TrackFormer: Multi-Object Tracking with Transformers

Tim Meinhardt[1]* Alexander Kirillov[2] Laura Leal-Taixé[1] Christoph Feichtenhofer[2]

[1]Technical University of Munich [2]Facebook AI Research (FAIR)

Abstract

The challenging task of multi-object tracking (MOT) requires simultaneous reasoning about track initialization, identity, and spatio-temporal trajectories. We formulate this task as a frame-to-frame set prediction problem and introduce TrackFormer, an end-to-end trainable MOT approach based on an encoder-decoder Transformer architecture. Our model achieves data association between frames via attention by evolving a set of track predictions through a video sequence. The Transformer decoder initializes new tracks from static object queries and autoregressively follows existing tracks in space and time with the conceptually new and identity preserving track queries. Both query types benefit from self- and encoder-decoder attention on global frame-level features, thereby omitting any additional graph optimization or modeling of motion and/or appearance. TrackFormer introduces a new tracking-by-attention paradigm and while simple in its design is able to achieve state-of-the-art performance on the task of multi-object tracking (MOT17) and segmentation (MOTS20). The code is available at https://github.com/timmeinhardt/trackformer

1. Introduction

Humans need to focus their *attention* to track objects in space and time, for example, when playing a game of tennis, golf, or pong. This challenge is only increased when tracking not one, but *multiple* objects, in crowded and real world scenarios. Following this analogy, we demonstrate the effectiveness of Transformer [50] attention for the task of multi-object tracking (MOT) in videos.

The goal in MOT is to follow the trajectories of a set of objects, *e.g*., pedestrians, while keeping their identities discriminated as they are moving throughout a video sequence. Due to the advances in image-level object detection [7, 38], most approaches follow the two-step *tracking-by-detection* paradigm: (i) detecting objects in individual video frames, and (ii) associating sets of detections between frames and thereby creating individual object tracks over time. Traditional tracking-by-detection methods associate detections via temporally sparse [22, 25] or dense [18, 21] graph optimization, or apply convolutional neural networks to predict matching scores between detections [8, 23].

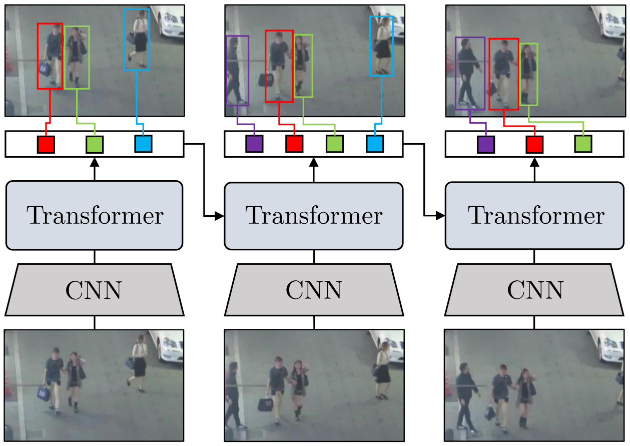

Figure 1. TrackFormer jointly performs object detection and tracking-by-attention with Transformers. Object and autoregressive *track queries* reason about track initialization, identity, and spatiotemporal trajectories.

Recent works [4,6,28,66] suggest a variation of the traditional paradigm, coined *tracking-by-regression* [12]. In this approach, the object detector not only provides frame-wise detections, but replaces the data association step with a continuous regression of each track to the changing position of its object. These approaches achieve track association implicitly, but provide top performance only by relying either on additional graph optimization [6, 28] or motion and appearance models [4]. This is largely due to the isolated and local bounding box regression which lacks any notion of object identity or global communication between tracks.

In this work, we introduce the *tracking-by-attention* paradigm which not only applies attention for data association [11, 67] but jointly performs tracking and detection. As shown in Figure 1, this is achieved by evolving a set of tracks from frame to frame forming trajectories over time.

*Work done during an internship at Facebook AI Research (FAIR).

We present a first straightforward instantiation of tracking-by-attention, TrackFormer, an end-to-end trainable Transformer [50] encoder-decoder architecture. It encodes frame-level features from a convolutional neural network (CNN) [17] and decodes queries into bounding boxes associated with identities. The data association is performed through the novel and simple concept of *track queries*. Each query represents an object and follows it in space and time over the course of a video sequence in an autoregressive fashion. New objects entering the scene are detected by static object queries as in [7, 68] and subsequently transform to future track queries. At each frame, the encoder-decoder computes attention between the input image features and the track as well as object queries, and outputs bounding boxes with assigned identities. Thereby, TrackFormer performs tracking-by-attention and achieves detection and data association jointly without relying on any additional track matching, graph optimization, or explicit modeling of motion and/or appearance. In contrast to tracking-by-detection/regression, our approach detects and associates tracks simultaneously in a single step via attention (and not regression). TrackFormer extends the recently proposed set prediction objective for object detection [7, 47, 68] to multi-object tracking.

We evaluate TrackFormer on the MOT17 [29] benchmark where it achieves state-of-the-art performance for public and private detections. Furthermore, we demonstrate the extension with a mask prediction head and show state-of-the-art results on the Multi-Object Tracking and Segmentation (MOTS20) challenge [51]. We hope this simple yet powerful baseline will inspire researchers to explore the potential of the tracking-by-attention paradigm.

