

TransTrack: Multiple Object Tracking with Transformer

Peize Sun¹, Jinkun Cao², Yi Jiang³, Rufeng Zhang⁴, Enze Xie¹, Zehuan Yuan³, Changhu Wang³, Ping Luo¹

¹The University of Hong Kong ²Carnegie Mellon University
³ByteDance AI Lab ⁴Tongji University

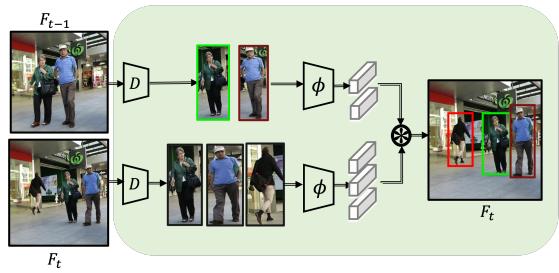
Abstract

In this work, we propose **TransTrack**, a simple but efficient scheme to solve the multiple object tracking problems. TransTrack leverages the transformer architecture, which is an attention-based query-key mechanism. It applies object features from the previous frame as a query of the current frame and introduces a set of learned object queries to enable detecting new-coming objects. It builds up a novel joint-detection-and-tracking paradigm by accomplishing object detection and object association in a single shot, simplifying complicated multi-step settings in tracking-by-detection methods. On MOT17 and MOT20 benchmark, TransTrack achieves 74.5% and 64.5% MOTA, respectively, competitive to the state-of-the-art methods. We expect TransTrack to provide a novel perspective for multiple object tracking. The code is available at: <https://github.com/PeizeSun/TransTrack>.

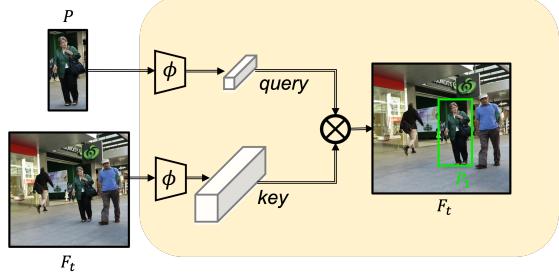
1. Introduction

Visual object tracking is a vital problem in many practical applications, such as visual surveillance, public security, video analysis, and human-computer interaction. According to the number of objects to track, the task of object tracking is divided into **Single Object Tracking (SOT)** and **Multiple Object Tracking (MOT)**. In recent years, the emerging of deep siamese networks [3, 37, 20, 19] have made great progress in solving SOT tasks. However, the existing MOT methods are still suffering from the model complexity and computational cost due to the multi-stage pipeline [50, 36, 43] as shown in Figure 1a.

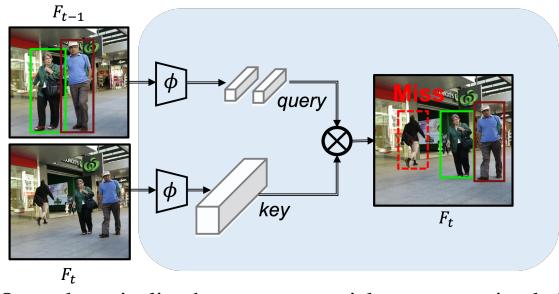
A critical dilemma in many existing MOT solutions is when object detection and re-identification are performed separately, they can not benefit each other. To tackle the problem in MOT, a joint-detection-and-tracking framework is needed to share knowledge between detection and object association. By reviewing SOT solutions, we emphasize that **Query-Key** mechanism is promising in this direc-



(a) Complex tracking-by-detection MOT pipeline.



(b) Simple query-key SOT pipeline.



(c) Query-key pipeline has great potential to setup a simple MOT method. However, it will miss new-coming objects.

Figure 1: **Motivation of TransTrack.** The dominant MOT method is the complex multi-step tracking-by-detection pipeline. Directly migrating the query-key mechanism from SOT to MOT will cause severe missing of new-coming objects. TransTrack is aimed to take advantage of query-key mechanism and to detect new-coming objects. The pipeline is shown in Figure 2.

