:2010.00067*, September, 2020. 2, 7
- [44] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision (ECCV)*, pages 145–161. Springer, 2020. 2, 6
- [45] Seyed Hamid Reza Tofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3047–3055, December, 2015. 2
- [46] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision (ECCV)*, pages 17–35. Springer, 2016. 6
- [47] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *IEEE International Conference on Computer Vision (ICCV)*, pages 300–311, October, 2017. 2
- [48] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Tracklets predicting based adaptive graph tracking. *arXiv preprint arXiv:2010.09015*, October, 2020. 2, 6
- [49] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, April, 2018. 6, 11, 12, 15, 16
- [50] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, December, 2020. 1, 2, 3, 6, 11, 12

- [51] Siyu Tang, Bjoern Andres, Miykhaylo Andriluka, and Bernt Schiele. Subgraph decomposition for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5033–5041, June, 2015. 2
- [52] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision (ECCV)*, pages 100–111. Springer, 2016. 2
- [53] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3548, July, 2017. 2
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Neural Information Processing Systems (NeurIPS)*, June, 2017. 1, 2, 4
- [55] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*, June, 2020. 2, 6, 7
- [56] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6499–6508, June, 2020. 2
- [57] Xinshuo Weng, Ye Yuan, and Kris Kitani. Joint 3d tracking and forecasting with graph neural network and diversity sampling. *arXiv preprint arXiv:2003.07847*, March, 2020. 2
- [58] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint Detection and Identification Feature Learning for Person Search. *arXiv preprint arXiv:1604.01850*, April, 2016. 6
- [59] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6787–6796, June, 2020. 2, 3, 6
- [60] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Bain-ing Guo. Learning texture transformer network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5791–5800, June, 2020. 1, 3
- [61] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3221, July, 2017. 6
- [62] Yang Zhang, Hao Sheng, Yubin Wu, Shuai Wang, Wei Ke, and Zhang Xiong. Multiplex labeling graph for near-online tracking in crowded scenes. *IEEE Internet of Things Journal*, 7(9):7892–7902, May, 2020. 2, 6, 7
- [63] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, April, 2020. 1, 2, 3, 5, 6, 7
- [64] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification:

A benchmark. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, December, 2015. 6

- [65] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1367–1376, July, 2017. 6
- [66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision (ECCV)*, pages 474–490. Springer, 2020. 1, 2, 3, 5, 6
- [67] 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. 1, 2, 3, 4, 8

Furthermore, we first complete the ablation study by showing the benefit of using dense queries. We compare the tracking performance between the transformer-based MOT model using image-independent sparse queries and our TransCenter. Second, we show some qualitative results of TransCenter. Finally, we provide the per-seq full state-of-the-art results of our best model fine-tuned from CrowdHuman submitted to the MOTchallenge.

## A. Sparse V.S. Dense Queries Models

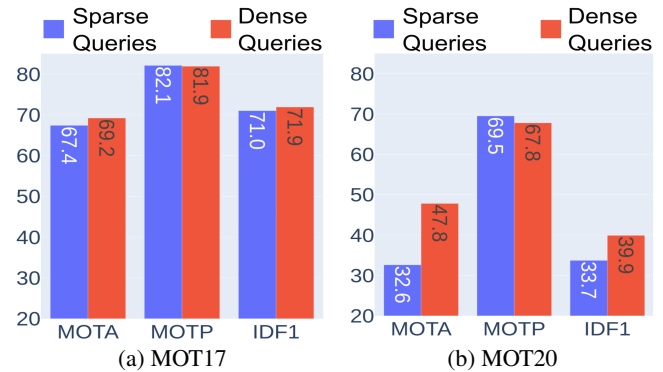


Figure 7: FP, FN, IDS ablation studies on MOT17, MOT20 validation of models trained on half MOT17.

Table 4: FP, FN, IDS on MOT17 and crowded scenes MOT20 validation of models trained on half MOT17.

Method	MOT17			MOT20		
	FP ↓	FN ↓	IDS ↓	FP ↓	FN ↓	IDS ↓
Sparse Queries [50]	1,086	7526	190	13,989	190,689	2,496
Dense Queries (ours)	1,202	6,951	203	12,337	145,546	2,889

Both models are pre-trained on CrowdHuman [49] and finetuned on the first half of sequences of MOT17 [40] dataset. From Fig. 7, we see that TransCenter outperforms the method [50] using sparse queries (+2% MOTA,

+0.9% IDF1) on MOT17 [40]. Without fine-tuning on MOT20 [10], we observe a great discrepancy between the performance of the method using dense and sparse queries (+15.2% MOTA and +6.2% IDF1).