In summary, we make the following contributions:

- An end-to-end trainable multi-object tracking approach which achieves detection and data association in a new tracking-by-attention paradigm.
- The concept of autoregressive track queries which embed an object's spatial position and identity, thereby tracking it in space and time.
- The TrackFormer model which obtains state-of-the-art results on two challenging multi-object tracking (MOT17) and segmentation (MOTS20) benchmarks.

2. Related work

In light of the recent trend in MOT to look beyond tracking-by-detection, we categorize and review methods according to their respective tracking paradigm.

Tracking-by-detection approaches form trajectories by associating a given set of detections over time.

Graphs have been used for track association and long-term re-identification by formulating the problem as a maximum flow (minimum cost) optimization [3] with distance based [20, 36, 62] or learned costs [24]. Other methods use association graphs [45], learned models [22], and motion information [21], general-purpose solvers [61], multi-cuts [48], weighted graph labeling [18], edge lifting [19], or trainable graph neural networks [6, 54]. However, graph-based approaches suffer from expensive optimization routines, limiting their practical application for online tracking.

Appearance driven methods capitalize on increasingly powerful image recognition backbones to track objects by relying on similarity measures given by twin neural networks [23], learned reID features [32, 41], detection candidate selection [8] or affinity estimation [10]. Similar to re-identification, appearance models struggle in crowded scenarios with many object-object-occlusions.

Motion can be modelled for trajectory prediction [1, 25, 42] using a constant velocity assumption (CVA) [2, 9] or the social force model [25, 34, 43, 58]. Learning a motion model from data [24] accomplishes track association between frames [63]. However, the projection of non-linear 3D motion [49] into the 2D image domain still poses a challenging problem for many models.

Tracking-by-regression refrains from associating detections between frames but instead accomplishes tracking by regressing past object locations to their new positions in the current frame. Previous efforts [4, 14] use regression heads on region-pooled object features. In [66], objects are represented as center points which allow for an association by a distance-based greedy matching algorithm. To overcome their lacking notion of object identity and global track reasoning, additional re-identification and motion models [4], as well as traditional [28] and learned [6] graph methods have been necessary to achieve top performance.

Tracking-by-segmentation not only predicts object masks but leverages the pixel-level information to mitigate issues with crowdedness and ambiguous backgrounds. Prior attempts used category-agnostic image segmentation [30], applied Mask R-CNN [16] with 3D convolutions [51], mask pooling layers [37], or represented objects as unordered point clouds [57] and cost volumes [56]. However, the scarcity of annotated MOT segmentation data makes modern approaches still rely on bounding boxes.

Attention for image recognition correlates each element of the input with respect to the others and is used in Transformers [50] for image generation [33] and object detection [7, 68]. For MOT, attention has only been used to associate a given set of object detections [11, 67], not tackling the detection and tracking problem jointly.

In contrast, TrackFormer casts the entire tracking objective into a single set prediction problem, applying attention not only for the association step. It jointly reasons about track initialization, identity, and spatio-temporal trajectories. We only rely on feature-level attention and avoid additional graph optimization and appearance/motion models.

3. TrackFormer

We present TrackFormer, an end-to-end trainable multi-object tracking (MOT) approach based on an encoder-decoder Transformer [50] architecture. This section describes how we cast MOT as a set prediction problem and introduce the new *tracking-by-attention* paradigm. Furthermore, we explain the concept of *track queries* and their application for frame-to-frame data association.

3.1. MOT as a set prediction problem

Given a video sequence with K individual object identities, MOT describes the task of generating ordered tracks $T_k = (b^k_{t_1}, b^k_{t_2}, \dots)$ with bounding boxes b_t and track identities k. The subset $(t_1, t_2, \dots)$ of total frames T indicates the time span between an object entering and leaving the the scene. These include all frames for which an object is occluded by either the background or other objects.

In order to cast MOT as a set prediction problem, we leverage an encoder-decoder Transformer architecture. Our model performs online tracking and yields per-frame object bounding boxes and class predictions associated with identities in four consecutive steps:

(i) Frame-level feature extraction with a common CNN backbone, *e.g.*, ResNet-50 [17].

(ii) Encoding of frame features with self-attention in a Transformer encoder [50].

(iii) Decoding of queries with self- and encoder-decoder attention in a Transformer decoder [50].

(iv) Mapping of queries to box and class predictions using multilayer perceptrons (MLP).

Objects are implicitly represented in the decoder *queries*, which are embeddings used by the decoder to output bounding box coordinates and class predictions. The decoder alternates between two types of attention: (i) self-attention over all queries, which allows for joint reasoning about the objects in a scene and (ii) encoder-decoder attention, which gives queries global access to the visual information of the encoded features. The output embeddings accumulate bounding box and class information over multiple decoding layers. The permutation invariance of Transformers requires additive feature and object encodings for the frame features and decoder queries, respectively.

3.2. Tracking-by-attention with queries

The total set of output embeddings is initialized with two types of query encodings: (i) static object queries, which allow the model to initialize tracks at any frame of the video, and (ii) autoregressive track queries, which are responsible for tracking objects across frames.

