Looking Beyond Two Frames: End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers

Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, Ian Reid, and Hamid Rezatofighi

Abstract—Tracking a time-varying indefinite number of objects in a video sequence over time remains a challenge despite recent advances in the field. Most existing approaches are not able to properly handle multi-object tracking challenges such as occlusion, in part because they ignore long-term temporal information. To address these shortcomings, we present MO3TR: a truly end-to-end Transformer-based online multi-object tracking (MOT) framework that learns to handle occlusions, track initiation and termination without the need for an explicit data association module or any heuristics. MO3TR encodes object interactions into long-term temporal embeddings using a combination of spatial and temporal Transformers, and recursively uses the information jointly with the input data to estimate the states of all tracked objects over time. The spatial attention mechanism enables our framework to learn implicit representations between all the objects and the objects to the measurements, while the temporal attention mechanism focuses on specific parts of past information, allowing our approach to resolve occlusions over multiple frames. Our experiments demonstrate the potential of this new approach, achieving results on par with or better than the current state-of-the-art on multiple MOT metrics for several popular multi-object tracking benchmarks.

Index Terms—Multi-object tracking, transformer, spatio-temporal model, pedestrian tracking, end-to-end learning

1 INTRODUCTION

Usually discriminating the identity of multiple objects in a scene and creating individual tracks of their movements over time, namely multi-object tracking, is one of the basic yet most crucial vision tasks, imperative to tackle many real-world problems in surveillance, robotics/autonomous driving, health and biology. While being a rather classical AI problem, it is still very challenging to design a reliable multi-object tracking (MOT) system capable of tracking an unknown and time-varying number of objects without the need for an explicit data association module or any heuristics. MO3TR encodes object interactions into long-term temporal embeddings using a combination of spatial and temporal Transformers, and recursively uses the information jointly with the input data to estimate the states of all tracked objects over time. The spatial attention mechanism enables our framework to learn implicit representations between all the objects and the objects to the measurements, while the temporal attention mechanism focuses on specific parts of past information, allowing our approach to resolve occlusions over multiple frames. Our experiments demonstrate the potential of this new approach, achieving results on par with or better than the current state-of-the-art on multiple MOT metrics for several popular multi-object tracking benchmarks.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3213073, provided by the authors. Digital Object Identifier no. 10.1109/TPAMI.2022.3213073

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 1. Looking beyond two frames with MO3TR: Temporal and spatial Transformers jointly pay attention to the current image $I_t$, and the entire embedding history of the two tracked objects (red and green, left). Detection of a previously untracked object (blue) causes initiation of new track (left → middle), while an object exiting the scene (green) leads to track termination (middle → right). Embeddings encoding spatial and temporal interactions are accumulated over time to form individual object-track histories.

multi-object frameworks as their final outputs, i.e. tracks, are generated through a non-learning process. For example, track initiation and termination are commonly tackled by applying different heuristics, and the track assignments are decided upon by applying additional optimization methods, e.g. the Hungarian algorithm [25], max-flow min-cut [26], etc., and the generated tracks may be smoothed by a process such as interpolation or filtering [27].

With the recent rise in popularity of Transformers [28], [29], this rather new deep learning tool has been adapted to solve computer vision problems like object detection [30] by building on top of advances in using deep architectures for set prediction tasks [31], [32]. Concurrent to our work, this idea has also been deployed to some new MOT frameworks [33], [34]. Nonetheless, these methods still either rely on conventional heuristics, e.g. Intersection over Union (IoU) matching [34], or formulate the problem as a two-frames task [33], [34], making them naive approaches to handle long-term occlusions.

In this paper, we show that the MOT problem can be learnt end-to-end without the use of heuristics. Our proposed method MO3TR: Multi-Object TRacking using spatial Transformers and temporal Transformers addresses the key tasks of track initiation and termination as well as occlusion handling. Being a truly end-to-end Transformer-based online multi-object tracking method, MO3TR learns to recursively predict the state of the objects directly from an image sequence stream and encodes long-term temporal information to estimate the states of all objects over time (Fig. 1). In contrast to most existing MOT frameworks, this state-based approach does not require any explicit data association module (see Section 3.4 for a more detailed discussion).

Precisely speaking, MO3TR incorporates long-term temporal information by casting temporal attention over all past embeddings of each individual object, and uses this information to predict an embedding suited for the current time step (Object-to-Temporal Attention, Fig. 2). This access to longer-term temporal information beyond two frames is crucial in enabling the network to learn the difference between occlusion and termination, which is further facilitated through a specific data augmentation strategy employed during training. To factor in the influence of other objects and the visual input measurements, we refine the predicted object embedding by casting spatial attention over all identified objects in the current frame (Object-to-Object Attention) as well as over the objects and the encoded input image (Object-to-Input Attention, Fig. 2).

