MeMOT: Multi-Object Tracking with Memory

Jiarui Cai[1*] Mingze Xu[2†] Wei Li[2] Yuanjun Xiong[2] Wei Xia[2] Zhuowen Tu[2] Stefano Soatto[2]
[1]University of Washington [2]AWS AI Labs
jrcai@uw.edu, {xumingze,wayl,yuanjx,wxia,ztu,soattos}@amazon.com

Abstract

We propose an online tracking algorithm that performs the object detection and data association under a common framework, capable of linking objects after a long time span. This is realized by preserving a large spatio-temporal memory to store the identity embeddings of the tracked objects, and by adaptively referencing and aggregating useful information from the memory as needed. Our model, called MeMOT, consists of three main modules that are all Transformer-based: 1) Hypothesis Generation that produce object proposals in the current video frame; 2) Memory Encoding that extracts the core information from the memory for each tracked object; and 3) Memory Decoding that solves the object detection and data association tasks simultaneously for multi-object tracking. When evaluated on widely adopted MOT benchmark datasets, MeMOT observes very competitive performance.

1. Introduction

Online multi-object tracking (MOT) [3, 13, 57, 70] aims at localizing a set of objects (*e.g.*, pedestrians), while following their trajectories over time so that the same object bears the same identities in the entire input video stream. Earlier methods mostly solved this problem with two separate stages: 1) the object detection stage that detects object instances in individual frames [14, 17, 28, 42, 72]; and 2) the data association stage that links the detected object instances across time [5, 70] by modeling the state changes of tracked objects and solving a matching problem between them and the detection results. Though recent studies [34, 69] suggest that combining these two stages could be beneficial, the combination usually leads to the undesired simplification of the association module in modeling the change of the objects in time.

In this paper, we propose a Transformer-based tracking model, called *MeMOT*, that performs object detection and association under a common framework in an online manner. The key design of MeMOT is to build a large spatio-temporal memory that stores the past observations of the tracked objects. The memory is actively encoded in every time step by referencing relevant information so that the states of the objects are more accurately approximated for the association task. The rich representation of the tracked objects extracted from the spatial-temporal memory enables us to solve the object detection and association tasks in a unified decoding module. It directly outputs object instances that have been tracked and reappears in the latest frame and novel object instances that are first time seen. The idea of MeMOT is illustrated in Fig. 1.

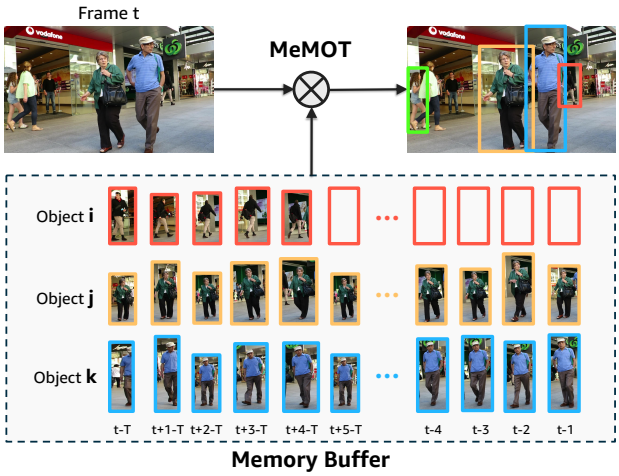

Figure 1. **Illustration of the idea of MeMOT**. A spatio-temporal memory stores a long range states of all tracked objects and is updated over time. Each row in the memory buffer represents an active tracklet. The "person crops" indicate that their the history states are preserved in the memory, and the blank box indicates this person does not appear in the frame at that time, occluded or not detected. The tracking plots show that MeMOT can maintain active tracks (yellow and blue boxes), link reappearing tracks after occlusion (red box), and generate new objects (green box).

At each time step, MeMOT runs the following three main components: 1) a *hypothesis generation* module that produces object proposals from input image feature maps as

*The work was done during an Amazon internship.
†Corresponding Author.

a set of embedding vectors; 2) a *memory encoding* module that encodes the spatial-temporal memory corresponding to each tracked object into a vector known as the track embedding; and 3) a *memory decoding* that inputs the proposal and track embeddings and solves the object detection and data association tasks simultaneously for multi-object tracking. The hypothesis generation module is implemented by a Transformer-based encoder-decoder network [6, 73]. It produces a set of embedding vectors, known as the *proposal embedding*, each representing one hypothetical object instance. The memory encoding module first divides the spatial temporal memory on each object into short- and long-term memories and aggregates them each into one embedding vector through cross attention modules [50]. The two vectors then interact through the self-attention mechanism to produce the *track embedding* of the tracked object at this time step. The proposal and track embeddings, together with the original image features, are then fed to the memory decoding module. For each track embedding, it produces the location and the visibility of the object being tracked in this frame. For each proposal embedding, it predicts whether this hypothetical object instance is depicting a novel object, a tracked object, or simply a background region. An illustration of the MeMOT model is shown in Fig. 2. The entire model can be trained end to end on video datasets with object bounding box and identity annotations. During inference, we obtain the tracking outputs in one inference run of the model at each time step, *without* any extra optimization [9, 41] or post-processing [3, 48, 70].

We evaluate MeMOT on the MOT Challenge [10, 35] benchmarks for pedestrian tracking. Experimental results show that MeMOT achieves the state-of-the-art performance among all algorithms with an in-network association solver and is competitive with those utilizing a post-network association process. Specifically, MeMOT outperforms other Transformer-based methods in both object detection and data association. Extensive ablation studies further validate the design and effectiveness of MeMOT.