The simultaneous decoding of object and track queries allows our model to perform detection and tracking in a unified way, thereby introducing a new *tracking-by-attention* paradigm. Different tracking-by-X approaches are defined by their key component responsible for track generation. For tracking-by-detection, the tracking is performed by computing/modelling distances between frame-wise object detections. The tracking-by-regression paradigm also performs object detection, but tracks are generated by regressing each object box to its new position in the current frame. Technically, our TrackFormer also performs regression in the mapping of object embeddings with MLPs. However, the actual track association happens earlier via attention in the Transformer decoder. A detailed architecture overview which illustrates the integration of track and object queries into the Transformer decoder is shown in the appendix.

Track initialization. New objects appearing in the scene are detected by a fixed number of N_{object} output embeddings each initialized with a static and learned object encoding referred to as *object queries* [7]. Intuitively, each object query learns to predict objects with certain spatial properties, such as bounding box size and position. The decoder self-attention relies on the object encoding to avoid duplicate detections and to reason about spatial and categorical relations of objects. The number of object queries is ought to exceed the maximum number of objects per frame.

Track queries. In order to achieve frame-to-frame track generation, we introduce the concept of *track queries* to the decoder. Track queries follow objects through a video sequence carrying over their identity information while adapting to their changing position in an autoregressive manner.

For this purpose, each new object detection initializes a track query with the corresponding output embedding of the previous frame. The Transformer encoder-decoder performs attention on frame features and decoder queries *continuously updating* the instance-specific representation of an object‘s identity and location in each track query embedding. Self-attention over the joint set of both query types allows for the detection of new objects while simultaneously avoiding re-detection of already tracked objects.

In Figure 2, we provide a visual illustration of the track query concept. The initial detections in frame $t = 0$ spawn new track queries following their corresponding objects to frame t and beyond. To this end, N_{object} ob-

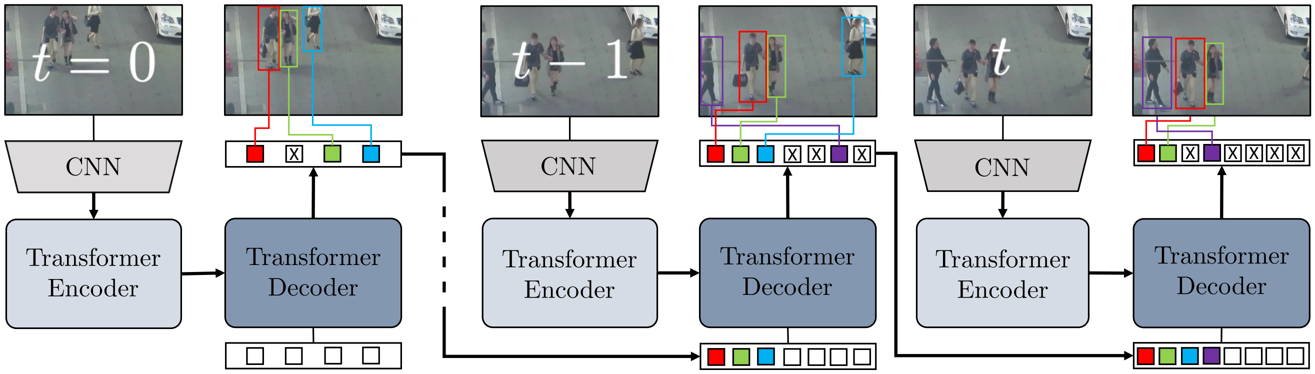

Figure 2. **TrackFormer** casts multi-object tracking as a set prediction problem performing joint detection and **tracking-by-attention**. The architecture consists of a CNN for image feature extraction, a Transformer [50] encoder for image feature encoding and a Transformer decoder which applies self- and encoder-decoder attention to produce output embeddings with bounding box and class information. At frame $t = 0$, the decoder transforms N_{object} object queries (white) to output embeddings either initializing new autoregressive **track queries** or predicting the background class (crossed). On subsequent frames, the decoder processes the joint set of $N_{\text{object}} + N_{\text{track}}$ queries to follow or remove (blue) existing tracks as well as initialize new tracks (purple).

ject queries (white) are decoded to output embeddings for potential track initializations. Each valid object detection $\{\boldsymbol{b}_0^0, \boldsymbol{b}_0^1, \dots\}$ with a classification score above σ_{object}, *i.e.*, output embedding not predicting the background class (crossed), initializes a new track query embedding. Since not all objects in a sequence appear on the first frame, the track identities $K_{t=0} = \{0, 1, \dots\}$ only represent a subset of all K. For the decoding step at any frame $t > 0$, track queries initialize additional output embeddings associated with different identities (colored). The joint set of $N_{\text{object}} + N_{\text{track}}$ output embeddings is initialized by (learned) object and (temporally adapted) track queries, respectively.

The Transformer decoder transforms the entire set of output embeddings at once and provides the input for the subsequent MLPs to predict bounding boxes and classes for frame t. The number of track queries N_{track} changes between frames as new objects are detected or tracks removed. Tracks and their corresponding query can be removed either if their classification score drops below σ_{track} or by non-maximum suppression (NMS) with an IoU threshold of σ_{NMS}. A comparatively high σ_{NMS} only removes strongly overlapping duplicate bounding boxes which we found to not be resolvable by the decoder self-attention.