The idea of this joint approach relates to the natural way humans perceive such scenarios: We expect certain objects to become occluded given their past trajectory and their surroundings, and predict when and where they are likely to reappear.

To summarize, our main contributions are as follows:

1) We introduce an end-to-end tracking approach that learns to encode longer-term information beyond two frames through temporal and spatial Transformers, and recursively predicts all states of the tracked objects.

2) We realize joint learning of object initialization, termination and occlusion handling without explicit data association and eliminate the need for heuristic post-processing.

3) MO3TR reaches performances comparable or better than current state-of-the-art methods on several popular multi-object tracking benchmarks.

2 RELATED WORK

Tracking-by-Detection. Tracking-by-detection treats the multi-object tracking (MOT) task as a two-stage problem. First, all objects in each frame are identified using an object detector [9], [11], [12], [13]. Detected objects are then associated over frames, resulting in tracks [3], [35]. The incorporation of appearance features and motion information has been proven to be of great importance for MOT. Appearance and ReID features have been extensively utilized to improve the robustness of multi-object tracking [15], [36], [37], [38], [39]. Further, incorporating motion has been achieved by utilizing a Kalman filter [27] to approximate the displacement of boxes between frames in a linear fashion and with the constant velocity assumption [1], [40] to associate detections [3], [18], or by incorporating environment information [41] and possible intentions [42], [43]. Recently, more complex and data-driven models have been proposed to model motion [24], [44], [45], [46] in a deterministic [16], [47] and probabilistic [44], [48], [49] manner. Graph neural networks have been also used in the recent detection-based MOT frameworks, conducive to extract a reliable global feature representation from visual and/or motion cues [50], [51], [52], [53], [54], [55], with many such methods utilising an offline approach to tackle the tracking problem.

Despite being highly interrelated, detection and tracking tasks are treated mostly as independent in this line of works. Further, the performance of tracking-by-detection methods often highly relies on incorporating heuristics and post-processing steps to infer track initiation and termination, handle occlusions and assign tracks.

Joint Detection and Tracking. The recent trend in MOT has been to move from disjoint detection and tracking components towards unified models, e.g. by using the same

1. Code publicly available at https://github.com/alanzty/MO3TR
network to jointly learn detection and embedding of objects [17]. While embeddings are here learnt explicitly with contrastive losses, our work learns the representations of the objects implicitly. Findings that the explicit learning of detection and appearance embeddings can often be contradicting have inspired approaches seeing to achieve fairness between the two tasks [56]. Other works have attempted to further improve joint detection and embedding learning through the structure of graph neural networks [57], or focused on regressing previous track locations to new locations in the current frame via a regression head [21], [22], [24]. Even deeper coupling of tracking and detection has recently been investigated by exploiting object motion cues from the previous frame(s) to better detect objects in the current frame [58]. This idea somewhat resembles the propagation of tracked objects in our framework which is one step towards learning tracking, but does not yet allow end-to-end learning of propagation, initialization and termination. Advances in occlusion scenarios have been achieved in recent work by relying on the assumption that any physical object existing now will likely continue to exist during the period of tracking, helping to hallucinated the location of even fully occluded objects [59]. Another recent stream of works is inspired by the concept of similarity learning that is dominant in many single generic object tracking methods [60] and provides strong discriminative power to distinguish objects of interest from their surroundings and can help propagation of existing objects [61], [62], [63]. Although detection and tracking are not disjoint components in these works, they still suffer from some shortcomings. These works formulate the problem as detection matching between very few (mostly two) frames, thus solving the problem locally and ignoring long-term temporal information. Furthermore, these approaches still rely on conventional post processing steps and heuristics to generate the tracks, and recent gains have mainly been achieved due to specifically tailored components such as hand-designed track management modules [20]. Given the exhibited difficulties of most methods in increasingly complex tracking scenarios, we argue that solving the challenging MOT task requires long-term temporal encoding of object dynamics to handle object initiation, termination, occlusion and tracking – a strategy that we take with our proposed method.

Transformers in Vision. Recently, Transformers [28] have been widely applied to many computer vision problems [29], [30], [64], [65], [66], [67], including two MOT methods [33], [34] that have been developed concurrently to this work. [34] performs multi-object tracking using a query-key mechanism which relies on heuristic post processing to generate final tracks. Trackformer [33] has been proposed as a Transformer-based model which achieves joint detection and tracking by converting the existing DETR [30] object detector to an end-to-end trainable MOT pipeline. However, it still considers solely local information (two consecutive frames) to learn and infer tracks and thus ignores long-term temporal object dynamics, which are essential for effective learning of all MOT components as we will demonstrate in the remainder of this work.