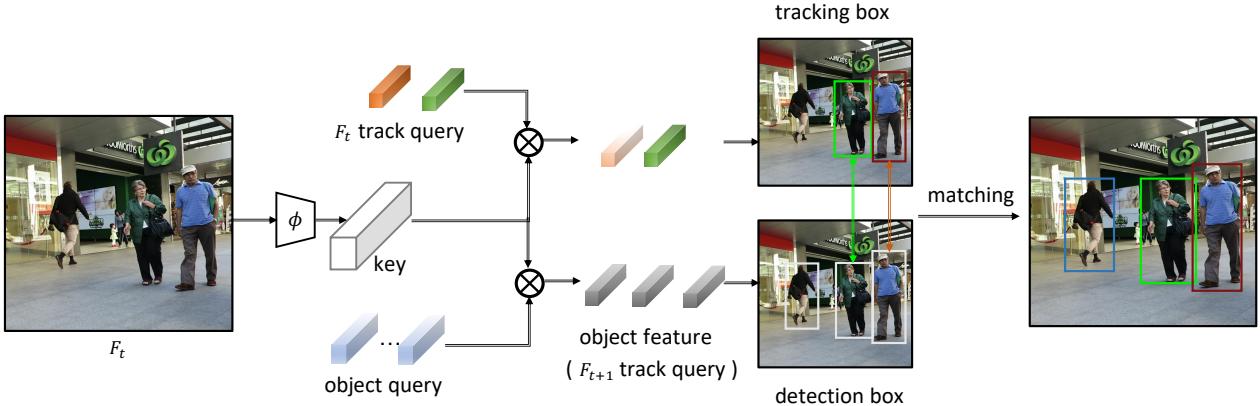


Figure 2: Pipeline of TransTrack. Both object feature query from the previous frame and learned object queries are taken as input. The image feature maps are a shared key. The learned object query detects objects in the current frame. The track query from the previous frame associates objects of the current frame with the previous ones. This process is performed sequentially over all adjacent frames and finally completes the multiple object tracking tasks.

tion. In existing works, the object target is the query and the image regions are the keys as shown in Figure 1b. For the same object, its feature in different frames is highly similar, which enables the query-key mechanism to output ordered object sets. This inspiration should also be beneficial to the MOT task.

However, merely transferring the vanilla query-key mechanism from SOT into the MOT task leads to poor performance, significantly causing much more false negatives. It is because when a new object comes into birth, there is no corresponding features for it. This defect causes severe object missing, as shown in Figure 1c. So what is a suitable query-key mechanism for MOT remains a critical question. A desirable solution should be able to well capture new-coming objects and propagate previously detected objects to the following frames at the same time.

In this paper, we make efforts in this direction by building an MOT framework based on transformer [38], which is an attention-based query-key mechanism. We term it as **TransTrack**. It leverages set prediction for detection [5] and the knowledge passed from the previous frame to gain reliable object association at the same time. There are two sets of keys (following previous works [5], they are confusingly termed as “object query” in transformer). One set contains the object queries learned as in existing transformer-based detector [5] and the other contains those generated from the features of objects on the previous frame, which are also termed as “track query” for clarification. The first set of queries provides a sense of new-coming objects and the track queries provide consistent object information to maintain tracklets. Two sets of bounding boxes are predicted respectively and TransTrack uses simple IoU matching to generate the final ordered object set from them.

In TransTrack, the two sets of boxes can be output from

a uniform decoder architecture with only different queries as input. Our model even removes the traditional NMS stage in detection. Therefore, our method is simple and straightforward where all components of the model can be trained at the same time. We evaluate TransTrack on the two real-world benchmarks MOT17 and MOT20 [26, 7]. It achieves 74.5 and 64.5 MOTA on the test set of MOT17 and MOT20 respectively. To the best of our knowledge, we are the first to introduce the transformer in the MOT task. As it has achieved comparable performance with state-of-the-art models, we hope it could provide a new perspective and efficient baseline for multi-object tracking tasks.

2. Related Work

In this section, we first review previous transformer applications in vision tasks. Then we introduce the two main MOT paradigms, namely tracking-by-detection and joint-detection-and-tracking methods.

Transformer in vision tasks. Recently, there is a popularity of using transformer architecture [38] in vision tasks, where it has been proven powerful and inspiring. As a special query-key mechanism, the transformer heavily relies on the attention mechanism to process extracted deep features. It first shows great efficiency in natural language processing [38] and later migrated to visual perception tasks [5] achieving remarkable success. Transformer appeals to the vision community with elegant structure and good performance. It has shown great potential in detection [5, 60], segmentation [57], 3D data processing [55] and even backbone construction [11]. Lately, the good efforts of using a transformer in processing sequential visual data also make remarkable shots in video segmentation [42]. With the natural strength of passing features along the temporal dimension,

the transformer shows the ability to contribute to diverse temporal-spatial processing tasks on visual data and even replaces the role of traditional RNN models [16]. However, to the best of our knowledge, there are still no published transformer-based solutions for object tracking while it is intuitive to leverage its demonstrated good capacity in visual perception and temporal processing there. Hence, in this paper, we follow the insight to propose a transformer-based model for MOT. It shows convincingly high performance on the popular MOT benchmark.

Tracking-by-detection. State-of-the-art multiple object trackers are mostly dominated by the tracking-by-detection paradigm. It firstly uses the object detectors [23, 30, 22] to localize all objects of interest, then associates these detected objects according to their Re-ID features and/or other information, e.g., Intersection over Unions (IoU) between each other. SORT [4] tracks bounding boxes using the Kalman Filter [44] and associates to the current frame by the Hungarian algorithm [18]. DeepSORT [45] replaces the association cost in SORT with the appearance features from deep convolutional networks. POI [50] achieves state-of-the-art tracking performance based on the high-performance detection and deep learning-based appearance feature. Lifted-Multicut [36] combines the deep representations and body pose feature obtained by the pose estimation model. STRN [48] presents a similarity learning framework between tracks and objects, which encodes various Spatial-Temporal relations. Tracking-by-detection pipeline achieves leading performance, but its model complexity and computational cost are not satisfying.