Track query re-identification. The ability to decode an arbitrary number of track queries allows for an attention-based short-term re-identification process. We keep decoding previously removed track queries for a maximum number of $T_{\text{track-reid}}$ frames. During this *patience window*, track queries are considered to be inactive and do not contribute to the trajectory until a classification score higher than $\sigma_{\text{track-reid}}$ triggers a re-identification. The spatial information embedded into each track query prevents their application for long-term occlusions with large object movement, but, nevertheless, allows for a short-term recovery from track loss. This is possible without any dedicated re-identification training; and furthermore, cements TrackFormer's holistic approach by relying on the same attention mechanism as for track initialization, identity preservation and trajectory forming even through short-term *occlusions*.

3.3. TrackFormer training

For track queries to work in interaction with object queries and follow objects to the next frame, TrackFormer requires dedicated frame-to-frame tracking training. As indicated in Figure 2, we train on two adjacent frames and optimize the entire MOT objective at once. The loss for frame t measures the set prediction of all output embeddings $N = N_{\text{object}} + N_{\text{track}}$ with respect to the ground truth objects in terms of class and bounding box prediction.

The set prediction loss is computed in two steps:

(i) Object detection on frame $t-1$ with N_{object} object queries (see $t = 0$ in Figure 2).

(ii) Tracking of objects from (i) and detection of new objects on frame t with all N queries.

The number of track queries N_{track} depends on the number of successfully detected objects in frame $t-1$. During training, the MLP predictions $\hat{y} = \{\hat{y}_j\}_{j=1}^N$ of the output embeddings from step (iv) are each assigned to one of the ground truth objects y or the background class. Each y_i represents a bounding box b_i, object class c_i and identity k_i.

Bipartite matching. The mapping $j = \pi(i)$ from ground truth objects y_i to the joint set of object and track query predictions $\hat{y}_j$ is determined either via track identity or costs based on bounding box similarity and object class. For the former, we denote the subset of ground truth track identities at frame t with $K_t \subset K$. Each detection from step (i) is assigned to its respective ground truth track identity k from the set $K_{t-1} \subset K$. The corresponding output embeddings, *i.e.*, track queries, inherently carry over the identity information to the next frame. The two ground truth track identity sets describe a hard assignment of the N_{track} track query outputs to the ground truth objects in frame t:

$K_t \cap K_{t-1}$: Match by track identity k.

$K_{t-1} \setminus K_t$: Match with background class.

$K_t \setminus K_{t-1}$: Match by minimum cost mapping.

The second set of ground truth track identities $K_{t-1} \setminus K_t$ includes tracks which either have been occluded or left the scene at frame t. The last set $K_{\text{object}} = K_t \setminus K_{t-1}$ of previously not yet tracked ground truth objects remains to be matched with the N_{object} object queries. To achieve this, we follow [7] and search for the injective minimum cost mapping $\hat{\sigma}$ in the following assignment problem,

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{k_i \in K_{\text{object}}} \mathcal{C}_{match}(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

with index $\sigma(i)$ and pair-wise costs $\mathcal{C}_{match}$ between ground truth y_i and prediction $\hat{y}_i$. The problem is solved with a combinatorial optimization algorithm as in [47]. Given the ground truth class labels c_i and predicted class probabilities $\hat{p}_i(c_i)$ for output embeddings i, the matching cost $\mathcal{C}_{match}$ with class weighting λ_{cls} is defined as

$$\mathcal{C}_{\text{match}} = -\lambda_{\text{cls}} \hat{p}_{\sigma(i)}(c_i) + \mathcal{C}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}). \tag{2}$$

The authors of [7] report better performance without logarithmic class probabilities. The $\mathcal{C}_{\text{box}}$ term penalizes bounding box differences by a combination of ℓ_1 distance and generalized intersection over union (IoU) [39] cost $\mathcal{C}_{\text{iou}}$,

$$\mathcal{C}_{\text{box}} = \lambda_{\ell_1} ||b_i - \hat{b}_{\sigma(i)}||_1 + \lambda_{\text{iou}} \mathcal{C}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}), \tag{3}$$

with weighting parameters $\lambda_{\ell_1}, \lambda_{\text{iou}}, \in \Re$. In contrast to ℓ_1, the scale-invariant IoU term provides similar relative errors for different box sizes. The optimal cost mapping $\hat{\sigma}$ determines the corresponding assignments in $\pi(i)$.

Set prediction loss. The final MOT set prediction loss is computed over all $N = N_{\text{object}} + N_{\text{track}}$ output predictions:

$$\mathcal{L}_{\text{MOT}}(y, \hat{y}, \pi) = \sum_{i=1}^{N} \mathcal{L}_{\text{query}}(y, \hat{y}_i, \pi). \tag{4}$$

The output embeddings which were not matched via track identity or $\hat{\sigma}$ are not part of the mapping π and will be assigned to the background class $c_i = 0$. We indicate the ground truth object matched with prediction i by $y_{\pi=i}$ and define the loss per query

$$\mathcal{L}_{\text{query}} = \begin{cases} -\lambda_{\text{cls}} \log \hat{p}_i(c_{\pi=i}) + \mathcal{L}_{\text{box}}(b_{\pi=i}, \hat{b}_i), & \text{if } i \in \pi \\ -\lambda_{\text{cls}} \log \hat{p}_i(0), & \text{if } i \notin \pi. \end{cases}$$

The bounding box loss $\mathcal{L}_{\text{box}}$ is computed in the same fashion as (3), but we differentiate its notation as the cost term $\mathcal{C}_{\text{box}}$ is generally not required to be differentiable.

