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 spatiotemporal trajectories. We formulate this task as a frame-to-frame set prediction problem and introduce TrackFormer, an end-to-end 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 new concept of identity preserving track queries. Both decoder query types benefit from self- and encoder-decoder attention on global frame-level features, thereby omitting any additional graph optimization and matching or modeling of motion and appearance. TrackFormer represents a new tracking-by-attention paradigm and yields 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 [47] 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. With progress in image-level object detectors [37, 7], most approaches follow the *tracking-by-detection* paradigm which consists of two-steps: (i) detecting objects in individual video frames, and (ii) associating sets of detections between frames, thereby creating individual object tracks over time. Traditional tracking-by-detection methods associate detections via temporally sparse [21, 24] or dense [20, 17] graph optimization, or apply convolutional neural networks to predict matching scores between detections [8, 22].

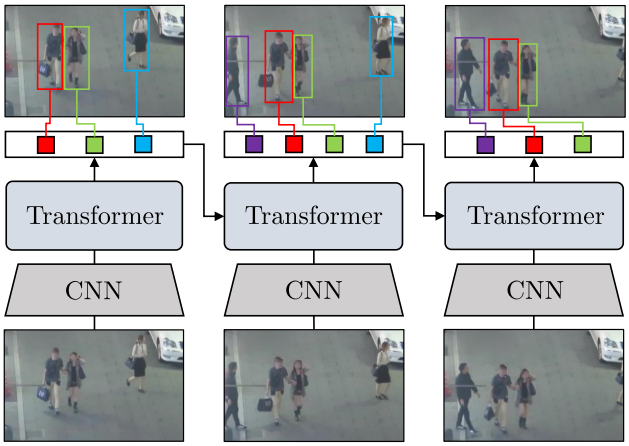

Figure 1. TrackFormer performs joint object detection and tracking by attention with Transformers. Object and autoregressive *track queries* reason about track initialization, identity, and occlusion of spatiotemporal trajectories.

Recent works [4, 6, 27] 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 achieve top performance only by relying either on additional graph optimization [6, 27] 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 a *tracking-by-attention* paradigm which formulates MOT as a set prediction problem. Our paradigm not only applies attention for data association [57, 11], but performs tracking and detection in a unified way. As shown in Figure 1, this is achieved by evolving a set of track predictions from frame to frame forming trajectories over time.

*Work done during an internship at Facebook AI Research.

We present TrackFormer, an end-to-end trainable Transformer [47] encoder-decoder architecture which encodes frame-level features from a convolutional neural network (CNN) [16] and decodes queries into bounding boxes associated with identities. The data association is performed through the nove 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, 58] and subsequently transform to future track queries. At each frame, the encoder-decoder processes the input image features, as well as the track and object queries, and outputs bounding boxes with assigned identities. Thereby, TrackFormer achieves detection and data association jointly via tracking-by-attention without relying on any additional track matching, graph optimization, or explicit modeling of motion and appearance. Our model is trained end-to-end and extends the recently proposed set prediction objective for object detection [45, 7, 58] to multi-object tracking.

We evaluate TrackFormer on the MOT17 [28] benchmark where it achieves state-of-the-art performance for public detections. Furthermore, we demonstrate the flexibility of our model with an additional mask prediction head and show state-of-the-art results on the Multi-Object Tracking and Segmentation (MOTS20) challenge [48].

In summary, we make the following contributions:

- An end-to-end multi-object tracking approach which achieves detection and data association in a 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.
- 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 to look beyond data association of given detections, 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 [19, 35, 54] or learned costs [23]. Other methods use association graphs [44], learned models [21], and motion information [20], general-purpose solvers [53], multicuts [46], weighted graph labeling [17], edge lifting [18], or trainable graph neural networks [6]. 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 [22], learned reID features [40], detection candidate selection [8] or affinity estimation [10]. Just like for re-identification, appearance models struggle in crowded scenarios with many object-object-occlusions.

Motion can be modelled for trajectory prediction [24, 1, 41] using a constant velocity assumption (CVA) [9, 2] or the social force model [42, 33, 50, 24]. Learning a motion model from data [23] can also accomplish track association between frames [55]. However, the projection of non-linear 3D motion 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 the new positions in the current frame. Previous efforts [13, 4] use regression heads on region-pooled object features. In [56], 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 [27] 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 from crowdedness and ambiguous background areas. Prior attempts have used category-agnostic image segmentation [30], applied Mask R-CNN [15] with 3D convolutions [48] and mask pooling layers [36], or represented objects as unordered point clouds [49]. However, the scarcity of annotated MOT segmentation data makes modern approaches still rely on bounding box predictions.

Attention for image recognition correlates each element of the input with respect to the others and is used in Transformers [47] for image generation [32] and object detection [7, 58]. For MOT, attention has only been used to associate a given set of object detections [57, 11], 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 as a post-processing matching step. It jointly reasons about track initialization, identity, and spatiotemporal trajectories. This allows us to refrain from any additional graph optimization, appearance or motion model by only relying on feature-level global attention.