Joint-detection-and-tracking. The joint-detection-and-tracking pipeline aims to achieve detection and tracking simultaneously in a single stage. D&T [13] proposes a multi-task architecture for frame-based object detection and across-frame track regression. Integrated-Detection [54] boosts the detection performance by combining the detection bounding boxes in the current frame and tracks in previous frames. More recently, Tracktor [1] directly uses the previous frame tracking boxes as region proposals and then applies the bounding box regression to provide tracking boxes on the current step, thus eliminating the box association procedure. JDE [43] and FairMOT [51] learn the object detection task and appearance embedding task from a shared backbone. CenterTrack [58] localizes objects by tracking-conditioned detection and predicts their offsets to the previous frame. ChainedTracker [29] chains paired bounding boxes estimated from overlapping nodes, in which each node covers two adjacent frames. Our proposed TransTrack falls into the joint-detection-and-tracking category. Previous works adopt anchor-based [30] or point-based [59] detection framework. Instead, we build the pipeline based on a query-key mechanism and the tracked

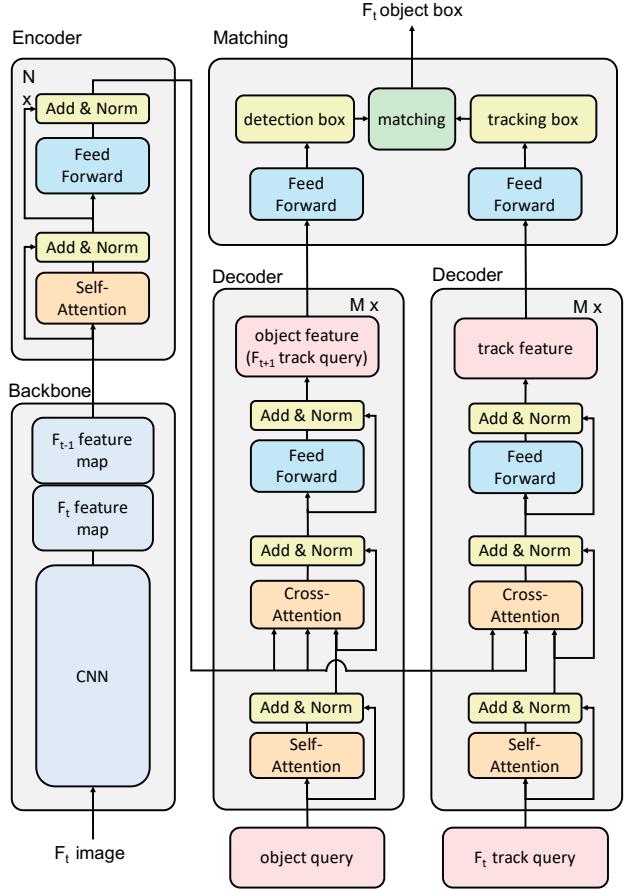


Figure 3: **The architecture details of TransTrack.** First, the current frame image is input to CNN backbone to extract feature map. Then, both the current frame feature map and the previous one are fed into encoder to generate composite feature. Next, learned object query is decoded into detection boxes and object feature of the previous frame is decoded into tracking boxes. Finally, IoU matching is employed to associate detection boxes to tracking boxes.

object feature is used as the query.

3. TransTrack

In MOT task, the desirable output is a **complete** and **correctly ordered** set of objects on each frame in a video. To these two ends, TransTrack uses queries from two sources to gain adaptive cues. On the one hand, similar to usual transformer-based detectors [5, 60], TransTrack takes an object query as input to provide common object detection results. On the other hand, TransTrack leverages features from previously detected objects to form another “track query” to discover associated objects on the following frames. Under this scheme, TransTrack generates in parallel two sets of bounding boxes, termed as “detection

boxes” and “tracking boxes”. Last, TransTrack uses the Hungarian algorithm, where the cost is IoU area among boxes, to achieve the final ordered box set from the two bounding box sets. The pipeline is illustrated in Figure 3.

3.1. Pipeline

In this section, we introduce the encoder-decoder architecture of TransTrack for object detection and object propagation. Given the detection boxes and tracking boxes from two decoders, box IoU matching is used to obtain the final tracking result. We also introduce the training and inference process of TransTrack.

Architecture. TransTrack is based on transformer, an encoder-decoder framework. It replies on stacked multi-head attention layers and feed-forward networks. Multi-head attention is called self-attention if the input query and the input key are the same, otherwise, cross-attention. In transformer architecture, The encoder generates keys and the decoder takes as input task-specific queries. The architecture overview is shown in Figure 3.