Track augmentations. The two-step loss computation, see (i) and (ii), for training track queries represents only a limited range of possible tracking scenarios. Therefore, we propose the following augmentations to enrich the set of potential track queries during training. These augmentations will be verified in our experiments. We use three types of augmentations similar to [66] which lead to perturbations of object location and motion, missing detections, and simulated occlusions.

1. The frame $t-1$ for step (i) is sampled from a range of frames around frame t, thereby generating challenging frame pairs where the objects have moved substantially from their previous position. Such a sampling allows for the simulation of camera motion and low frame rates from usually benevolent sequences.
2. We sample false negatives with a probability of p_{FN} by removing track queries before proceeding with step (ii). The corresponding ground truth objects in frame t will be matched with object queries and trigger a new object detection. Keeping the ratio of false positives sufficiently high is vital for a joined training of both query types.
3. To improve the removal of tracks, *i.e.*, by background class assignment, in occlusion scenarios, we complement the set of track queries with additional false positives. These queries are sampled from output embeddings of frame $t-1$ that were classified as background. Each of the original track queries has a chance of p_{FP} to spawn an additional false positive query. We chose these with a large likelihood of occluding with the respective spawning track query.

Another common augmentation for improved robustness, is to applying spatial jittering to previous frame bounding boxes or center points [66]. The nature of track queries, which encode object information implicitly, does not allow for such an explicit perturbation in the spatial domain. We believe our randomization of the temporal range provides a more natural augmentation from video data.

4. Experiments

In this section, we present tracking results for TrackFormer on two MOTChallenge benchmarks, namely, MOT17 [29] and MOTS20 [51]. Furthermore, we verify individual contributions in an ablation study.

4.1. MOT benchmarks and metrics

Benchmarks. The MOT17 [29] benchmark consists of a train and test set, each with 7 sequences and pedestrians annotated with full-body bounding boxes. To evaluate the tracking (data association) robustness independently, three sets of public detections with varying quality are provided, namely, DPM [15], Faster R-CNN [38] and SDP [59].

MOTS20 [51] provides mask annotations for 4 train and test sequences of MOT17 but without annotations for small objects. The corresponding bounding boxes are not full-body, but based on the visible segmentation masks.

Metrics. Different aspects of MOT are evaluated by a number of individual metrics [5]. The community focuses on two compound metrics, namely, Multiple Object Tracking Accuracy (MOTA) and Identity F1 Score (IDF1) [40]. While the former focuses on object coverage, the identity preservation of a method is measured by the latter. For MOTS, we report MOTSA which evaluates predictions with a ground truth matching based on mask IoU.

Public detections. The MOT17 [29] benchmark is evaluated in a private and public detection setting. The latter allows for a comparison of tracking methods independent of the underlying object detection performance. MOT17 provides three sets of public detections with varying quality. In contrast to classic tracking-by-detection methods, TrackFormer is not able to directly produce tracking outputs from detection inputs. Therefore, we report the results of TrackFormer and CenterTrack [66] in Table 1 by filtering the initialization of tracks with a minimum IoU requirement. For more implementation details and a discussion on the fairness of such a filtering, we refer to the appendix.

4.2. Implementation details

TrackFormer follows the ResNet50 [17] CNN feature extraction and Transformer encoder-decoder architecture presented in Deformable DETR [68]. For track queries, the deformable reference points for the current frame are dynamically adjusted to the previous frame bounding box centers. Furthermore, for the decoder we stack the feature maps of the previous and current frame and compute cross-attention with queries over both frames. Queries are able to discriminate between features from the two frames by applying a temporal feature encoding as in [55]. For more detailed hyperparameters, we refer to the appendix.

Decoder Queries. By design, TrackFormer can only detect a maximum of N_{object} objects. To detect the maximum number of 52 objects per frame in MOT17 [29], we train TrackFormer with $N_{\text{object}} = 500$ learned object queries. For optimal performance, the total number of queries must exceed the number of ground truth objects per frame by a large margin. The number of possible track queries is adaptive and only practically limited by the abilities of the decoder.

Simulate MOT from single images. The encoder-decoder multi-level attention mechanism requires substantial amounts of training data. Hence, we follow a similar approach as in [66] and simulate MOT data from the CrowdHuman [44] person detection dataset. The adjacent training frames $t-1$ and t are generated by applying random spatial augmentations to a single image. To generate challenging tracking scenarios, we randomly resize and crop of up to 20% with respect to the original image size.

Training procedure. All trainings follow [68] and apply a batch size of 2 with initial learning rates of 0.0002 and 0.00002 for the encoder-decoder and backbone, respectively. For public detections, we initialize with the model weights from [68] pretrained on COCO [27] and then fine-tune on MOT17 for 50 epochs with a learning rate drop after 10 epochs. The private detections model is trained from scratch for 85 epochs on CrowdHuman [44] with simulated adjacent frames and we drop the initial learning rates after 50 epochs. To avoid overfitting to the small MOT17 dataset, we then fine-tune for additional 40 epochs on the combined CrowdHuman and MOT17 datasets. The fine-tuning starts with the initial learning rates which are dropped after 10 epochs. By the nature of track queries each sample has a different number of total queries $N = N_{\text{object}} + N_{\text{track}}$. In order to stack samples to a batch, we pad the samples with additional false positive queries. The training of the private detections model takes around 2 days on 7 × 32GB GPUs.