2. Related Work

Classical Tracking Methods. Tracking is well studied in computer vision [2, 23, 24, 61]. Coping with the underlying uncertainties of the tracking results [23] and object appearances/positions/shapes [2] has been a central challenge. Classical non-deep learning methods [61] have laid out solid mathematical and statistical foundations. Specifically, Kalman [55] and particle filters [20] are widely adopted for tackling tracking problems [22,46,62]. The progressive observation-based Bayesian inference method [63] is proposed for MOT in online sports videos. A spatial and temporal shape representation-based Bayesian framework [18] is proposed for multi-cue 3D deformable object tracking. In these methods, an optimal filter maintains tracking states that summarize history information and estimate new frame's tracking results. In a linear-Gaussian case, the optimal state can indeed be estimated, while for more general non-linear, non-Gaussian cases, it is difficult to estimate the optimal state with a finite-dimensional state representation. For instance, occlusion in visual multiple object tracking is clearly non-linear and non-Gaussian. To tackle this challenge, tracking methods [7, 40] that can access multiple frames states (offline tracking) is desired.

MOT with CNNs. A typical scheme for MOT [8,15,51,57] is "tracking-by-detection", which directly uses ready-made detectors [14, 17, 28, 42, 72] and focuses on the data association. Tracktor++ [3] propagates the bounding boxes of tracked objects as region proposals to the next frame. CenterTrack [71] takes an additional point-based heatmap as input and matches objects anywhere within the receptive field. JDE [26, 54, 65, 70] is built with two homogeneous branches for object detection and ReID feature extraction, respectively. Joint detection and tracking models improve the runtime, but sacrifice the tracking recovery after occlusion and cannot reconnect long-term missing objects.

MOT with Transformers. Vision Transformers have been successfully applied in image recognition [6, 11, 29, 73] and video analysis [1, 4, 30, 45] lately. In tracking, TrackFormer [34] and MOTR [69] simultaneously perform the object detection and association by concatenating the object and autoregressive track queries as inputs to the Transformer decoder in the next time step. On the other hand, TransCenter [67] and TransTrack [48] only use Transformers as feature extractor and recurrently pass track features to aggressively learn aggregated embedding of each object. TransMOT [9] still uses CNNs as detector and feature extractor, and learns an affinity matrix with Transformers. The above work explores the mechanism of representing object states as dynamic embeddings. However, the modeling of long-term spatio-temporal observations and adaptive feature aggregation methods are underdeveloped.

Memory Networks. Pioneering work using memory networks has been proposed in NLP [19, 47, 56] by focusing on temporal reasoning tasks such as question answering [25, 64] and dialogue systems [58]. Video analysis tasks, such as action recognition [59, 66], and video object segmentation [32, 36], leverage an external memory to store and access time-indexed features in prolonged sequences, significantly improving the ability to remember the past. Recently, memory networks have been introduced into tracking. MemTrack [68] reads a residual template from memory and combines it with the initial template to update the representations of targets. STMTrack [16] guides the information retrieval with the current frame and adaptively obtains all useful information as it needs. However, these work focuses on single object tracking (SOT), and does not need to concern about the inter-object associa-

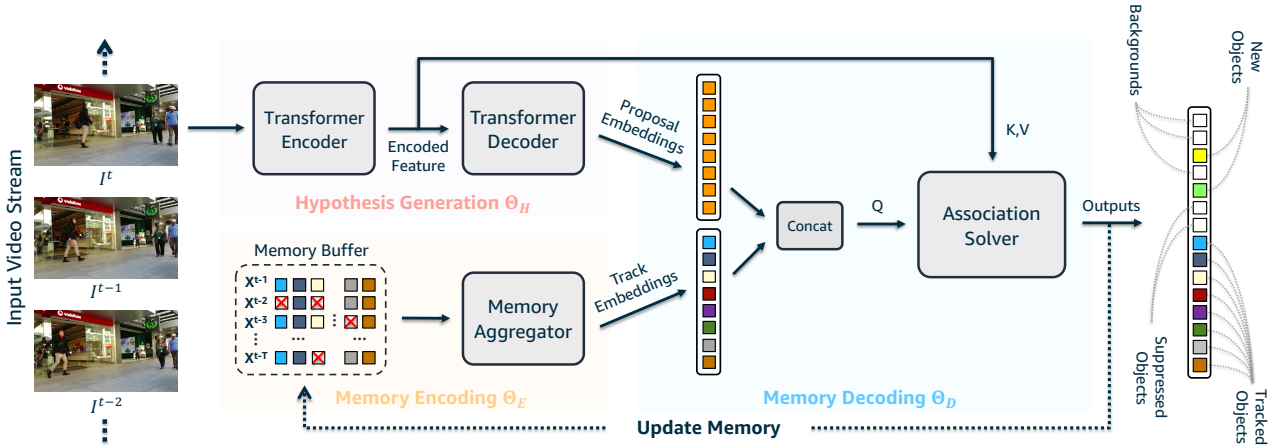

Figure 2. **Visualization of MeMOT**, which runs three main components: 1) a hypothesis generation module Θ_H that produces object proposals for the current video frame, 2) a memory encoding module Θ_E that retrieves core information for each tracked objects, and 3) a memory decoding module Θ_D that solves the object detection and data association tasks simultaneously. MeMOT maintains a memory buffer to store long-range states of tracked objects, together with an efficient encoding-decoding process that retrieves useful information for linking objects after a long time span. Each hypothetical object is predicted as a new object, a tracked object, or a background region.

tion. We propose to use a large spatio-temporal memory to achieve robust object association across time for MOT.

3. Multi-Object Tracking with Memory

3.1. Overview

Given a sequence of video frames $\boldsymbol{I} = \{I^0, I^1, \cdots, I^T\}$, the goal of online MOT is to localize a set of K objects while following their trajectories $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_K\}$ over time by causal processing. In this paper, we propose an end-to-end tracking algorithm, called *MeMOT*, which jointly learns the object detection and association. Different from most existing methods [3] that propagate the states of tracked objects between adjacent frames, we build a *spatio-temporal memory* that stores long-range states of all tracked objects, together with a *memory encoding-decoding* process that efficiently retrieves useful information for linking objects after a long time span.