The encoder of TransTrack takes the composed feature maps of two consecutive frames as input. To avoid duplicated computation, the extracted features of the current frame are temporarily saved and then re-used for the next frame. Two parallel decoders are employed in TransTrack. Feature maps generated from the encoder are used as common keys by the two decoders. The two decoders are designed to perform object detection and object propagation, respectively. Specifically, a decoder takes learned object query as input and predicts *detection boxes*. The other decoder takes the object feature from previous frames, namely “track query”, as input and predicts the locations of the corresponding objects on the current frame, whose bounding boxes are termed as *tracking boxes*.

Object Detection. Following DETR [5], TransTrack leverages learned object query for object detection. The object query is a set of learnable parameters, trained together with all other parameters in the network. During detection, the key is the global feature maps generated from the input image and the object query looks up objects of interest in the image and outputs the final detection predictions, termed as “detection boxes”. This stage is performed by the left-hand decoder block in Figure 3.

Object Propagation. Given detected objects in the previous frame, TransTrack propagates these objects by passing their features to the next frame as the track query. The stage is performed by the right-hand decoder block in Figure 3. The decoder has the same architecture as the left-hand one but takes queries from different sources. This inherited object feature conveys the appearance and location information of previously seen objects, so this decoder could well locate the position of the corresponding object on the cur-

rent frame and output “tracking boxes”.

Box Association. Provided the detection boxes and tracking boxes, TransTrack uses the box IoU matching method to get the final tracking result, as shown in Figure 3. Applying the Kuhn-Munkres (KM) algorithm [18] to IoU similarity of detection boxes and tracking boxes, detection boxes are matched to tracking boxes. Those unmatched detection boxes are kept to create new tracklets.

3.2. Training

Training Data. We build training dataset from two sources. As usual, the training data of could be two consecutive frames or two randomly selected frames from a real video clip. Furthermore, training data could also be the static image [58], where the adjacent frame is simulated by randomly scaling and translating the static image.

Training Loss. In TransTrack, tracking boxes and detection boxes are the predictions of object boxes in the same image. It allows us to simultaneously train two decoders by the same training loss.

TransTrack applies set prediction loss to supervise detection boxes and tracking boxes of classification and box coordinates. Set-based loss produces an optimal bipartite matching between predictions and ground truth objects. Following [5, 60, 35, 34, 39], the matching cost is defined as

$$\mathcal{L} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{giou} \cdot \mathcal{L}_{giou} \quad (1)$$

where \mathcal{L}_{cls} is focal loss [23] of predicted classifications and ground truth category labels, \mathcal{L}_{L1} and \mathcal{L}_{giou} are L1 loss and generalized IoU loss [31] between normalized center coordinates and height and width of predicted boxes and ground truth box, respectively. λ_{cls} , λ_{L1} and λ_{giou} are coefficients of each component. The training loss is the same as the matching cost except that only performed on matched pairs. The final loss is the sum of all pairs normalized by the number of objects inside the training batch.

3.3. Inference

In the inference stage, TransTrack first detects objects on the first frame, where the feature maps are from two copies of the first frame. Then TransTrack operates object propagation and box association for the following frames and finally completes tracklets over the entire video sequence.

We use track rebirth in the inference procedure of TransTrack to enhance robustness to occlusions and short-term disappearing [1, 58, 29]. Specifically, if a tracking box is unmatched, it keeps as an “inactive” tracking box until it remains unmatched for K consecutive frames. Inactive tracking boxes can be matched to detection boxes and regain their ID. Following [58], we choose $K = 32$.

Benchmark	Method	Data	MOTA \uparrow	IDF1 \uparrow	MOTP \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	IDS \downarrow
MOT17	TubeTK [27]	No	63.0	58.6	78.3	31.2	19.9	27060	177483	4137
	ChainedTracker [29]	No	66.6	57.4	78.2	32.2	24.2	22284	160491	5529
	QuasiDense [28]	No	68.7	66.3	79.0	40.6	21.9	26589	146643	3378
	GSDT [41]	5D2	73.2	66.5	80.7	41.7	17.5	26397	120666	3891
	CSTrack [21]	5D1	74.9	72.6	80.9	41.5	17.5	23847	114303	3567
	FairMOT [51]	5D1	73.7	72.3	81.3	43.2	17.3	27507	117477	3303
	FUFET [32]	5D1	76.2	68.0	81.1	51.1	13.6	32796	98475	3237
	MLT [53]	5D1	75.3	75.5	81.7	49.3	19.5	27879	109836	1719
	CorrTracker [40]	5D1	76.5	73.6	81.2	47.6	12.7	29808	99510	3369
	CenterTrack [58]	CH	67.8	64.7	78.4	34.6	24.6	18489	160332	3039
	TraDeS [46]	CH	69.1	63.9	78.9	36.4	21.5	20892	150060	3555
	TransMOT [6]	CH	76.7	75.1	82.0	51.0	16.4	36231	93150	2346
	TransCenter [49]	CH	73.2	62.2	81.1	40.8	18.5	23112	123738	4614
	TransTrack(ours)	CH	74.5	63.9	80.6	46.8	11.3	28323	112137	3663
MOT20	GSDT [41]	5D2	67.1	67.5	79.1	53.1	13.2	31507	135395	3230
	CSTrack [21]	5D1	66.6	68.6	78.8	50.4	15.5	25404	144358	3196
	FairMOT [51]	5D1	61.8	67.3	78.6	68.8	7.6	103440	88901	5243
	CorrTracker [40]	5D1	65.2	73.6	-	47.6	12.7	29808	99510	3369
	TransCenter [49]	CH	58.3	46.8	79.7	35.7	18.6	35959	174893	4947
	TransTrack(ours)	CH	64.5	59.2	80.0	49.1	13.6	28566	151377	3565