3. TrackFormer

We present TrackFormer, an end-to-end multi-object tracking (MOT) approach based on an encoder-decoder Transformer [47] architecture. This section describes how we cast MOT as a set prediction problem and introduce the *tracking-by-attention* paradigm. We then introduce the concept of *track queries*, and how these are trained 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 tracking online 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 [16].

(ii) Encoding of frame features with self-attention in a Transformer encoder [47].

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

(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 current frame. The output embeddings accumulate bounding box and class information over multiple consecutive decoding layers. The permutation invariance of Transformers requires additive positional and object encodings for the frame features and decoder queries, respectively.

3.2. Tracking with decoder 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 Transformer decoding of object and track queries allows our model to perform detection and tracking in a unified way, and thereby introduces a new *tracking-by-attention* paradigm. A detailed architecture overview where we illustrate the integration of track and object queries into the Transformer decoder is shown in appendix A.3.

Track initialization. New objects that appear 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 decoding step. 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 current frame features and decoder queries *continuously updating* the instance-specific representation of object 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. TrackFormer thereby achieves implicit multi-frame attention over past frames.

In Figure 2, we provide a visual illustration of the track query concept. The initial detection in frame $t = 0$ spawns new track queries following their corresponding object to frame t and beyond. To this end, N_{object} object queries (white) are decoded to output embeddings for potential track initializations. Each successful object detection $\{\boldsymbol{b}^0_0, \boldsymbol{b}^1_0, \dots\}$ with a classification score above σ_{object}, *i.e.*, output embedding not predicting the background class (crossed), initializes a new track query embedding. As not all objects in a sequence might already appear on the first frame, the track identities $K_0 = \{0, 1, \dots\}$ only represent a subset of all K. For the decoding step at any frame $t > 0$, each track query initializes an additional output embedding associated to a different identity (colored). The joint set of $N_{\text{object}} + N_{\text{track}}$ output embeddings is initialized by (learned) object and (temporally adapted) track queries, respectively.

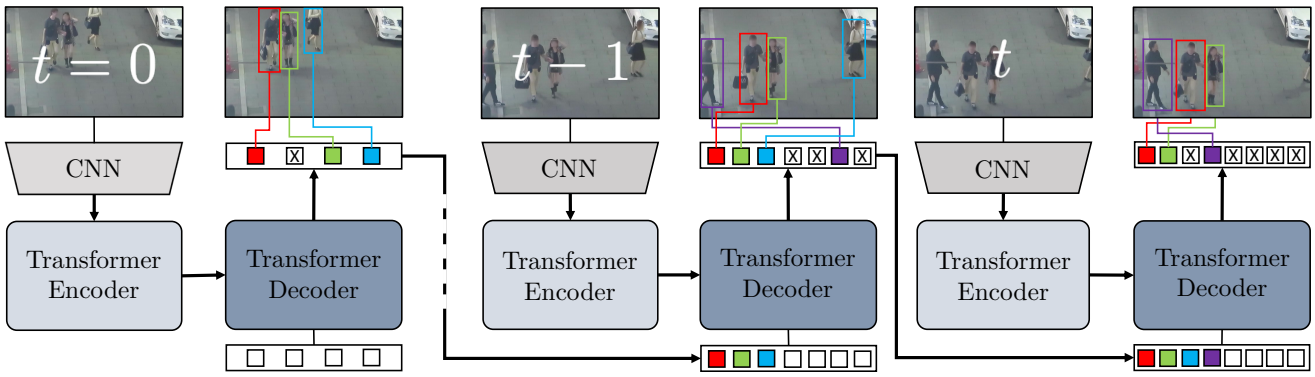

Figure 2. **TrackFormer** casts multi-object tracking as a set prediction problem performing joint detection and **tracking-by-attention**. The architecture consists of a CNN for image feature extraction, a Transformer [47] encoder for image feature encoding and a Transformer decoder which applies self- 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).

The Transformer decoder transforms the entire set of output embeddings at once and yields bounding box and class predictions for frame t. The number of track queries N_{track} changes between frames as new objects are detected or tracks are removed. Existing tracks followed by a track 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}. The application of a comparatively high σ_{NMS} only removes strongly overlapping duplicate bounding boxes which we found to be not 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 through short-term *occlusions*.

3.3. TrackFormer training

For track queries to follow objects to the next frame and work in interaction with object queries, TrackFormer requires dedicated frame-to-frame tracking training. This is accomplished by training on two adjacent frames, as indicated in Figure 2, and optimizing the entire MOT objective at once. The set prediction 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 prediction and bounding box similarity.