Specifically, as shown in Figure 2, MeMOT consists of three main components: 1) a frame-level hypothesis generation module Θ_H that produces region proposals for the current video frame I^t, 2) a track-level memory encoding module Θ_E that aggregates track embeddings, and 3) a memory decoding module Θ_D that associates the new detections with tracked objects. At time step t, Θ_H generates N^t_{pro} region proposals, represented as proposal embeddings $\boldsymbol{Q}^t_{pro} \in \mathbb{R}^{N^t_{pro} \times d}$ using a Transformer-based architecture. The memory encoder Θ_E adaptively translates the "history states" of each track into one compact representation, denoted as track embeddings $\boldsymbol{Q}^t_{tck} \in \mathbb{R}^{N^t_{tck} \times d}$. By querying the encoded image feature with $[\boldsymbol{Q}^t_{pro}, \boldsymbol{Q}^t_{tck}]$, the memory decoder Θ_D computes the inter-object relations and updates the embeddings as $[\widehat{\boldsymbol{Q}}^t_{pro}, \widehat{\boldsymbol{Q}}^t_{tck}]$ accordingly. Then the locations $[\boldsymbol{B}^t_{pro}, \boldsymbol{B}^t_{tck}]$ and confidence scores $[\boldsymbol{S}^t_{pro}, \boldsymbol{S}^t_{tck}]$ of new and tracked objects are predicted based on these output embeddings. Finally, the locations and states of the previously tracked objects are used to update their trajectory and the memory. The "new-born" objects are initialized in $\boldsymbol{\mathcal{T}}$ and their states are added into the memory.

3.2. Hypothesis Generation

The hypothesis generation network Θ_H is built with an encoder-decoder architecture based on Transformers [6,73]. It produces a set of N^t_{pro} region proposals that either initiate "new-born" objects for the current video frame or update tracked objects with their latest identity and position information. The Θ_H encoder takes a sequentialized feature map $z^t_0 \in \mathbb{R}^{C \times HW}$ as inputs, which is extracted by a CNN backbone from the input frame I^t. Each element in z^t_0 is supplemented with a unique positional encoding that indicates its spatial location. The image feature is encoded as $z^t_1 \in \mathbb{R}^{d \times HW}$ using a multi-layer Transformer encoder. The Θ_H decoder receives the encoded feature z^t_1 and empty object queries (represented as learnable embeddings), and produces the final set of proposal embeddings $\boldsymbol{Q}^t_{pro} \in \mathbb{R}^{N^t_{pro} \times d}$. The objectness scores and bounding boxes of each proposal can be predicted from $\boldsymbol{Q}^t_{pro}$.

3.3. Spatio-Temporal Memory

We store the history states of all N tracked objects in a spatio-temporal memory buffer $\boldsymbol{X} \in \mathbb{R}^{N \times T \times d}$. It reserves at most N_{max} objects and a maximum of T_{max} time steps for each object. The memory is implemented with a first-in-first-out (FIFO) data structure. At time step t, the mem-

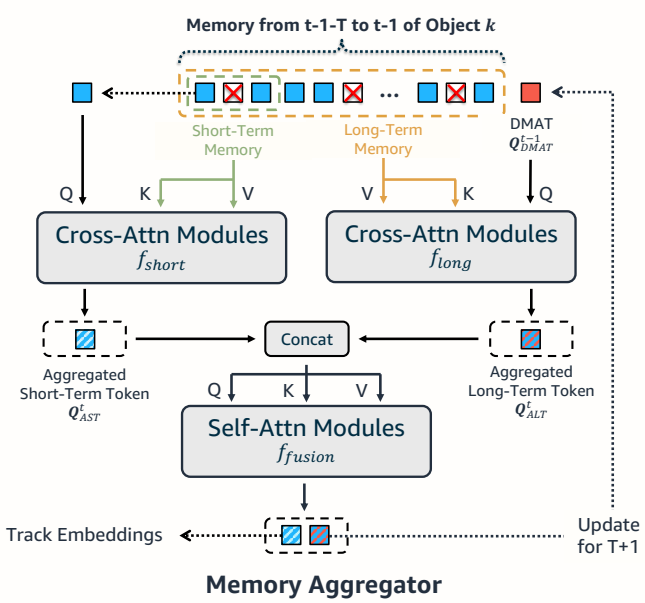

Figure 3. **Illustration of Memory Aggregator**, which consists three attention modules: 1) short-term f_{short} that smoothes out noises in recent frames, 2) long-term f_{long} that extracts supportive features from long-range context, and 3) fusion blocks that aggregate short- and long-term branches. The aggregated embeddings will be used as the track embeddings (blue-white query) and update DMAT (blue-red query) for the next time step.

ory is represented as the states of N_{tck}^{t-1} active objects in the past T frames, $\boldsymbol{X}^{t-1-T:t-1} = \{x_k^{t-1-T:t-1}\}_{k=1:N_{tck}^{t-1}}$, where $x_k^{t-1-T:t-1}$ is the states of the k-th object and is padded with $\boldsymbol{0}$ if this object does not appear in the frame I^t. When T is larger than T_{max}, the "oldest" state x_k^{t-1-T} of each tracklet graduates from the memory. N_{max} is set to be significantly large (*e.g.*, 300 or 600) to cover the typical number of objects in a video, and a choice of T_{max} is 24.

3.4. Memory Encoding

As shown in Fig. 3, we encode the memory and extract the track embedding with three attention modules: 1) a short-term block f_{short} for assembling embeddings of neighboring frames to smooth out the noises, 2) a long-term block f_{long} for extracting relevant features in the temporal window covered by the memory, and 3) a fusion block f_{fusion} for aggregating embeddings from short- and long-term branches.