Table 1: **Evaluation on MOT17 and MOT20 test sets.** We compare TransTrack with recent methods in private protocol, where external data can be used: CH for CrowdHuman [33], 5D1 for the use of 5 extra datasets, including CrowdHuman [33], Caltech Pedestrian [9, 10], CityPersons [52], CUHK-SYS [47], and PRW [56], 5D2 is the same as 5D1 replacing CrowdHuman by ETH [12], NO for using no extra dataset.

4. Experiments

To measure the performance of our proposed method, we conduct experiments on the pedestrian-tracking dataset MOT17 [26] and MOT20 [7]. In the ablation study, we follow previous practice [58] to split the MOT17 training set into two parts, one for training and the other for validation. We adopt the widely-used MOT metrics set [2] for quantitative evaluation where multiple object tracking accuracy (MOTA) is the primary metric to measure the overall performance.

4.1. Implementation details

We use ResNet-50 [15] as the network backbone. The optimizer is AdamW [24] and the batch size is set to be 16. The initial learning rate is 2e-4 for the transformer and 2e-5 for the backbone. The weight decay is 1e-4. All transformer weights are initialized with Xavier-init [14], and the backbone model is pretrained on ImageNet [8] with frozen batch-norm layers [17]. We use data augmentation including random horizontal, random crop, scale augmentation, resizing the input images whose shorter side is by 480 - 800 pixels while the longer side is by at most 1333 pixels. We train the model for 150 epochs and the learning rate drops by a factor of 10 at the 100th epoch. In the ablation

study, the model is first pre-trained on CrowdHuman [33] and then fine-tuned on MOT. When evaluating on the test set, we train our network on combination of CrowdHuman and MOT. More details are discussed in Appendix.

4.2. MOT17 benchmark

We evaluate models on MOT17 under the private detector setting. The results We evaluate models on MOT17 under the private detector setting. The results are shown in Table 1. TransTrack achieves comparable results with the current state-of-the-art methods, especially in terms of MOTP and FN. The excellent MOTP demonstrates TransTrack can precisely locate objects in the image. The good FN score represents that most objects are successfully detected. Those prove the success of introducing learned object query into the pipeline. As for ID-switch, TransTrack is comparable with the popular trackers, *e.g.*, FairMOT [51] and CenterTrack [58], which proves the effectiveness of object feature query to associate adjacent frames. Although the ID-switch score of TransTrack is inferior to SOTA methods, it is a promising direction to further improve the overall performance of TransTrack.

4.3. MOT20 benchmark

We evaluate models on MOT20 under the private detector setting. The results are shown in Table 1. MOT20 includes more crowded scenes than MOT17. Its more severe object occlusion and smaller object size bring more challenges for object detection and tracking. Therefore, all methods show lower performance on MOT20 than on MOT17. But still, TransTrack achieves comparable results with the current state-of-the-art methods on MOT20, in terms of detection metrics and association metrics.

4.4. Ablation study

4.4.1 Transformer Architecture

We ablate the effect of Transformer architecture. Four transformer structures are put into comparison. **Transformer** follows the settings of DETR [5] detector, where transformer is built on top of the feature maps of res5 stage [15]. **Transformer-DC5** increases the feature maps resolution. To be precise, we apply dilation convolution to res5 stage and remove a stride from the first convolution of this stage. **Transformer-P3** adopts FPN [22] on the input feature maps. The encoder of the Transformer is directly removed from the whole pipeline for memory limitation. After removing the encoder, the learning rate of the backbone could be raised to the same as transformers. Finally, we also tried **Deformable Transformer** [60], which is a recently proposed architecture to solve the issue of limited resolution in the transformer. Within plausible memory usage, it fuses multiple-scale features into the whole encoder-decoder pipeline and achieves excellent performance in the general object detection dataset.

The quantitative results are shown in Table 2. The final performance of **Transformer** is only 55.4 MOTA. With higher feature resolution, **Transformer-DC5** yields 3.6 MOTA improvement. However, it also leads to the drawback of dilation convolution, such as big memory usage. **Transformer-P3** only outputs close performance as Transformer-DC5, saying that higher resolution than DC5 fails to bring further performance gain. And the reason behind this might be the absence of encoder blocks. At last, **Deformable Transformer** fuses multiple-scale feature into the whole encoder-decoder pipeline and achieves excellent performance, up to 65.0 MOTA. Therefore, we use Deformable Transformer as the default architecture choice of TransTrack.