The discrepancy is also reflected in Tab. 4, compared to [50] in MOT20 [10], TransCenter, without training on MOT20, can help detect much more objects (-45,143 FNs) while having fewer FPs (-1,652). The rise of IDS is due to the fact that we have more detected objects causing more severe occlusions.

The reason is because of the use of pixel-level queries correlated to the input image. Independent of the number of objects in the image, we do not need to re-parameterize the number of queries according to the number of objects in the image as models using image-independent sparse queries. TransCenter thus generalizes better in more crowded scenes.

## B. Qualitative Visualization

We visualize some qualitative results in Fig. 8 and Fig. 9 on the MOT20 testset showing the capability of TransCenter in tracking people in very crowded scenes.

## C. Detailed Results

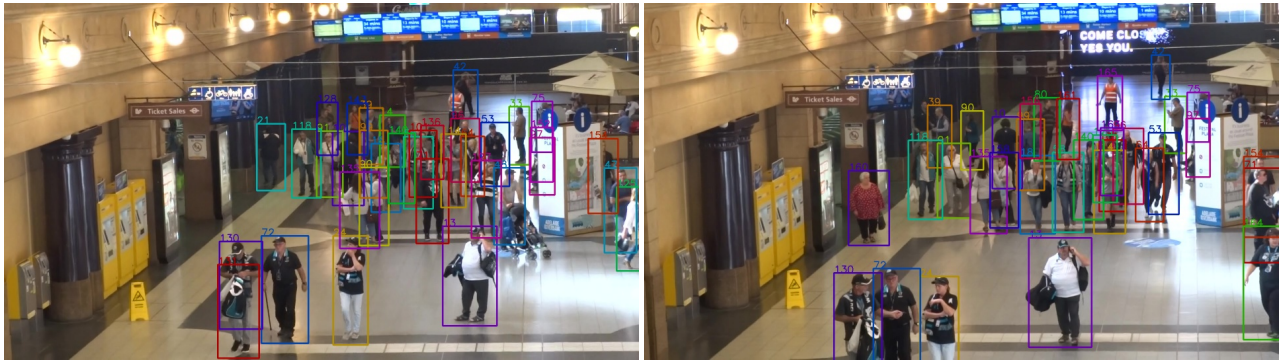
We provide the detailed results on MOT17 [40] (see Tab. 5) and MOT20 [10] (see Tab. 6) testsets with TransCenter trained on CrowdHuman [49] and MOT17 trainset or MOT20 trainset, respectively.



(a)

(b)

MOT20-04



(c)

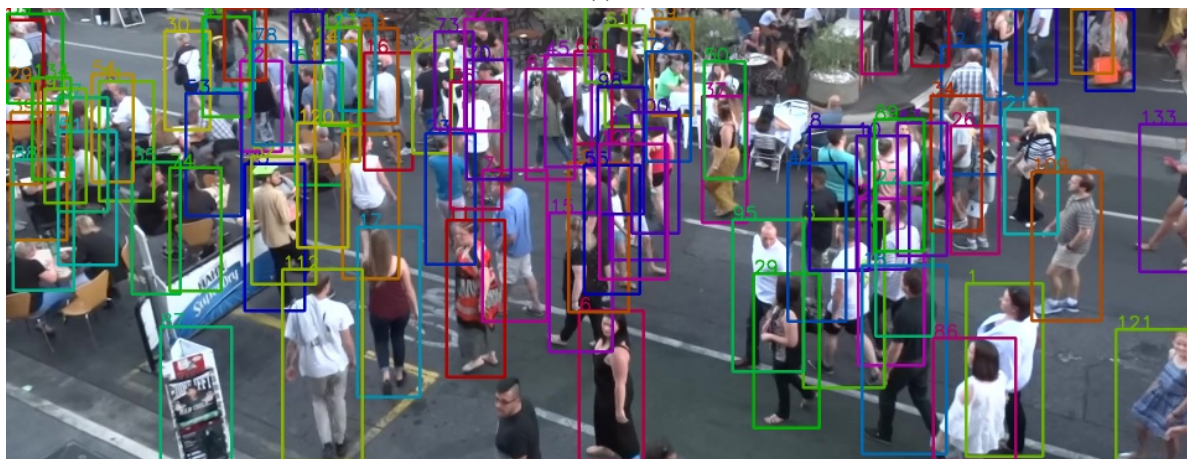
(d)

MOT20-07

Figure 8: Tracking results visualization of TransCenter on MOT20-04, MOT20-07 in the test set using the Private Detection setting.