For each tracklet, the short-term module f_{short} takes as inputs its previous T_s states while the long-term memory module f_{long} utilizes longer history with length of T_l ($T_s \ll T_l$). f_{short} and f_{long} are implemented with multi-head cross-attention modules, where the history states are key and value inputs. The input query for f_{short} is the most recent state $\boldsymbol{X}^{t-1}$, while an dynamically updated embedding, called *Dynamic Memory Aggregation Tokens (DMAT)*, $\boldsymbol{Q}_{dmat}^{t-1} = \{{q_{dmat}}_k^{t-1}\}_{k=1:N_{tck}}$ is used for f_{long}. When every tracklet is initiated, it is associated with the same DMAT as others; after that, at time step $t > 0$, DMAT is iteratively updated from the previous step. This design will be further validated in Sec. 4.5. The outputs of the short- and long-term branches, denoted as *Aggregated Short-term Token (AST)* $\boldsymbol{Q}_{AST}^t$ and *Aggregated Long-term Token (ALT)* $\boldsymbol{Q}_{ALT}^t$, are then fused by f_{fusion}. It outputs the track embedding $\boldsymbol{Q}_{tck}^t$ and an updated $\boldsymbol{Q}_{dmat}^t$ where the latter is retained for the next timestep.

3.5. Memory Decoding

The memory decoder Θ_D takes the proposal embedding, track embedding, and the image feature as inputs to produce the final tracking results. It is realized by using stacked Transformer decoder units, where the concatenated proposal and track embeddings $[\boldsymbol{Q}_{pro}^t, \boldsymbol{Q}_{tck}^t]$ are used as queries. Θ_D takes the encoded image feature z_1^t from Θ_H as key and value.

For each entry q_i^t in Θ_D's outputs $[\widehat{\boldsymbol{Q}}_{pro}^t, \widehat{\boldsymbol{Q}}_{tck}^t]$, the decoding process generates three predictions: the bounding box (in the format of offsets w.r.t. the learned reference points), the *objectness score*, and the *uniqueness score*. The objectness score o_i^t for a query q_i^t ranges from 0 to 1, where $o_i^t = 1$ means the model determines the entry is depicting a visible object. The uniqueness score u_i^t also ranges from 0 to 1. When $u_i^t = 1$ the model predicts that the object depicted by q_i^t is unique and should be included in the tracking outputs. Otherwise it needs to be suppressed. We define that $u_i^t = 1$ if $q_i^t \in \widehat{\boldsymbol{Q}}_{tck}^t$. When the model learns to predict u_i^t for each proposal entry, we enforce that a proposal is only considered novel ($u_i^t = 1$) when it is not related to any object being tracked. We can then define a unified **confidence score** for both proposal and track entries as the multiplication of objectiveness and uniqueness scores:

$$s_k^t = o_k^t \cdot u_k^t. \tag{1}$$

The predicted confidence scores of the proposal and track queries are referred to as $\boldsymbol{S}_{tck}^t$ and $\boldsymbol{S}_{pro}^t$, respectively. For each entry q_i^t, the model predicts its bounding box $\mathbf{b}_i^t$, where $\mathbf{b}_i^t \in \mathbb{R}^{4\times 1}$ includes the object's center coordinates, width, and height.

The above formulation allows us to solve the object detection and data association problem simultaneously. In inference, we threshold each entry of $[\widehat{\boldsymbol{Q}}_{pro}^t, \widehat{\boldsymbol{Q}}_{tck}^t]$ with the threshold ϵ and only retain entries with $s_i^t \geq \epsilon$. The resulted entries will automatically bear an track identity or initialize a new track according to whether they come from $\widehat{\boldsymbol{Q}}_{pro}^t$ or $\widehat{\boldsymbol{Q}}_{tck}^t$. We can then obtain the final tracking results by combining the inherited or newly formed track identities with the corresponding bounding box prediction. There is *no* need for further post processing [3, 57, 70].

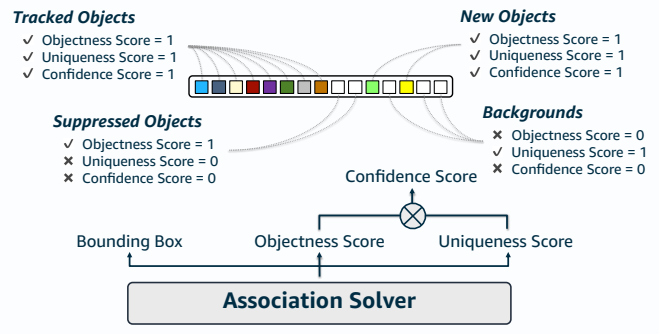

Figure 4. **Illustration of ground truth assignment** to tracked objects, new objects, suppressed objects, and backgrounds. We show the assigned groundtruth scores for each type of entry.

To generate the supervision signals for o_i^t, u_i^t, and $\mathbf{b}_i^t$ on each frame, we first assign the objectiveness score and bounding box to entries in $\widehat{\boldsymbol{Q}}_{tck}^t$ based on whether the tracked object is present in this frame. Then for each entry in $\widehat{\boldsymbol{Q}}_{pro}^t$, we assign the groundtruth bounding boxes, regardless of new or already tracked, to each entry through bipartite matching [6, 12]. Then we assign the groundtruth uniqueness score to each proposal entry, as shown in Fig. 4, based on whether its matched object has been seen before.

3.6. Training MeMOT

We supervise MeMOT with the *tracking loss* computed on the o_i^t, u_i^t, $\mathbf{b}_i^t$ following the assignment process above as

$$L_{tck} = \lambda_{cls}(L'_{obj} + L'_{uni}) + \lambda_{L_1} L'_{bbox} + \lambda_{iou} L'_{iou}, \quad (2)$$

where λs are hyper-parameters for weight scaling, L_{obj} and L_{uni} are focal loss on objectness scores and uniqueness scores, L_{bbox} is L1 loss for bounding box regression, L_{iou} is the generalized IoU loss [43].

Additionally, we apply a detection loss to the proposal embedding similar to Deformable DETR's [73] to enhance MeMOT's localization capability. Specifically, we attach an auxiliary linear decoder to the proposal embedding to output bounding boxes and object classification scores. We then assign the object instances to them as in normal object detection tasks [73] and similarly compute the loss as

$$L_{det} = \lambda_{cls} L_{obj} + \lambda_{L_1} L_{bbox} + \lambda_{iou} L_{iou}. \quad (3)$$

Note the auxiliary decoder is discarded after training.