Mask training. TrackFormer predicts instance-level object masks with a segmentation head as in [7] by generating spatial attention maps from the encoded image features and decoder output embeddings. Subsequent upscaling and convolution operations yield mask predictions for all output embeddings. We adopt the private detection training pipeline from MOT17 but retrain TrackFormer with the original DETR [7] attention. This is due to the reduced memory consumption for single scale feature maps and inferior segmentation masks from sparse deformable attention maps. Furthermore, the benefits of deformable attention vanish on MOTS20 as it excludes small objects. After training on MOT17, we freeze the model and only train the segmentation head on all COCO images containing persons. Finally, we fine-tune the entire model on MOTS20.

	Method	Data	FPS ↑	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
				Public						
Offline	jCC [21]	–	–	51.2	54.5	493	872	25937	247822	1802
	FWT [18]	–	–	51.3	47.6	505	830	24101	247921	2648
	eHAF [45]	–	–	51.8	54.7	551	893	33212	236772	1834
	TT [63]	–	–	54.9	63.1	575	897	20236	233295	1088
	MPNTrack [6]	M+C	–	58.8	61.7	679	788	17413	213594	1185
	Lif_T [19]	M+C	–	60.5	65.6	637	791	14966	206619	1189
Online	FAMNet [10]	–	–	52.0	48.7	450	787	14138	253616	3072
	Tracktor++ [4]	M+C	1.3	56.3	55.1	498	831	**8866**	235449	1987
	GSM [28]	M+C	–	56.4	57.8	523	813	14379	230174	**1485**
	CenterTrack [66]	–	17.7	60.5	55.7	580	777	11599	208577	2540
	TMOH [46]	–	–	62.1	**62.8**	633	739	10951	201195	**1897**
	TrackFormer	–	7.4	**62.3**	57.6	**688**	**638**	16591	**192123**	4018
				Private						
Online	TubeTK [31]	JTA	–	63.0	58.6	735	468	27060	177483	4137
	GSDT [54]	6M	–	73.2	66.5	981	411	26397	120666	3891
	FairMOT [64]	CH+PD	–	73.7	72.3	1017	408	27507	117477	3303
	PermaTrack [49]	CH+PD	–	73.8	68.9	1032	405	28998	115104	3699
	GRTU [53]	CH+6M	–	75.5	76.9	1158	495	27813	108690	1572
	TLR [52]	CH+6M	–	76.5	73.6	1122	300	29808	99510	3369
	CTracker [35]	–	–	66.6	57.4	759	570	22284	160491	5529
	CenterTrack [66]	CH	17.7	67.8	64.7	816	579	**18498**	160332	3039
	QuasiDense [32]	–	–	68.7	66.3	957	516	26589	146643	3378
	TraDeS [56]	CH	–	69.1	63.9	858	507	20892	150060	3555
	TrackFormer	CH	7.4	**74.1**	**68.0**	**1113**	**246**	34602	**108777**	**2829**

Table 1. Comparison of multi-object tracking methods on the **MOT17** [29] test set. We report private as well as public detection results and separate between online and offline approaches. Both TrackFormer and CenterTrack filter tracks by requiring a minimum IoU with public detections. For a detailed discussion on the fairness of such a filtering, we refer to the appendix. We indicated additional training *Data*: CH=CrowdHuman [44], PD=Parallel Domain [49] (synthetic), 6M=6 tracking datasets as in [64], JTA [13] (synthetic), M=Market1501 [65] and C=CUHK03 [26]. Runtimes (FPS) are self-measured.

4.3. Benchmark results

MOT17. Following the training procedure described in Section 4.2, we evaluate TrackFormer on the MOT17 [29] test set and report results in Table 1.

First of all, we isolate the tracking performance and compare results in a public detection setting by applying a track initialization filtering similar to [66]. However to improve fairness, we filter not by bounding box center distance as in [66] but a minimum IoU as detailed in the appendix. TrackFormer performs on-par with state-of-the-art results in terms of MOTA without pretraining on CrowdHuman [44]. Our identity preservation performance is only surpassed by [46] and offline methods which benefit from the processing of entire sequences at once.

On private detections, we achieve a new state-of-the-art both in terms of MOTA (+5.0) and IDF1 (1.7) for methods only trained on CrowdHuman [44]. Only the methods [49, 52, 53] which follow [64] and pretrain on 6 additional tracking datasets (6M) surpass our performance. In contrast to our public detection model not only the detection but tracking performance are greatly improved. This is due to the additional tracking data provided by simulating adjacent frames on CrowdHuman which satisfies the large data requirements of Transformers.

Method	TbD	sMOTSA ↑	IDF1 ↑	FP ↓	FN ↓	ID Sw. ↓
		Train set (4-fold cross-validation)				
MHT_DAM [22]	×	48.0	–	–	–	–
FWT [18]	×	49.3	–	–	–	–
MOTDT [8]	×	47.8	–	–	–	–
jCC [21]	×	48.3	–	–	–	–
TrackRCNN [51]		52.7	–	–	–	–
MOTSNet [37]		56.8	–	–	–	–
PointTrack [57]		58.1	–	–	–	–
TrackFormer		**58.7**	–	–	–	–
		Test set				
Track R-CNN [51]		40.6	42.4	**1261**	12641	567
TrackFormer		**54.9**	**63.6**	2233	**7195**	**278**

Table 2. Comparison of multi-object tracking and segmentation methods evaluated on the **MOTS20** [51] train and test sets. Methods indicated with *TbD* first perform tracking-by-detection without segmentation on SDP [60] public detections and then predict apply a Mask R-CNN [16] fine-tuned on MOTS20.