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 C_{match} between ground truth y_i and prediction $\hat{y}_i$. The problem is solved with a combinatorial optimization algorithm as in [45]. Given the ground truth class labels c_i and predicted class probabilities $\hat{p}_i(c_i)$ for output embeddings i, the matching cost C_{match} is defined as

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

In [7], a class cost term without logarithmic probabilities yielded better empirical performance. The $\mathcal{C}_{\text{box}}$ term penalizes bounding box differences by a linear combination of a ℓ_1 distance and a generalized intersection over union (IoU) [38] 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 $\lambda_{\ell_1}, \lambda_{\text{iou}}, \in \Re$. The scale-invariant IoU term provides similar relative errors for different box sizes and mitigates inconsistency of the ℓ_1 distance. 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

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

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 [56] 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 by assigning the background class 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 input bounding boxes or center points [56]. The nature of track queries, which encode spatial 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 [48]. Furthermore, we verify individual contributions in an ablation study.

4.1. MOT benchmarks and metrics

Datasets. 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 [14], Faster R-CNN [37] and SDP [51].

MOTS20 [48] provides mask annotations for 4 train and test sequences of MOT17. The corresponding bounding boxes are not full-body, but based on the visible segmentation masks, and only large objects are annotated.

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) [39]. 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 [28] 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 [56] 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 A.

4.2. Implementation details

TrackFormer follows the ResNet50 [16] CNN feature extraction and Transformer encoder-decoder architecture presented in Deformable DETR [58]. However, the Focal loss [25] applied in [58] emphasizes only the track queries, and ignores new object detections during training. Therefore, we resort to the original cross-entropy loss of DETR [7] as in Section 3.3. Deformable DETR [58] substantially reduces training time and improves detection performance for small objects which are very prominent in the MOT17 dataset. It [58] achieves this by replacing the original attention over single scale feature maps with multi-scale deformable attention modules. For track queries, the deformable reference points for the current frame t are dynamically adjusted to the previous frame bounding box centers.

Queries and the background class. By design, TrackFormer can only detect a maximum of N_{object} objects. To detect the maximum number of 52 objects per frame in MOT17 [28], we train TrackFormer with $N_{\text{object}} = 300$ learned object queries. The number of possible track queries is adaptive and only practically limited by the ability of the decoder to discriminate them. For optimal performance, the total number of queries must exceed the number of ground truth objects per frame by a large margin. To mitigate the resulting class imbalance, we follow [7] and downweigh the class prediction loss for background class queries by a factor of 0.1. To facilitate the training of track removal, we do not apply downweighting for false positive track augmentations.

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 [56] and simulate MOT data from the CrowdHuman [43] person detection dataset. The adjacent training frames $t-1$ and t are generated by applying random spatial augmentations to a single image. To simulate high frame rates as in MOT17 [28], we only randomly resize and crop of up to 5% with respect to the original image size.

Training procedure. We follow [58] and pretrain the model for 50 epochs without track queries on COCO [26]. The tracking capabilities are learned by training on MOT17 frame pairs or simulating adjacent MOT frames from single images. As in [58], the backbone and encoder-decoder are trained with individual learning rates of 0.00001 and 0.0001, respectively. The MOT17 public detections model is trained for a total of 40 epochs with a learning rate drop by a factor of 10 after the first 10 epochs. The private detections model is pretrained for 50 epochs on CrowdHuman and then fine-tuned on MOT17 with reduced learning rates for additional 20 epochs. Excluding the COCO pretraining, we train the public detections model for around 2 days on 7 16GB RTX 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. Since the two datasets have several sequences in common, we adopt the private detection training pipeline from MOT17 including the pretraining on CrowdHuman. However, for our MOTS20 model, we 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 beneficial effect of deformable attention vanishes on MOTS20 as it excludes small objects from segmentation. After training on MOT17, we freeze the model and train only the segmentation head on all COCO images containing persons. Finally, we fine-tune the entire model on the MOTS20 dataset.

	Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
	Private							
Online	TubeTK [31]	63.0	58.6	735	468	27060	177483	4137
Online	CTracker [34]	66.6	57.4	759	570	22284	160491	5529
Online	CenterTrack [56]	**67.8**	**64.7**	816	579	**18498**	160332	**3039**
Online	**TrackFormer**	65.0	63.9	**1074**	**324**	70443	**123552**	3528
	Public							
Offline	jCC [20]	51.2	54.5	493	872	25937	247822	1802
Offline	FWT [17]	51.3	47.6	505	830	24101	247921	2648
Offline	eHAF [44]	51.8	54.7	551	893	33212	236772	1834
Offline	TT [55]	54.9	63.1	575	897	20236	233295	1088
Offline	MPNTrack [6]	58.8	61.7	**679**	**788**	17413	213594	**1185**
Offline	Lif_T [18]	**60.5**	**65.6**	637	791	**14966**	**206619**	1189
Online	FAMNet [10]	52.0	48.7	450	787	14138	253616	3072
Online	Tracktor++ [4]	56.3	55.1	498	831	**8866**	235449	1987
Online	GSM [27]	56.4	57.8	523	813	14379	230174	**1485**
Online	CenterTrack [56]	60.5	55.7	580	777	11599	208577	2540
Online	**TrackFormer**	**62.5**	**60.7**	**702**	**632**	32828	**174921**	3917