Following MOTR [69], we compute the tracking loss in a clip by the sum of all individual track queries' losses normalized by the total number of object instances. For a clip with T frames, the overall loss L_{clip} is a combination of the tracking loss and auxiliary detection loss as:

$$\begin{aligned} L_{clip} &= \lambda_{tck} L_{clip-tck} + \lambda_{det} L_{clip-det} \\ &= \frac{\lambda_{tck}}{\sum_{t=0}^{T} N_t} \sum_{t=0}^{T} \sum_{i=0}^{|\boldsymbol{Q}_{tc}^t, \boldsymbol{Q}_{pro}^t|} L_{tck}^{(i,t)} + \frac{\lambda_{det}}{T} \sum_{t=0}^{T} \frac{1}{N_t} \sum_{j=0}^{|\boldsymbol{Q}_{pro}^t|} L_{det}^{(j,t)}, \end{aligned} \quad (4)$$

where $\lambda_{tck} \in \mathbb{R}$ and $\lambda_{det} \in \mathbb{R}$ are the loss weights for balancing the tracking loss and the auxiliary detection loss, respectively. Here N_t denotes the total number of visible objects in the frame at time t.

4. Experiments

4.1. Datasets and Metrics

We evaluate MeMOT on **MOT Challenge** [10, 35] (*i.e.*, MOT16, 17 & 20) datasets. As standard protocols, **CLEAR MOT Metrics** [35] and **HOTA** [33] are used for evaluation.

4.2. Settings

Implementation Details. We implemented our proposed method in PyTorch [38], and performed all experiments on a system with 8 Tesla A100 GPUs. The input frames were resized such that their shorter side is 800 pixels. We used routine data augmentations, including random flip and crop. We adopted ResNet50 [21] and Deformable DETR [73] pre-trained on COCO [27] for hypothesis generation. For all Transformer units, we reduced their number of layers to 4. Our memory buffer contained a maximum of 300 tracks for MOT16/17 benchmarks and 600 tracks for MOT20. Its maximum temporal length is 22 for MOT16/17 and 20 for MOT20, which is mainly limited by the GPU memory. We followed prior work [6, 73] and selected the coefficients of Hungarian loss with λ_{cls}, λ_{L_1} and λ_{iou} as 2, 5, 2, respectively. We set $\lambda_{det} = \lambda_{tck} = 1$ in Eq. 4.

Hyperparameters. We adopted clip-centric training. The length of each clip started from 2 and increased with stride 4 at every 20 epochs. Frames in each clip were sampled with a random interval between 1 to 10. Our model was trained with AdamW [31] optimizer for 200 epochs. The learning rate was initiated to 2×10^{-4} and decreased by 10 at the 100-th epoch. The batch size was set to 1 clip per GPU.

Training Data. When compared with the state-of-the-art methods, MeMOT is trained on the CrowdHuman [44] validation set and MOT17 training set for MOT16 and MOT17 benchmarks. No extra data was used for MOT20. Training with extra data can significantly boost the tracking performance [70]. Thus, as shown in Table 1, we mark out the size of additional training data (*i.e.*, number of frames) that each method used, where the MOT training set itself is referred to as $1.0\times$. More details are included in the appendix.

4.3. Comparison with the State-of-the-art Methods

For fair comparison, we mainly compare MeMOT to methods with an *in-network association solver (IAS)* that predict identities directly without any post-processing. The other type of method applies a *post-network association solver (PAS)* on detection results to perform a series of rule-based linking, such as Hungarian matching with Kalman

Method	Training Data	Transformer	IDF1 ↑	MOTA ↑	HOTA ↑	AssA ↑	IDsw ↓	MT(%) ↑	ML(%) ↓	FP ↓	FN ↓
MOT16 [35]											
FairMOT [70]	13.1x		72.3	69.3	58.3	58.0	815	40.3	16.7	13501	41653
TubeTK [37]	44.5x		62.2	66.9	50.8	47.3	1236	39.0	16.1	11544	47502
CTracker [39]	1.0x		57.2	67.6	48.8	43.7	1897	32.9	23.1	8934	48350
JDE [54]	10.2x		55.8	64.4	-	-	1544	35.4	20.0	-	-
MOTR [69]	1.9x	✓	67.0	66.8	-	-	**586**	34.1	25.7	**10364**	49582
MeMOT (ours)	1.9x	✓	**69.7**	**72.6**	**57.4**	**55.7**	845	**44.9**	**16.6**	14595	**34595**
MOT17 [35]											
CorrTracker [52]	13.1x		73.6	76.5	60.7	58.9	3396	47.6	12.7	29808	99510
FairMOT [70]	13.1x		72.3	73.7	59.3	58.0	3303	43.2	17.3	27507	117477
PermaTrack [49]	18.7x		68.9	73.8	55.5	53.1	3699	43.8	17.2	28998	115104
GSDT [53]	10.2x		66.5	73.2	55.2	51.0	3891	41.7	17.5	263397	120666
TraDeS [60]	3.8x		63.9	69.1	52.7	50.8	3555	36.4	21.5	20892	150060
TransTrack [48]	3.8x	✓	63.5	75.2	54.1	47.9	4614	55.3	10.2	50157	86442
TransCenter [67]	3.8x	✓	62.2	73.2	54.5	49.7	3663	40.8	18.5	23112	123738
TubeTK [37]	44.5x		58.6	63.0	48.0	45.1	4137	31.2	19.9	27060	177483
CTracker [39]	1.0x		57.4	66.6	49.0	37.8	5529	32.2	24.2	22284	160491
TrackFormer [34]	1.0x	✓	63.9	65.0	-	-	3258	-	-	70443	123552
MOTR [69]	1.9x	✓	67.0	67.4	-	-	**1992**	34.6	21.5	**32355**	149400
MeMOT (ours)	1.9x	✓	**69.0**	**72.5**	**56.9**	**55.2**	2724	**43.8**	**18.0**	37221	**115248**
MOT20 [10]											
FairMOT [70]	8.2x		67.3	61.8	54.6	54.7	5243	68.8	7.6	103440	88901
TransTrack [48]	2.7x	✓	59.4	65.0	48.9	45.2	3608	50.1	13.4	27191	150197
TransCenter [67]	2.7x	✓	49.6	58.5	43.5	37.0	4695	48.6	14.9	64217	146019
MeMOT (ours)	1.0x	✓	**66.1**	**63.7**	**54.1**	**55.0**	**1938**	**57.5**	**14.3**	**47882**	**137983**

Table 1. **Evaluation results on MOT challenge datasets**. Trackers with gray background use the in-network association solver (IAS), and others with white background use the post-model association solver (PAS). Best results of IAS are marked in bold.