4.4.2 Query in Decoder

We study the effect of what the input query is used. In the detection task, an input query is generated from the input image only [5, 60]. But in tracking, the knowledge of previously detected objects is expected to be helpful, so we set

Architecture	MOTA↑	FP↓	FN↓	IDs↓
Transformer [5]	55.4	7.4%	35.2%	2.0%
Transformer-DC5 [5]	59.0	5.2%	34.0%	1.8%
Transformer-P3	59.3	5.1%	33.8%	1.8%
Deformable Transformer [60]	65.0	4.3%	30.3%	0.4%

Table 2: **Ablation study on Transformer architecture.** Original transformer suffers from low feature resolution. Deformable DETR with multi-scale feature input achieves best performance.

Query	MOTA↑	FP↓	FN↓	IDs↓
Object query	58.3	4.0%	29.7%	8.0%
Track query	-	15.6%	93.8%	0.3%
Track query + Object query	65.0	4.3%	30.3%	0.4%

Table 3: **Ablation study on input query.** Using only object query obtains limited association performance. Using only track query leads to numerous FN since it misses new-coming objects. By using both object query and track query, the detection and tracking performance are improved.

experiments to compare the model performance when object query and track query are used or absent respectively. The results are reported in Table 3.

Only object query. When only learned object query is input as decoder query, we adopt a naive pipeline where the output detection boxes are associated according to their index in the output set. Surprisingly, this solution achieves a not bad performance by 58.3 MOTA. This is because each object query predicts the object in a certain area on images, and most objects just move around a small distance in the video sequence. However, solely relying on the index in the output set leads to non-negligible wrong matching, especially when the object moves through a long distance. When the object moves around a wide range, this pattern fails easily as visualized in Figure 4.

Only track query. When only the track query, which is generated from the previous frame, is input to the decoder, we have no common detection results on the image. The visualization in the second row of Figure 4 shows that this method is capable to associate the object with a large range of motion. Nevertheless, only the object that appears in the first frame can be tracked successively. For the whole video sequence, most of the objects will be missed and the FN metric collapses as shown in the second row of Table 3.

Object query + track query. As the default setting of TransTrack, both object query and track query are input to the decoder. Now it can handle most failure cases in the previous two cases with the help of the other query. Visualization in Figure 4 and performance reported in Table 3 prove the giant improvement.



Figure 4: **Visualization of TransTrack with different input query.** 1st row is **only learned object query**. 2nd row is **only object feature query from the previous frame**. 3rd row is **both learned object query and object feature query from the previous frame**. Only learned object query or object feature query from the previous frame causes ID switch case or missing object case. TransTrack takes both as input query and exhibits best detection and tracking performance.

Matching	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow
Previous	64.8	4.8%	29.8%	0.6%
Current	65.0	4.3%	30.3%	0.4%

Table 4: **Ablation study of matching strategy of tracking boxes.** **Previous** indicates directly inheriting the index of track query for box matching on the previous frame. **Current** indicates using optimal bipartite matching with current object boxes.

4.4.3 Matching strategy of tracking boxes

TransTrack builds tracklets based on two sets of detection results and box matching. To emphasize temporal correlation in tracking tasks, it is natural to consider matching tracking boxes with previous frame objects. To ablate the influence of tracking boxes matching, we conduct two strategies. One way is to match initial tracking boxes to previous object boxes by optimal bipartite matching (**Previous**), in other words, the matching index is directly from the matching index of corresponding track queries. The other strategy is to supervise the output of tracking

Association	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow
Hungarian	65.0	4.3%	30.3%	0.4%
NMS	65.0	4.3%	30.3%	0.4%

Table 5: **Ablation study of box association.** Two sets of bounding boxes, track boxes and detection boxes, are merged into the desired ordered object set. The results show that Hungarian algorithm and NMS actually have the same effect in this stage.

boxes with optimal bipartite matching to current object boxes (**Current**). The results are shown in Table 4. The results show that bipartite matching with previous frame objects does not help to void ID switch (0.6% v.s. 0.4%). This shows that the inherit the property of the query-key mechanism could well locate the position of the corresponding object on the current frame already.

4.4.4 Bounding Box Association

We study the effect of different box association post-processing strategies. We choose the classic Hungarian al-

Motion model	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow	4xIDs \downarrow
None	64.4	4.3%	30.3%	1.0%	1.2%
Kalman filter	64.9	4.3%	30.4%	0.4%	1.0%
Track query(Ours)	65.0	4.3%	30.3%	0.4%	0.5%

Table 6: **The effect of motion model.** All models use DETR as detectors. For **None**, object box is associated by IoU similarity. For **Kalman filter**, the output bounding boxes are processed by Kalman filter. **Ours** follows the two-query-set setting where track query is used to associate across-frame objects.

gorithm [18] and the NMS merging method used in [58, 25]. Results are shown in Table 5. It suggests both two strategies show equivalent effect in the box association stage.