Table 1. Comparison of modern multi-object tracking methods evaluated on the **MOT17** [28] test set. We report private as well as public detections results and separate between online and offline approaches. TrackFormer achieves state-of-the-art results in terms of MOTA among all public tracking methods. Both TrackFormer and CenterTrack filter track initializations by requiring a minimum IoU with public detections. For a detailed discussion on the fairness of such a filtering, we refer to B.1 and Table A.1.

Method	AP ↑	FP ↓	FN ↓
DPM [14]	0.61	42308	36557
FRCNN [37]	0.72	10081	25963
SDP [51]	0.81	7599	18865
CenterTrack [56]	0.77	7662	9900
TrackFormer	0.73	13178	15441

Table 2. Detection performance on the MOT17 test set. Both TrackFormer and CenterTrack were pretrained on CrowdHuman [43] and evaluated only for single frame detection.

4.3. Benchmark results

MOT17. Following the training procedure described in Section 4.2, we evaluate TrackFormer on the MOT17 [28] test set and report results in Table 1. For private detections, we achieve results comparable with modern state-of-the-art methods. This is due to the detection performance of [7, 58], and thereby TrackFormer, which still lacks behind modern object detectors.

As shown in Table 2, TrackFormer applied for single-frame detection of pedestrians, *i.e.*, many small objects with object-object occlusions, is merely on par with Faster R-CNN. Note, the superior CenterTrack detection performance, which only translates to 2.8 points higher MOTA compared to our method. This demonstrates TrackFormer as a powerful approach for tracking.

Method	TbD	sMOTSA ↑	IDF1 ↑	FP ↓	FN ↓	ID Sw. ↓
Train set (4-fold cross-validation)						
MHT_DAM [21]	×	48.0	–	–	–	–
FWT [17]	×	49.3	–	–	–	–
MOTDT [8]	×	47.8	–	–	–	–
jCC [20]	×	48.3	–	–	–	–
TrackRCNN [48]		52.7	–	–	–	–
MOTSNet [36]		56.8	–	–	–	–
PointTrack [49]		58.1	–	–	–	–
TrackFormer		**58.7**	–	–	–	–
Test set						
Track R-CNN [48]		40.6	42.4	**1261**	12641	567
TrackFormer		**54.9**	**63.6**	2233	**7195**	**278**

Table 3. Comparison of modern multi-object tracking and segmentation methods evaluated on the **MOTS20** [48] train and test sets. Methods indicated with *TbD* originally perform tracking-by-detection without segmentation. Hence, they are evaluated on SDP [52] public detections and predict masks with an additional Mask R-CNN [15] fine-tuned on MOTS20. TrackFormer achieves state-of-the-art results in terms of MOTSA and IDF1 on both sets.

To further isolate the tracking performance, we compare results in a public detection setting in the lower two sections of Table 1. In that case, TrackFormer achieves state-of-the-art results both in terms of MOTA and IDF1 for online methods without pretraining on CrowdHuman [43]. Our identity preservation performance is only surpassed by offline methods which benefit from the processing of entire sequences at once.

TrackFormer 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 any post-processing with heuristic greedy matching procedures [56] or additional graph optimization [27].

MOTS20. In addition to object detection and tracking, TrackFormer is able to predict instance-level segmentation masks. As reported in Table 3, 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 [48] 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 [49]. In Figure 3, we present a qualitative comparison of TrackFormer and Track R-CNN [48] on two test sequences.