Our tracking-by-attention approach achieves top performance via global attention between encoded input pixels and decoder queries without relying on additional motion [4, 10] or appearance models [4, 8, 10]. Furthermore, the frame to frame association with track queries avoids post-processing with heuristic greedy matching procedures [66] or additional graph optimization [28]. Our proposed TrackFormer represents the first application of Transformers to the MOT problem and could work as a blueprint for future research in this promising direction. In particular, we expect great potential for methods going beyond the two-frame training/inference regime.

MOTS20. In addition to object detection and tracking, TrackFormer is able to predict instance-level segmentation masks. As reported in Table 2, we achieve state-of-the-art MOTS results in terms of object coverage (MOTSA) and identity preservation (IDF1). All methods are evaluated in a private setting. A MOTS20 test set submission is only recently possible, hence we also provide the 4-fold cross-validation evaluation established in [51] and report the mean best epoch results over all splits. TrackFormer surpasses all previous methods without relying on a dedicated tracking formulation for segmentation masks as in [57]. In Figure 3, we present a qualitative comparison of TrackFormer and Track R-CNN [51] on two test sequences.

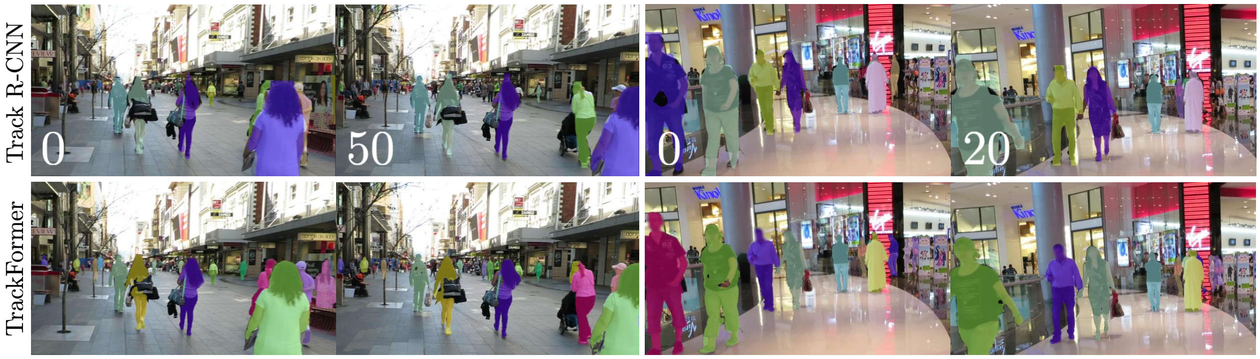

Figure 3. We compare **TrackFormer segmentation results** with the popular Track R-CNN [51] on selected MOTS20 [51] test sequences. The superiority of TrackFormer in terms of MOTSA in Table 2 can be clearly observed by the difference in pixel mask accuracy.

Method	MOTA ↑	Δ	IDF1 ↑	Δ
TrackFormer	71.3		73.4	
w\o				
Pretraining on CrowdHuman	69.3	-2.0	71.8	-1.6
Track query re-identification	69.2	-0.1	70.4	-1.4
Track augmentations (FP)	68.4	-0.8	70.0	-0.4
Track augmentations (Range)	64.0	-4.4	59.2	-10.8
Track queries	61.0	-3.0	45.1	-14.1

Table 3. **Ablation study** on TrackFormer components. We report MOT17 [29] training set private results on a 50-50 frame split. The last row without (w\o) all components is only trained for object detection and associates tracks via greedy matching as in [66].

4.4. Ablation study

The ablation study on the MOT17 and MOTS20 training sequences are evaluated in a private detection setting with a 50-50 frame and 4-fold cross-validation split, respectively.

TrackFormer components. We ablate the impact of different TrackFormer components on the tracking performance in Table 3. Our full pipeline including pretraining on the CrowdHuman dataset provides a MOTA and IDF1 of 71.3 and 73.4, respectively. The baseline without (w\o) pretraining reduces this by -2.0 and -1.6 points, an effect expected to even more severe for the generalization to test. The attention-based *track query re-identification* has a negligible effect on MOTA but improves IDF1 by 1.4 points.

If we further ablate our false positives (FP) and frame range *track augmentations*, we see another drop of -5.2 MOTA and -11.2 IDF1 points. Both augmentations provide the training which rich tracking scenarios and prevent an early overfitting. The false negative track augmentations are indispensable for a joint training of object and track queries, hence we refrain from ablating these.

Our baseline without any tracking components and *track queries* is only trained for object detection. Data association is performed via greedy center distance matching as in [66] resulting in a huge drop of -3.0 MOTA and -14.1 IDF1. This version represents previous post-processing and matching methods and demonstrates the benefit of jointly addressing track initialization, identity and trajectory forming in our unified TrackFormer formulation.

Method	Mask training	MOTA ↑	IDF1 ↑
TrackFormer	×	61.9	56.0
		61.9	54.8

Table 4. We demonstrate the **effect of jointly training for tracking and segmentation** on a 4-fold split on the MOTS20 [51] train set. We evaluate with regular MOT metrics, *i.e.*, matching to ground truth with bounding boxes instead of masks.