Filter and re-ID features. Generally, these empirical linking strategies limit their practicability and scalability.

Results on normal scenarios. Table 1 shows that MeMOT achieves the state-of-the-art performance in MOT16/17 among the IAS methods (w/ gray background). It also obtains encouraging detection accuracy (72.6 and 72.5 MOTA on MOT16/17) compared to the PAS methods that are pre-trained with larger detection datasets. For more comprehensive metrics, IDF1 and HOTA, MeMOT achieves comparable results with the state-of-the-art JDE tracker (FairMOT), but uses 5× less training data. MeMOT can keep track of more objects but produce much fewer ID switches (IDsw). For example, on MOT16, MeMOT obtains 44.9% Mostly Tracked (MT) and 16.6% Mostly Lost (ML), outperforming other methods by at least 4.5%, but only getting 845 IDsw. On MOT17, TransTrack and TransCenter show promising detection results with better MT (55.3) and ML (10.2), however, they produce 34% and 69% more IDsw and lower IDF1 (63.5 vs. 62.2 vs. 69.0) than ours. Compared to all Transformer-based methods, MeMOT is significantly better in data association measured by Association Accuracy (AssA). This shows the effectiveness of the learnable association powered by our memory design.

Results on crowded scenarios. MOT20 is a more challenging benchmark with crowded scenarios and serious occlusions. Table 1 shows that MeMOT achieves comparable performance with the state-of-the-art JDE method (FairMOT), but gets 63% reduction in IDsw. Note that FairMOT is trained with 8× more training data than ours. Comparing to other Transformer-based methods, MeMOT outperforms them by 6.7 IDF1 and 5.2 HOTA. By getting a much lower IDsw, our learnable association solver shows its advantage to deal with the occlusion problem. We observe that IoU-based association methods (*e.g.*, TransCenter and TransTrack) fail to handle frequent occlusion, and for the re-identification feature-based methods (*e.g.*, FairMOT), it is hard to obtain high-quality embeddings to measure inter-object similarity due to the small object sizes.

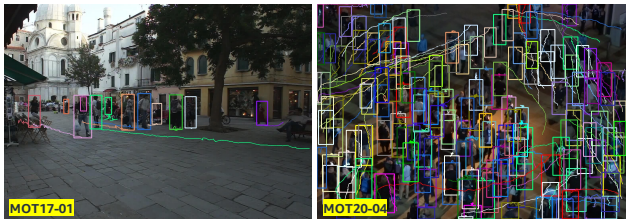

Figure 5. **Examples of our tracking performance** on MOT17 and MOT20. Each identity is shown in a colored bounding box and trajectory in the past 150 frames are displayed.

4.4. Visualization

Object trajectories are visualized in Fig. 5. Results of MOT17-01 show that MeMOT generates long, consistent predictions even when objects pass by each other frequently. Results on MOT20-04 suggest MeMOT's supe-

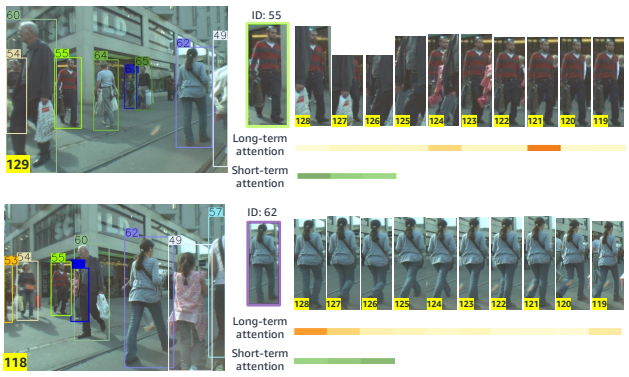

Figure 6. **Visualization of long- and short-term attentions**. **Left**: Tracking results of frame 118 and 129, where tracked objects are displayed in colors with confidence scores. **Right**: Learned long- and short-term attention maps for the intermediate frames of two selected identities (*i.e.*, ID 55 and 62). Darker color represents stronger attention.

rior object detection and association ability in crowd scenarios. Due to the small object size and poor lighting, feature similarity-based association methods [54, 70] are precarious, causing higher IDsw. We provide video demos in the supplementary material for detailed comparison.

In Fig. 6, we also visualize the attention weights of the memory aggregator to elaborate what information is referenced from the memory. For object 55, who is occluded by object 60 from frame 125 to 128, the embedding right before occlusion (frame 124) and a full-body feature (frame 121) contribute the most to re-linking (frame 129) after occlusion. And his short-term attention weights are higher on a less-occluded frame (frame 128) than fully-occluded frames (frame 126 and 127). As for a non-occluded object (*i.e.*, object 62), attention weights are higher on the short-term memory (frame 126 to 128) and the far-away frames are less attended. These observations validate that our memory aggregator is capable of capturing distinctive object features, especially when objects are crossing each other.

4.5. Ablation Studies

We experiment with different memory and model design choices. Unless noted otherwise, we use trimmed models by reducing their number of layers of all Transformer units from 4 to 2. The models are trained on MOT17 training set and validated on MOT15 training set. Validation videos that are overlapped with the training set are excluded.