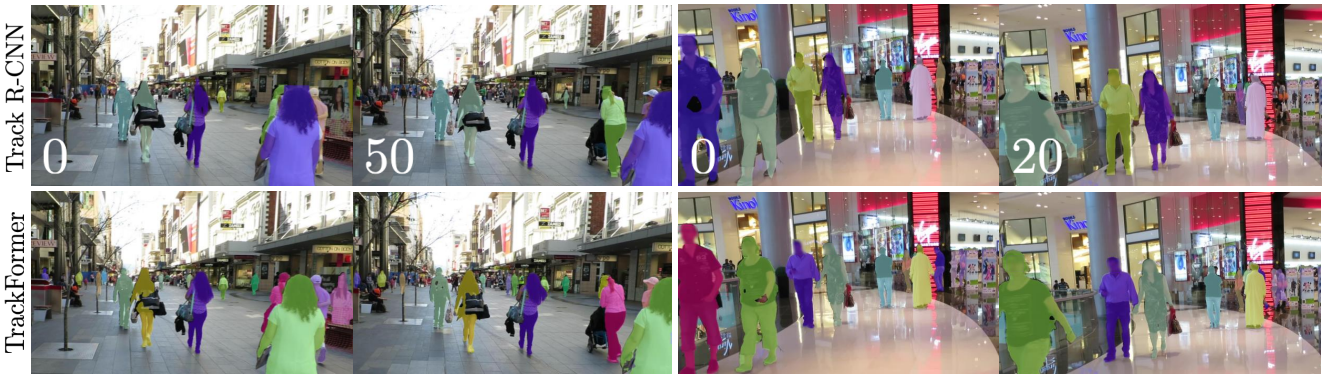

Figure 3. We compare TrackFormer segmentation results with the popular Track R-CNN [48] on selected MOTS20 [48] test sequences. The superiority of TrackFormer in terms of MOTSA in Table 3 can be clearly observed by the difference in pixel mask accuracy.

Method	MOTA ↑	Δ	IDF1 ↑	Δ
TrackFormer	51.4		55.3	
w\o				
Pretraining on CrowdHuman	42.8	-8.6	45.2	-10.1
Track query re-identification	42.7	-0.1	43.6	-1.6
Track augmentations (FP)	40.1	-2.6	42.9	-0.7
Track augmentations (Range)	38.1	-2.0	41.0	-1.9
Track queries	37.8	-0.3	27.4	-13.6

Table 4. Ablation study on individual TrackFormer components. We report mean best epoch results in a private setting on a 7-fold split on the MOT17 [28] training set. For the last row without (w\o) all components, we train only for object detection and associate tracks based on output embedding distance.

4.4. Ablation study

The ablation study on the MOT17 and MOTS20 training sequences are evaluated in a private detection setting with a 7- and 4-fold cross-validation split, respectively.

TrackFormer components. We ablate the impact of different TrackFormer components on the tracking performance in Table 4. Our full system including pretraining on the CrowdHuman dataset provides a MOTA and IDF1 of 51.4 and 55.3, respectively. The baseline without (w\o) pretraining reduces this by -8.6 and -10.1 points which demonstrates the limitations of the MOT17 dataset. The attention-based *track query re-identification* has a negligible effect on MOTA but improves IDF1 by 1.6 points.

If we further ablate our false positives (FP) and frame range *track augmentations*, we see another drop of -4.6 MOTA and -2.6 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 final baseline is without (w\o) any tracking components, not using *track queries* and is only trained for object detection. Data association is performed with a greedy center distance matching as in [56]. This leads to a dramatic drop of -13.6 in IDF1, as shown in the last row of Table 4.

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

Table 5. We demonstrate the effect of jointly training for tracking and segmentation on a 4-fold split on the MOTS20 [48] train set. We evaluate with regular MOT metrics, *i.e.*, matching to ground truth with bounding boxes instead of masks.

The version represents previous post-processing and matching methods and demonstrates the strength of jointly addressing track initialization, identity and trajectory forming in a unified TrackFormer configuration.

Mask information improves tracking. This final ablation is studying if segmentation mask prediction can improve tracking performance. Table 5 shows that a unified segmentation and tracking training procedure ican improve IDF1 by +1.2. In contrast to [7], we trained the entire model including the mask head and evaluate its bounding box tracking performance. The additional mask information did not improve track coverage (MOTA) but resolved ambiguous occlusion scenarios during training, thereby improving identity preservation (IDF1).

5. Conclusion

We have presented a unified tracking-by-attention paradigm for detection and multi-object tracking with Transformers. Our end-to-end TrackFormer architecture introduces track query embeddings which follow objects over a sequence in an autoregressive manner. An encoder-decoder architecture transforms each track query to the changing position of its corresponding object. TrackFormer jointly tackles track initialization, identity and spatiotemporal trajectory forming solely by attention operations and does not rely on any additional matching, graph optimization, motion or 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 detection and multi-object tracking.

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

Appendix

This section provides additional material for the main paper: §A contains further implementation details for TrackFormer (§A.1), a visualization of the Transformer encoder-decoder architecture (§A.3), and parameters for multi-object tracking (§A.4). §B contains a discussion related to public detection evaluation (§B.1), and detailed per-sequence results for MOT17 and MOTS20 (§B.2).

A. Implementation details

A.1. Backbone and training

We provide additional hyperparameters for TrackFormer. This supports our implementation details reported in Section 4.2 of the main paper. The Deformable DETR [58] encoder and decoder both apply 6 individual layers of feature-width 256. Each attention layer applies multi-headed self-attention [47] with 8 attention heads. We do not use the "DC5" (dilated conv_5) version of the backbone as this will incur a large memory requirement related to the larger resolution of the last residual stage. We expect that using "DC5" or any other heavier, or higher-resolution, backbone to provide better accuracy and leave this for future work.