Mask information improves tracking. This ablation studies the synergies between segmentation and tracking training. Table 4 only evaluates bounding box tracking performance and shows a +1.2 IDF1 improvement when trained jointly with mask prediction. The additional mask information does not improve track coverage (MOTA) but resolves ambiguous occlusion scenarios during training.

5. Conclusion

We have presented a unified tracking-by-attention paradigm for detection and multi-object tracking with Transformers. As an example of said paradigm, our end-to-end trainable TrackFormer architecture applies autoregressive track query embeddings to follow objects over a sequence. We jointly tackle track initialization, identity and trajectory forming with a Transformer encoder-decoder architecture and not relying on additional matching, graph optimization or motion/appearance modeling. Our approach achieves state-of-the-art results for multi-object tracking as well as segmentation. We hope that this paradigm will foster future work in Transformers for multi-object tracking.

Acknowledgements: We are grateful for discussions with Jitendra Malik, Karttikeya Mangalam, and David Novotny.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[2] Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 2

[3] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011. 2

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *Int. Conf. Comput. Vis.*, 2019. 1, 2, 7

[5] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008. 6

[6] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 2, 7

[7] Nicolas Carion, F. Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 3, 5, 6

[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 *Int. Conf. Multimedia and Expo*, 2018. 1, 2, 7

[9] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. *Eur. Conf. Comput. Vis.*, 2010. 2

[10] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Int. Conf. Comput. Vis.*, 2019. 2, 7

[11] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845, 2017. 1, 2

[12] Patrick Dendorfer, Aljosa Osep, Anton Milan, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *Int. J. Comput. Vis.*, 2020. 1

[13] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*, 2018. 7

[14] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 2

[15] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009. 6

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2, 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2, 3, 6

[18] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 1, 2, 7

[19] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *Int. Conf. Mach. Learn.*, 2020. 2, 7

[20] Hao Jiang, Sidney S. Fels, and James J. Little. A linear programming approach for multiple object tracking. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2007. 2

[21] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 1, 2, 7

[22] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In *Int. Conf. Comput. Vis.*, 2015. 1, 2, 7

[23] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: siamese cnn for robust target association. *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2016. 1, 2

[24] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 2

[25] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. *Int. Conf. Comput. Vis. Workshops*, 2011. 1, 2

[26] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person reidentification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 7

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv:1405.0312*, 2014. 6

[28] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In *Int. Joint Conf. Art. Int.*, 2020. 1, 2, 7

[29] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 2, 6, 7, 8

[30] Aljoša Ošep, Wolfgang Mehner, Paul Voigtlaender, and Bastian Leibe. Track, then decide: Category-agnostic vision-based multi-object tracking. *IEEE Int. Conf. Rob. Aut.*, 2018. 2

[31] 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 (CVPR)*, June 2020. 7

[32] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021. 2, 7

[33] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 2

[34] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: modeling social behavior for multi-target tracking. *Int. Conf. Comput. Vis.*, 2009. 2

[35] 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 *Proceedings of the European Conference on Computer Vision*, 2020. 7

[36] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 2

[37] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 7

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inform. Process. Syst.*, 2015. 1, 6

[39] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 5

[40] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Eur. Conf. Comput. Vis. Workshops*, 2016. 6

[41] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[42] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory prediction. *Eur. Conf. Comput. Vis.*, 2016. 2

[43] Paul Scovanner and Marshall F. Tappen. Learning pedestrian dynamics from the real world. *Int. Conf. Comput. Vis.*, 2009. 2

[44] 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:1805.00123*, 2018. 6, 7

[45] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang. Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 2, 7

[46] Daniel Stadler and Jurgen Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10958–10967, June 2021. 7

[47] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016. 2, 5

[48] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[49] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Int. Conf. Comput. Vis.*, 2021. 2, 7

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017. 1, 2, 3, 4

[51] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 6, 7, 8

[52] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 7

[53] Shuai Wang, Hao Sheng, Yang Zhang, Yubin Wu, and Zhang Xiong. A general recurrent tracking framework without real data. In *Int. Conf. Comput. Vis.*, 2021. 7

[54] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *IEEE Int. Conf. Rob. Aut.*, May 2021. 2, 7

[55] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[56] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2, 7

[57] Zhenbo Xu, Wei Zhang, Xiao Tan, Wei Yang, Huan Huang, Shilei Wen, Errui Ding, and Liusheng Huang. Segment as points for efficient online multi-object tracking and segmentation. In *Eur. Conf. Comput. Vis.*, 2020. 2, 7

[58] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and where are you going? *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 2

[59] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale

dependent pooling and cascaded rejection classifiers. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 6

[60] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2129–2137, 2016. 7

[61] Qian Yu, Gerard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2007. 2

[62] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[63] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Lyu, W. Ke, and Z. Xiong. Long-term tracking with deep tracklet association. *IEEE Trans. Image Process.*, 2020. 2, 7

[64] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, pages 1–19, 2021. 7

[65] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Int. Conf. Comput. Vis.*, 2015. 7

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020. 1, 2, 5, 6, 7, 8

[67] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Eur. Conf. Comput. Vis.*, 2018. 1, 2

[68] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *Int. Conf. Learn. Represent.*, 2021. 2, 6