4.5. Comparisons with other trackers

Two commonly used signals to upgrade a detector to a tracker are motion and appearance features. So “detector + motion model” and “detector + Re-ID” are widely-used and intuitive methods, thus it is necessary to compare TransTrack with these two models to have a clear idea about how much TransTrack gains from its design except for improvement from the detector it replies on.

4.5.1 Motion model

We combine the widely-used Kalman filter and DETR to build a “detector + motion model” tracker. The results are shown in Table 6. Kalman filter and our method provide similar IDs performance. We explain that the MOT17 dataset is the video sequence of high frame rate (14-30 FPS), where the object motion between two adjacent frames is minor. Therefore, different association methods make no big difference. However, when we sample one frame every 4 frames, the object motion becomes larger, then the improvement brought by the feature query is obvious (0.5% vs. 1.0%), shown in the last column of Table 6. Similar phenomena are discussed in [58].

4.5.2 Re-ID features

To maintain the joint-detection-and-tracking paradigm, we do not implement an independent Re-ID model but use the Re-ID branch to formulate a “detector + Re-ID” tracker. As features generated in the detector have conflicts with appearance-based Re-ID features [51], we study the influence of using an independent Re-ID passway, *e.g.*, a cross-attention layer in the decoder. The two patterns are illustrated in Figure 5. The results are included in Table 7. It agrees that when passed through an independent passway, the Re-ID feature brings better results than using a

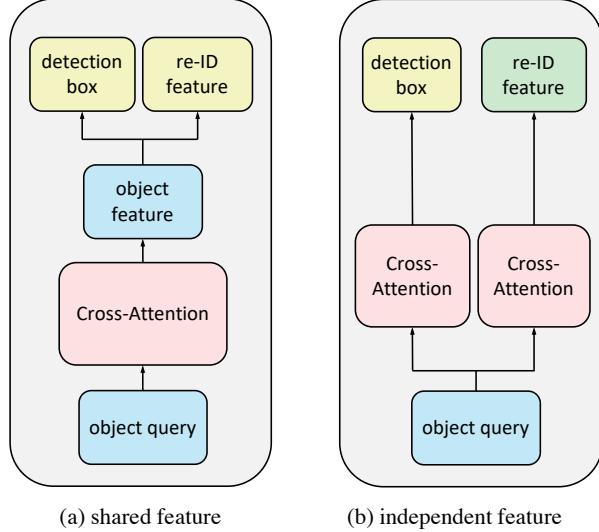


Figure 5: **Two designs to introduce Re-ID into DETR.** The left one uses a shared feature from a single cross-attention layer to train detection and re-identification. The right scheme uses two cross-attention layers to generate independent Re-ID features and detection features for the two sources of supervision.

Re-ID feature	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow
Shared	61.1	5.9%	32.3%	0.7%
Independent	64.7	3.2%	31.7%	0.4%
None (Ours)	65.0	4.3%	30.3%	0.4%

Table 7: **The effect of Re-ID features.** When passing Re-ID features to an independent cross-attention layer, the performance is better than using shared cross-attention layer for detection features and Re-ID features. However, this also results in degradation of detector, so the overall performance does not beat default TransTrack.

shared module with detection features. However, the overall MOTA score is not improved against default TransTrack.

5. Conclusion

In this work, we set up a joint-detection-and-tracking MOT pipeline, TransTrack, based on the transformer. It uses the learned object query as input to detect objects and track query, which is the features from previous frame, to propagate previously detected objects to the following frames. TransTrack is the first work solving MOT in such a paradigm. It achieves a competitive 74.5 MOTA on the MOT17 dataset and 64.5 MOTA on a more challenging MOT20 dataset. We expect it to provide a novel perspective and insight to the MOT community.

Appendix

A. Training Data

We follow the common practice of the state-of-the-art MOT methods [58] to train TransTrack on CrowdHuman [33] first and then fine-tune the model on MOT17. We conduct a comparison to study the effect of the external CrowdHuman data. The result is reported in Table 8. Only using the training set of MOT17 merely obtains 61.6 MOTA. When first pre-trained on CrowdHuman then trained on MOT17, the performance achieves 64.8 MOTA. It suggests adding external data boosts the model performance significantly.

Pre-train	Fine-tune	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow
CH	-	53.8	13.0%	32.3%	1.0%
-	MOT17	61.6	3.4%	34.2%	0.9%
CH	MOT17	65.0	4.3%	30.3%	0.4%

Table 8: **Ablation study on pre-training data.** The first row is the model trained only on CrowdHuman dataset. The second row indicates model trained on the training set of MOT dataset only. The third shows the performance when the model is trained on CrowdHuman first and then finetuned on MOT dataset. All models are evaluated on the validation set of MOT17 dataset.