(a)



(b)

MOT20-06

Figure 9: Tracking results visualization of TransCenter on MOT20-06 test set, in the Private Detection setting.

Table 5: Per-sequence detailed results on MOT17 [40] testset for TransCenter trained on CrowdHuman [49] and MOT17 [40]. In the private detection setting, the results for DPM, SDP and FRCNN are the same. We, therefore, do not specify their associated public detections.

	Sequence	MOTA $\uparrow$	MOTP $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDS $\downarrow$
Public Dets.	MOT17-01-DPM	49.8	78.8	40.3	8	9	448	2,745	46
	MOT17-01-FRCNN	49.8	78.8	40.1	8	9	480	2,709	47
	MOT17-01-SDP	50.4	78.7	39.7	8	9	490	2,662	49
	MOT17-03-DPM	89.6	82.0	73.8	126	5	2,514	8,167	226
	MOT17-03-FRCNN	88.2	82.1	73.7	123	8	2,505	9,619	224
	MOT17-03-SDP	88.9	82.0	73.0	122	9	2,731	8,697	240
	MOT17-06-DPM	61.2	80.8	56.3	76	56	497	3,900	170
	MOT17-06-FRCNN	63.5	80.5	56.8	82	42	543	3,552	201
	MOT17-06-SDP	62.9	80.6	56.9	84	50	556	3,617	194
	MOT17-07-DPM	58.7	79.4	48.1	15	8	683	6,100	190
	MOT17-07-FRCNN	58.8	79.4	48.5	15	6	674	6,085	194
	MOT17-07-SDP	59.8	79.3	47.7	16	6	702	5,885	197
	MOT17-08-DPM	46.2	81.5	36.1	22	21	422	10,662	280
	MOT17-08-FRCNN	45.7	81.6	36.5	21	21	395	10,815	269
	MOT17-08-SDP	46.6	81.4	36.1	22	20	427	10,571	279
	MOT17-12-DPM	59.5	84.0	62.3	30	28	334	3,121	51
	MOT17-12-FRCNN	59.3	84.0	61.8	30	29	272	3,208	50
	MOT17-12-SDP	59.7	83.8	61.7	30	27	361	3,077	53
	MOT17-14-DPM	34.9	76.3	36.5	17	61	674	11,050	316
	MOT17-14-FRCNN	36.9	75.8	37.1	19	55	843	10,445	383
	MOT17-14-SDP	37.6	75.7	38.2	20	55	827	10,321	387
	<b>MOT17-all</b>	71.9	81.4	62.3	894 (38.0%)	534 (22.7%)	17,378	137,008	4,046
Private det.	MOT17-01	49.3	78.6	39.7	8	9	568	2,650	49
	MOT17-03	90.6	81.8	73.5	136	0	3,410	6,116	266
	MOT17-06	64.0	80.4	56.4	87	35	651	3,364	227
	MOT17-07	60.0	79.2	47.9	16	6	807	5,755	200
	MOT17-08	47.2	81.3	36.3	22	18	445	10,423	286
	MOT17-12	57.3	83.5	60.6	30	24	666	2,963	69
	MOT17-14	37.4	75.4	37.5	21	53	1,157	9,975	441
	<b>MOT17-all</b>	73.2	81.1	62.2	960 (40.8%)	435 (18.5%)	23,112	123,738	4,614

Table 6: Per-sequence detailed results on MOT20 [40] testset for TransCenter trained on CrowdHuman [49] and MOT20 [40].

	Sequence	MOTA $\uparrow$	MOTP $\uparrow$	IDF1 $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	IDS $\downarrow$
Public Dets.	MOT20-04	68.8	80.4	54.4	240	82	11,020	72,756	1,732
	MOT20-06	46.5	78.8	34.4	88	75	12,109	57,202	1,761
	MOT20-07	76.1	81.5	61.7	79	7	1,925	5,719	271
	MOT20-08	35.6	77.2	30.6	34	68	8,637	40,164	1,086
	<b>MOT20-all</b>	58.6	79.8	46.7	441 (35.5%)	232 (18.7%)	33,691	175,841	4,850
Private det.	MOT20-04	68.7	80.4	54.1	240	82	11,289	72,674	1,730
	MOT20-06	46.0	78.7	35.3	88	75	12,947	56,915	1,801
	MOT20-07	75.4	81.5	61.5	80	7	2,173	5,687	271
	MOT20-08	35.1	77.1	30.8	36	67	9,550	39,617	1,145
	<b>MOT20-all</b>	58.3	79.7	46.8	444 (35.7%)	231 (18.6%)	35,959	174,893	4,947