Our training hyperparameters mostly follow the original DETR [7]. The weighting parameters of the individual box cost $\mathcal{C}_{\text{box}}$ and loss $\mathcal{L}_{\text{box}}$ are set to $\lambda_{\ell_1} = 5$ and $\lambda_{\text{iou}} = 2$. The probabilities for the track augmentation at training time are $p_{\text{FN}} = 0.4$ and $p_{\text{FP}} = 0.1$

A.2. Dataset splits

All experiments evaluated on dataset splits (ablation studies and MOTS20 training set in Table 3) apply the same training pipeline presented in Section 4.2 to each split. We average validation metrics over all splits and report the results from a single epoch (which yields the best mean MOTA / MOTSA) over all splits, *i.e.*, we do not take the best epoch for each individual split. For our ablation on the MOT17 [28] training set, we separate the 7 sequences into 7 splits each with a single sequence as validation set. Before training each of the 4 MOTS20 [48] splits, we pre-train the model on all MOT17 sequences excluding the corresponding split of the validation sequence.

A.3. Transformer encoder-decoder architecture

To foster the understanding of TrackFormer's integration of track queries within the decoder self-attention block, we provide a simplified visualization of the encoder-decoder architecture in Figure A.1. In comparison to the original

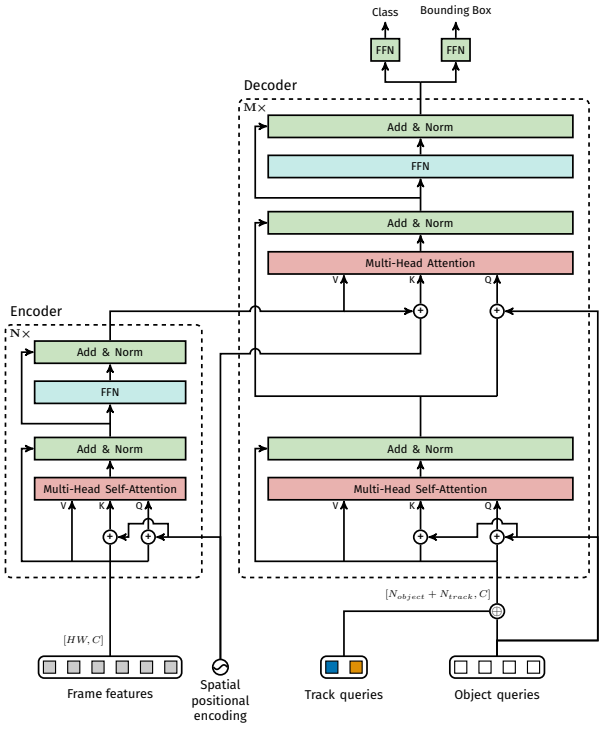

Figure A.1. The TrackFormer encoder-decoder architecture. We indicate the tensor dimensions in squared brackets.

illustration in [7], we indicate *track identities* instead of spatial encoding with *color-coded* queries. The frame features (indicated in grey) are the final output of the CNN feature extractor and have the same number of channels as both query types. The entire Transformer architecture applies N and M independently supervised encoder and decoder layers, with spatial positional and object encoding as in [7]. Track queries are fed *autoregressively* from the *previous frame* output embeddings of the last decoding layer (before the final feed-forward class and bounding box networks (FFN)). The object encoding is achieved by re-adding the object queries to the corresponding embeddings in the decoder key (K) and query (Q).

A.4. Multi-object tracking parameters

In Section 3.2, we explain the process of track initialization and removal over a sequence. The corresponding hyperparameters were optimized by a grid search on the MOT17 training set cross-validation splits. The grid search yielded track initialization and removal thresholds of $\sigma_{\text{detection}} = 0.9$ and $\sigma_{\text{track}} = 0.8$, respectively. The lower σ_{track} score prevents tracks from being removed too early and improves identity preservation performance. TrackFormer benefits from an NMS operation for the removal of strong occlusion cases with an intersection over union larger than $\sigma_{\text{NMS}} = 0.9$.

Method	IN	IoU	CD	MOTA ↑	IDF1 ↑
Offline					
MHT_DAM [21]	×			50.7	47.2
jCC [20]	×			51.2	54.5
FWT [17]	×			51.3	47.6
eHAF [44]	×			51.8	54.7
TT [55]	×			54.9	63.1
MPNTrack [6]	×			58.8	61.7
Lif_T [18]	×			60.5	65.6
Online					
MOTDT [8]	×			50.9	52.7
FAMNet [10]	×			52.0	48.7
Tracktor++ [4]	×			56.3	55.1
GSM_Tracktor [27]	×			56.4	57.8
CenterTrack [56]		×		60.5	55.7
TrackFormer		×		62.5	60.7
CenterTrack [56]			×	61.5	59.6
TrackFormer			×	63.4	60.0

Table A.1. Comparison of modern multi-object tracking methods evaluated on the **MOT17** [28] test set for different **public detection processing**. Public detections are either directly processed as input (IN) or applied for filtering of track initializations by center distance (CD) or intersection over union (IoU). We report mean results over the three sets of public detections provided by [28] and separate between online and offline approaches. The arrows indicate low or high optimal metric values.