Effect of short-term memory length. Table 2 compares the performance using different short-term memory lengths, by keeping the long-term memory length T_l as 24. It shows that only using the last two observations (*i.e.*, T_s=2) for short-term memory aggregation slightly decreases the performance. This observation is consistent with results in prior work [34, 48] that propagates tracking results only between adjacent frames. On the other hand, increasing the short-term length from 3 to 5 does not make a big difference. We think these information gaps are compensated by the long-term memory. Considering the accuracy-efficiency trade-off, we set T_s to 3 as default in other experiments.

T_s	T_l	IDF1	MOTA	HOTA	IDsw	DetA	AssA
2	24	72.52	65.62	58.99	**76**	56.97	62.14
3		**73.15**	**68.08**	**59.75**	93	**57.92**	**63.10**
4		72.40	67.11	59.51	93	57.58	62.82
5		72.75	66.65	59.48	92	57.42	62.87

Table 2. Comparisons on different length of short-term memory.

T_s	T_l	IDF1	MOTA	HOTA	IDsw	DetA	AssA
3	3	71.27	67.14	59.09	136	57.85	61.41
	5	71.70	67.94	59.31	136	58.24	61.50
	10	71.66	**68.29**	59.53	117	**58.35**	59.49
	20	72.83	68.21	**59.85**	96	58.03	63.00
	24	**73.15**	68.08	59.75	**93**	57.92	**63.10**

Table 3. Comparisons on different length of long-term memory.

q_s	q_l	IDF1	MOTA	HOTA	IDsw	DetA	AssA
	✓	**73.15**	68.08	**59.75**	**93**	57.92	63.10
✓		67.25	**69.88**	57.36	112	**58.01**	61.76
✓	✓	41.09	59.80	43.28	207	45.36	38.52
		72.30	62.68	58.84	103	55.72	**63.37**

Table 4. Comparisons on different configuration of the short-term cross-attention query q_s and long-term cross-attention query q_l.

Effect of long-term memory length. MeMOT uses a long-term memory to mitigate the occlusion problem. Table 3 shows the effect of different long-term memory lengths T_l from 3 to 24. Note that we set the max length as 24 due to hardware limitation. As T_l grows, the association performance keeps increasing with fewer IDsw and higher IDF1.

Comparing to heuristic memory aggregations. We explore the design of memory aggregation module by first comparing to heuristic algorithms. Considering the tracklet length can be relatively long (up to 24), we do not concatenate the embeddings but test the pooling methods. Then the aggregation can be conducted by using either arithmetic mean or maximum norm over the most recent T frames. Table 5 shows that using these simple pooling methods is incapable of capturing informative track features, resulting in a huge performance drop in IDF1 and MOTA.

Comparing to attention-based memory aggregations. We experiment with another two attention-based aggregation designs in memory encoding. The first one is to only use a cross-attention module, without the separation of long and short memory. This baseline uses the latest observation to query an object's past T embeddings. As shown in Table 6, it produces worse association performance, with -0.51% and -0.34% MOTA for $T = 3$ and $T = 24$, respectively. The IDsw also increases by 6 and 44. Inspired by LSTR [66], the second one is to use the aggregated short-

Method	Parameter	IDF1	MOTA	HOTA	IDs
Ours	-	**73.15**	**68.08**	**59.75**	**93**
Pooling	Average (T=3)	25.04	30.72	21.89	267
	Max (T=3)	46.83	41.28	35.44	235
	Average (T=24)	-	-7.29	-	-
	Max (T=24)	25.78	10.20	6.54	332

Table 5. Comparisons with heuristic memory aggregation design.

Method	Parameter	IDF1	MOTA	HOTA	IDs
Ours	-	**73.15**	68.08	**59.75**	**93**
Single	T=3	72.64	**68.74**	58.94	137
	T=24	72.81	66.25	58.73	99
Long-after-short	-	70.30	65.39	57.03	101

Table 6. Comparisons on adaptive memory aggregation design.

Update	IDF1	MOTA	HOTA	IDsw	DetA	AssA
✓	**73.15**	**68.08**	**59.75**	**93**	**57.92**	**63.10**
	61.03	43.42	49.24	161	42.40	57.96

Table 7. Comparison on the updating of $\boldsymbol{Q}_{dmat}$.

Confidence	IDF1	MOTA	HOTA	IDs	DetA	AssA
Single	69.09	63.51	52.86	104	55.76	61.77
Dual	**73.15**	**68.08**	**59.75**	**93**	**57.92**	**63.10**

Table 8. Comparisons between single and dual confident scores.

term embeddings to retrieve useful information from the long-term memory. The result shows that this design also decreases the performance. We argue that, in the action detection task that LSTR focuses on, the result of each frame is independent and deficient short-term features have a limited effect on future predictions. However, association errors can propagate in MOT, thus using long-term features to compensate for short-term features is more desirable.

Using learnable tokens vs. latest observation for memory aggregation. We explore using either the learnable tokens or the latest observation for long- and short-term memory aggregation, as shown in Table 4. For the short-term token (row2 vs. row4), using the latest observation (row4) yields better association performance (+5.05 IDF1). After fixing the short-term token, using learnable tokens for long-term memory aggregation obtains slightly better performance (row1 vs. row4), with +0.85 IDF1 and -10 IDsw. It is worth noting that using learnable tokens for both long- and short-term branches is risky, getting the IDF1 and MOTA drop to 41.09% and 59.80%. These observations validate our intuition for separating the long- and short-term branches. 1) Due to the temporal variance, the long-term memory may be less informative to match the latest observation but provides diverse features within a tracklet. To extract the supportive context information, using learnable tokens is more effective. 2) Short-term memory features share a high similarity, thus directly querying them with the latest observation can smooth out the noises. 3) These two branches can acquire complementary information.

Dynamic update of memory aggregation tokens. Since we model online tracking as an iterative process, it is worth studying if the long-term memory aggregation tokens should be updated during inference. Results in Table 7 shows that dynamically updating the queries with the most recent information contributes to better detection and tracking performance. By passing the firsthand observation to the long-term queries, more detailed information for current association is extracted, rather than general information.