Besides the pre-training data settings, we find the data used for fine-tuning is also critical. We conduct an ablation study on it and the results are shown in Table 9. Interestingly, fine-tuning on the combination of CrowdHuman and MOT shows better performance than fine-tuning on the MOT dataset only.

Dataset	Pre-train	Fine-tune	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow
MOT17	CH	MOT17	68.4	22137	152064	3942
	CH	CH+MOT17	74.5	28323	112137	3663
MOT20	CH	MOT20	57.4	32921	184047	3705
	CH	CH+MOT20	64.5	28566	151377	3565

Table 9: **Ablation study on fine-tuning data.** For each benchmark, the first row is the model fine-tuned only on MOT train dataset. The second row indicates the model fine-tuned on the combination of CrowdHuman and MOT training set. All models are evaluated on the test set of MOT benchmark.

B. Accuracy vs. Speed

We analyze the inference speed of TransTrack. The time cost is measured using a single Tesla V100 GPU. Table 10

shows the effect of number of decoders. Increasing decoders hurts inference speed, for example, from 1 decoder to 6, FPS decreases from 15FPS to 10FPS. However, more decoders significantly boost MOTA performance. Therefore, we choose 6 as the default decoder number. Table 11 shows the effect of input image size. Gradually increasing input image size, MOTA performance is saturated when the short-side of the input image is by 800 pixels so we set it as the default setting in TransTrack.

Decoders	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow	FPS
1	47.0	10.0%	40.0%	3.0%	15
3	64.3	3.3%	31.4%	1.0%	12
6	65.0	4.3%	30.3%	0.4%	10

Table 10: **Ablation study on number of decoders.** Increasing decoders has minor impact on inference time while significantly improves MOTA performance. Therefore, we choose 6 decoders as default.

Short-side	MOTA \uparrow	FP \downarrow	FN \downarrow	IDs \downarrow	FPS
540 pix	62.4	3.9%	32.8%	0.9%	14
800 pix	65.0	4.3%	30.3%	0.4%	10
1080 pix	59.2	4.7%	35.0%	1.1%	7

Table 11: **Ablation study on input image size.** Gradually increasing input image size, MOTA performance is saturated when the short-side of image is 800 pixels.

References

- [1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 3, 4
- [2] 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. 5
- [3] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking, 2016. 1
- [4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End object detection with transformers. In *ECCV*, 2020. 2, 3, 4, 6
- [6] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking. *arXiv preprint arXiv:2104.00194*, 2021. 5
- [7] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, Mar. 2020. arXiv: 2003.09003. 2, 5
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [9] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. IEEE, 2009. 5
- [10] Piotr Dollar, 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(4):743–761, 2011. 5
- [11] 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, 2020. 2
- [12] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 5
- [13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect, 2018. 3
- [14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 5
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6
- [16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3
- [17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5
- [18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3, 4, 8
- [19] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks, 2018. 1
- [20] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 1
- [21] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyeong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *arXiv preprint arXiv:2010.12138*, 2020. 5
- [22] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3, 6
- [23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 3, 4
- [24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [25] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 8
- [26] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 2, 5
- [27] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6318, 2020. 5
- [28] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. *arXiv preprint arXiv:2006.06664*, 2020. 5
- [29] 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. *arXiv preprint arXiv:2007.14557*, 2020. 3, 4, 5
- [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 3
- [31] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 4

- [32] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Fgagt: Flow-guided adaptive graph tracking. *arXiv preprint arXiv:2010.09015*, 2020. 5
- [33] 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*, 2018. 5, 9
- [34] Peize Sun, Yi Jiang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Onenet: Towards end-to-end one-stage object detection. *arXiv preprint arXiv:2012.05780*, 2020. 4
- [35] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 4
- [36] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3
- [37] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking, 2016. 1
- [38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2
- [39] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. *arXiv preprint arXiv:2012.03544*, 2020. 4
- [40] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. *arXiv preprint arXiv:2104.03541*, 2021. 5
- [41] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*, 5, 2020. 5
- [42] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers, 2020. 2
- [43] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019. 1, 3
- [44] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter, 1995. 3
- [45] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 3
- [46] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. *arXiv preprint arXiv:2103.08808*, 2021. 5
- [47] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2017. 5
- [48] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking, 2019. 3
- [49] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021. 5
- [50] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016. 1, 3
- [51] Yifu Zhan, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A simple baseline for multi-object tracking. *arXiv preprint arXiv:2004.01888*, 2020. 3, 5, 8
- [52] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017. 5
- [53] 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, 2020. 5
- [54] Zheng Zhang, Dazhi Cheng, Xizhou Zhu, Stephen Lin, and Jifeng Dai. Integrated object detection and tracking with tracklet-conditioned detection, 2018. 3
- [55] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer, 2020. 2
- [56] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1367–1376, 2017. 5
- [57] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, 2020. 2
- [58] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points, 2020. 3, 4, 5, 8, 9
- [59] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. 3
- [60] 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. 2, 3, 4, 6