For the track query re-identification, our search proposed an optimal inactive patience and score of $T_{\text{track-reid}} = 5$ and $\sigma_{\text{track-reid}} = 0.8$, respectively.

B. Experiments

B.1. Public detections and track filtering

TrackFormer implements a new tracking-by-attention paradigm which requires track initializations to be filtered for an evaluation with public detections. Here, we provide a discussion on the comparability of TrackFormer with earlier methods and different filtering schemes.

Common tracking-by-detection methods directly process the MOT17 public detections and report their mean tracking performance over all three sets. This is only possible for methods that perform data association on a bounding box level. However, TrackFormer and point-based methods such as CenterTrack [56] require a procedure for filtering track initializations by public detections in a comparable manner. Unfortunately, MOT17 does not provide a standardized protocol for such a filtering. The authors of CenterTrack [56] filter detections based on bounding box center distances (CD). Each public detection can possibly initialize a single track but only if its center point falls in the bounding box area of the corresponding track.

Sequence	sMOTSA ↑	IDF1 ↑	MOTSA ↑	FP ↓	FN ↓	ID Sw. ↓
MOTS20-01	59.8	68.0	79.6	255	364	16
MOTS20-06	63.9	65.1	78.7	595	1335	158
MOTS20-07	43.2	53.6	58.5	834	4433	75
MOTS20-12	62.0	76.8	74.6	549	1063	29
ALL	54.9	63.6	69.9	2233	7195	278

Table A.2. We present TrackFormer tracking and segmentation results on each individual sequence of the **MOTS20** [48] test set. MOTS20 is evaluated in a private detections setting. The arrows indicate low or high optimal metric values.

In Table A.1, we revisit our MOT17 test set results but with this public detections center distance (CD) filtering, while also inspecting the CenterTrack per-sequence results in Table A.3. We observe that this filtering does not reflect the quality differences in each set of public detections, *i.e.*, DPM [14] and SDP [51] results are expected to be the worst and best, respectively, but their difference is small.

We hypothesize that a center distance filtering is not in accordance with the common public detection setting and propose a filtering based on Intersection over Union (IoU). For IoU filtering, public detections only initialize a track if they have an IoU larger than 0.5. The results in Table A.1, show that for TrackFormer and CenterTrack, using IoU filtering performs worse compared to the CD filtering which is expected as this is a more challenging evaluation protocol. We believe IoU-based filtering (instead of CD-based) provides a fairer comparison to previous MOT methods which directly process public detections as inputs (IN). This is validated by the per-sequence results in Table A.4, where IoU filtering shows differences across detectors that are more meaningfully correlated with detector performance, compared to the relatively uniform performance across detections with the CD based method in Table A.3 (where DPM, FRCNN and SDP show *very similar* performance).

Consequently, we follow the IoU-based filtering protocol to compare with CenterTrack in our main paper. While our gain over CenterTrack seems similar across the two filtering techniques for MOTA (see Table A.1), the gain in IDF1 is significantly larger under the more challenging IoU-based protocol, which suggests that CenterTrack benefits from the less challenging CD-based filtering protocol, while TrackFormer does not rely on the filtering for achieving its high IDF1 tracking accuracy.

B.2. MOT17 and MOTS20 sequence results

In Table A.4, we provide per-sequence MOT17 [28] test set results for public detection filtering via Intersection over Union (IoU). Futhermore, we present per-sequence TrackFormer results on the MOTS20 [48] test set in Table A.2.

Sequence	Public detection	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
MOT17-01	DPM [14]	41.6	44.2	5	8	496	3252	22
MOT17-03	DPM	79.3	71.6	94	8	1142	20297	191
MOT17-06	DPM	54.8	42.0	54	63	314	4839	175
MOT17-07	DPM	44.8	42.0	11	16	1322	7851	147
MOT17-08	DPM	26.5	32.2	11	37	378	15066	88
MOT17-12	DPM	46.1	53.1	16	45	207	4434	30
MOT17-14	DPM	31.6	36.6	13	78	636	11812	196
MOT17-01	FRCNN [37]	41.0	42.1	6	9	571	3207	25
MOT17-03	FRCNN	79.6	72.7	93	7	1234	19945	180
MOT17-06	FRCNN	55.6	42.9	57	59	363	4676	190
MOT17-07	FRCNN	45.5	41.5	13	15	1263	7785	156
MOT17-08	FRCNN	26.5	31.9	11	36	332	15113	89
MOT17-12	FRCNN	46.1	52.6	15	45	197	4443	30
MOT17-14	FRCNN	31.6	37.6	13	77	780	11653	202
MOT17-01	SDP [51]	41.8	44.3	7	8	612	3112	27
MOT17-03	SDP	80.0	72.0	93	8	1223	19530	181
MOT17-06	SDP	55.5	43.8	56	61	354	4712	181
MOT17-07	SDP	45.2	42.4	13	15	1332	7775	147
MOT17-08	SDP	26.6	32.3	11	36	350	15067	91
MOT17-12	SDP	46.0	53.0	16	45	221	4426	30
MOT17-14	SDP	31.7	37.1	13	76	749	11677	205
	All	61.5	59.6	621	752	14076	200672	2583