Effect of uniqueness score for training MeMOT. We introduce the uniqueness score to link new detections with the tracked objects and reject false positives. Here we evaluate its contribution by removing the prediction branch of uniqueness score, as shown in Table 8. Without the uniqueness branch (single output), there are more false positive detections and IDsw. We separate the mixed meanings of the output classification score by splitting the prediction of objectness and uniqueness into two heads. In a single-head architecture, for tracked object queries, the score means the confidence of existence; for the proposal queries, it means the confidence of being a new-born object. While the two purposes share the same classification layer, low confidence values are ambiguous: it means non-objectness, or not a new object. Our design removes the ambiguity and avoids under-training of the classification layers.

4.6. Limitations

As MeMOT is currently trained with supervised learning, it requires video datasets with tracking annotation. However, existing datasets for tracking are still limited in size and diversities, due to the high cost of annotating videos. Developing annotation efficient training methods is crucial to overcoming this difficulty. Although the spatio-temporal memory is shown to be effective in tracking objects consistently, it indeed increases the GPU memory cost in training. This limits the temporal length of the memory and therefore calls for further improvement in efficiency.

5. Conclusion

We proposed MeMOT for online MOT by jointly performing the object detection and data association. MeMOT preserves a large spatio-temporal memory and actively encodes the past observations via an attention-based aggregator. By representing objects as dynamically updated query embeddings, MeMOT predicts object states with an attention mechanism without any post-processing. Extensive experiments validate the effectiveness of MeMOT on object localization and association in crowded scenes.

There are many real-world applications of MOT technology, such as patient or elderly health monitoring, autonomous driving, and collaborative robots. However, there could be unintended usages and we advocate responsible usage complying with applicable laws and regulations.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv:2103.15691*, 2021. 2

[2] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 2010. 2

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019. 1, 2, 3, 4

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2

[5] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020. 1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3, 5

[7] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*, 2010. 2

[8] Peng Chu and Haibin Ling. FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *ICCV*, 2019. 2

[9] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. TransMOT: Spatial-temporal graph transformer for multiple object tracking. *arXiv:2104.00194*, 2021. 2

[10] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003*, 2020. 2, 5, 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[12] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014. 5

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 1

[14] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2009. 1, 2

[15] Weitao Feng, Zhihao Hu, Wei Wu, Junjie Yan, and Wanli Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. *arXiv:1901.06129*, 2019. 2

[16] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, 2021. 2

[17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv:2107.08430*, 2021. 1, 2

[18] Jan Giebel, Darin M Gavrila, and Christoph Schnörr. A bayesian framework for multi-cue 3d object tracking. In *ECCV*, 2004. 2

[19] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv:1410.5401*, 2014. 2

[20] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *TSP*, 2002. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[22] Carine Hue, J-P Le Cadre, and Patrick Pérez. A particle filter to track multiple objects. In *MOT Workshop*, 2001. 2

[23] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *IJCV*, 1998. 2

[24] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *ICCVW*, 2015. 2

[25] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, 2016. 2

[26] Wei Li, Yuanjun Xiong, Shuo Yang, Mingze Xu, Yongxin Wang, and Wei Xia. Semi-TCL: Semi-supervised track contrastive representation learning. *arXiv:2107.02396*, 2021. 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1, 2

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv:2103.14030*, 2021. 2

[30] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv:2106.13230*, 2021. 2

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017. 5

[32] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 2

[33] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 2021. 5

[34] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv:2101.02702*, 2021. 1, 2, 6, 7

[35] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 2, 5, 6

[36] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 2

[37] 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 *CVPR*, 2020. 6

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[39] 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 *ECCV*, 2020. 6

[40] AG Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006. 2

[41] Akshay Rangesh, Pranav Maheshwari, Mez Gebre, Siddhesh Mhatre, Vahid Ramezani, and Mohan M Trivedi. TrackMPNN: A message passing graph neural architecture for multi-object tracking. *arXiv:2101.04206*, 2021. 2

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2

[43] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 5

[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. 5

[45] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *arXiv:2103.13915*, 2021. 2

[46] Chunhua Shen, Anton Van den Hengel, and Anthony Dick. Probabilistic multiple cue integration for particle filter based tracking. *Australian Pattern Recognition Society*, 2003. 2

[47] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NeurIPS*, 2015. 2

[48] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv:2012.15460*, 2020. 2, 6, 7

[49] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, 2021. 6

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[51] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *ACM MM*, 2019. 2

[52] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *CVPR*, 2021. 6

[53] Yongxin Wang, Kris Kitani, and Xinshuo Weng. Joint object detection and multi-object tracking with graph neural networks. In *ICRA*, 2021. 6

[54] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv:1909.12605*, 2019. 2, 6, 7

[55] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2

[56] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv:1410.3916*, 2014. 2

[57] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 1, 2, 4

[58] Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. In *ICLR*, 2019. 2

[59] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 2

[60] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021. 6

[61] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 2

[62] Junliang Xing, Haizhou Ai, and Shihong Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*, 2009. 2

[63] Junliang Xing, Haizhou Ai, Liwei Liu, and Shihong Lao. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *TIP*, 2010. 2

[64] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 2

[65] Mingze Xu, Chenyou Fan, Yuchen Wang, Michael S Ryoo, and David J Crandall. Joint person segmentation and identification in synchronized first-and third-person videos. In *ECCV*, 2018. 2

[66] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *NeurIPS*, 2021. 2, 7

[67] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with dense queries for multiple-object tracking. *arXiv:2103.15145*, 2021. 2, 6

[68] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018. 2

[69] Fangao Zeng, Bin Dong, Tiancai Wang, Cheng Chen, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. *arXiv:2105.03247*, 2021. 1, 2, 5, 6

[70] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of

detection and re-identification in multiple object tracking. *arXiv:2004.01888*, 2020. 1, 2, 4, 5, 6, 7

[71] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 2

[72] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019. 1, 2

[73] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv:2010.04159*, 2020. 2, 3, 5