Table A.3. We report the original per-sequence **CenterTrack** [56] MOT17 [28] test set results with **Center Distance (CD)** public detection filtering. The results do not reflect the varying object detection performance of DPM, FRCNN and SDP, respectively. The arrows indicate low or high optimal metric values.

Sequence	Public detection	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
MOT17-01	DPM [14]	48.2	38.4	5	8	266	3012	60
MOT17-03	DPM	73.8	70.4	90	14	6102	21083	236
MOT17-06	DPM	55.6	54.6	58	76	499	4533	195
MOT17-07	DPM	53.6	46.3	11	17	686	7047	112
MOT17-08	DPM	35.0	34.9	12	28	544	13024	164
MOT17-12	DPM	49.9	57.6	21	36	508	3789	44
MOT17-14	DPM	39.2	42.4	19	57	947	9958	338
MOT17-01	FRCNN [37]	49.5	40.7	8	7	363	2831	66
MOT17-03	FRCNN	75.5	71.4	91	12	6490	18907	243
MOT17-06	FRCNN	59.0	56.7	64	50	644	3962	224
MOT17-07	FRCNN	52.8	45.2	11	16	867	6980	131
MOT17-08	FRCNN	34.2	35.0	13	30	552	13201	142
MOT17-12	FRCNN	48.0	56.5	18	38	532	3932	40
MOT17-14	FRCNN	38.8	42.9	20	50	1596	9238	485
MOT17-01	SDP [51]	55.7	43.0	8	5	391	2396	69
MOT17-03	SDP	77.5	71.3	103	12	7159	16063	302
MOT17-06	SDP	58.5	56.5	78	58	724	3950	214
MOT17-07	SDP	55.8	46.1	14	14	958	6370	141
MOT17-08	SDP	36.4	35.6	15	26	720	12525	193
MOT17-12	SDP	50.7	59.8	21	31	666	3559	44
MOT17-14	SDP	42.4	42.8	22	47	1614	8561	474
	All	62.5	60.7	702	632	32828	174921	3917

Table A.4. We report **TrackFormer** results on each individual sequence and set of public detections evaluated on the MOT17 [28] test set. We apply our minimum **Intersection over Union (IoU)** public detection filtering. The arrows indicate low or high optimal metric values.

Evaluation metrics In Section 4.1 we explained two compound metrics for the evaluation of MOT results, namely, Multi-Object Tracking Accuracy (MOTA) and Identity F1 score (IDF1). [5] However, the MOTChallenge benchmark implements all CLEAR MOT [5] evaluation metrics. In addition to MOTA and IDF1, we report the following additional CLEAR MOT metrics:

MT:	Ground truth tracks covered for at least 80%.
ML:	Ground truth tracks covered for at most 20%.
FP:	False positive bounding boxes not corresponding to any ground truth.
FN:	False negative ground truth boxes not covered by any bounding box.
ID Sw.:	Bounding box switching the corresponding ground truth identity. in the previous frame.
sMOTSA:	Mask-based Multi-Object Tracking Accuracy (MOTA) which counts true positives instead of only masks with IoU larger than 0.5.

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, 10

[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, 12

[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, 10

[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, 7, 8, 9

[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, 10

[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, 10

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

[14] 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, 7, 10, 11

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

[16] 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

[17] 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, 10

[18] 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, 10

[19] 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

[20] 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, 10

[21] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In *Int. Conf. Comput. Vis.*, 2015. 1, 2, 7, 10

[22] 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

[23] 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

[24] 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

[25] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Int. Conf. Comput. Vis.*, pages 2999–3007, 2017. 6

[26] 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

[27] 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, 10

[28] 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, 9, 10, 11

[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. 5

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

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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, 7, 11

[38] 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

[39] 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

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

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

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

[43] 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

[44] 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, 10

[45] 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

[46] 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

[47] 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, 9

[48] 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, 5, 6, 7, 8, 9, 10

[49] 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

[50] 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

[51] 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, 7, 10, 11

[52] 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

[53] 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

[54] 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

[55] 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, 10

[56] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *ECCV*, 2020. 2, 5, 6, 7, 8, 10, 11

[57] 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

[58] 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, 7, 9