This Paper. To overcome many of the aforementioned limitations of previous works, we propose an end-to-end multi-object tracking approach that leverages the architectural benefits of current Transformer models. Our method learns to jointly track multiple existing objects, handle their occlusion or terminate their tracks (depending on the situation), and to initiate new tracks – all while considering long-term temporal object information.

3 MO3TR

Learning an object representation that encodes both the object’s own state over time and the interaction with its surroundings is vital to allow reasoning about three key challenges present in end-to-end multi-object tracking (MOT), namely track initiation, termination and occlusion handling. In this section, we demonstrate how such a representation can be acquired and continuously updated through our proposed framework: Multi-Object TRacking
using spatial TRansformers and temporal TRansformers – short MO3STR (Fig. 2). We further introduce a training paradigm that facilitates learning of how to resolve these three challenges in a joint and completely end-to-end trainable manner. We start off by first presenting an overview of our framework and introduce the notation used throughout this paper, followed by a detailed introduction of the core components.

### 3.1 System Overview and Notation

The goal of tracking multiple objects in a video sequence of \( T \) frames is to retrieve an overall set of tracks \( T^t \) representing the individual trajectories for all uniquely identified objects present in at least one frame of the sequence. Given the first frame \( I_0 \) at time \( t_0 \), our model tentatively initializes a set of tracks \( T_0 \) based on all objects identified for this frame. From the next time step \( t > 0 \) onward, the model aims to compute a set of embeddings \( \mathbf{z}_t = \{z^1_t, z^2_t, \ldots, z^M_t\} \) representing all \( M \) objects present in the scene at time \( t \) (Fig. 2). Taking in the track history \( T_{t-1}^{t-1} \) from the previous time step, we predict a set of embeddings \( \mathbf{z}_t \) for the current time step based on the past representations of all objects using temporal attention (Fig. 2: Object-to-Temporal Attention; Section 3.2). Together with a learnt set of representation queries \( Z_Q \) proposing the initiation of new object tracks, these predicted object representations are processed by our first spatial attention module to reason about the interaction occurring between different objects (Fig. 2: Object-to-Object Attention; Section 3.3). This refined set of intermediate object representations \( \mathbf{z}_t^0 \) is then passed to the second spatial attention module which takes the interaction between the objects and the scene into account by casting attention over the object embeddings and the visual information of the current frame \( I_t \) transformed into its feature map \( \mathbf{x}_t \) (Fig. 2: Object-to-Input Attention; Section 3.3). This two-step incorporation of spatial information into the embeddings is iteratively performed multiple times over several Transformer layers (alternating between self- and cross-attention), returning the final output set of refined object representations \( \mathbf{z}_t \) for time step \( t \). Please refer to Section 4.1 for architectural details.

The encoding of temporal and spatial information into a representative ‘state’ embedding for any object \( m \) at time \( t \)

\[
\mathbf{z}_t^m = f(T_{t-1}^{t-1}, Z_Q, \mathbf{z}_t),
\]

(1)

can be summarized as a learnt function \( f(\cdot) \) of the track history \( T_{t-1}^{t-1} \), the learnt set of initiation queries \( Z_Q \) and the encoded image feature map \( \mathbf{x}_t \). This function representation demonstrates our main objective to enable the framework to learn the best possible way to relate the visual input to the objects’ internal states, without enforcing overly-restrictive constraints or explicit data association.

The use of the resulting embedding set \( \mathbf{z}_t \) is our framework is twofold. Tracking results in the form of object-specific class scores \( c^m \) and corresponding bounding boxes \( b^m \) for the current frame are obtained through simple classification and bounding box regression networks (Fig. 2). Further, the subset of embeddings yielding a reasonably high probability of representing an object present in the current frame \( (q^m_{\mathbf{x}_t}(\mathbf{c}_m) > 0.5) \) is added to the track history to form the basis for the prediction performed in the next time step. Throughout the entire video sequence, new tracks \( T^m_n \) representing objects that enter the scene are initialized, while previous tracks may be terminated for objects no longer present. This leads to an overall set of tracks \( T_T = \{T^1_1, \ldots, T^N_N\} \) for all \( N \) uniquely identified objects present in at least one frame of the video sequence of length \( T \), with their life span indicated by the subscript as initiation time (\( \text{start ‘s’} \)) and termination time (\( \text{end ‘e’} \)).

### 3.2 Learning Long-Term Temporal Embeddings

Discerning whether an object is not visible in a given frame due to occlusion or because it is no longer present in the scene can be challenging. Considering that visual features extracted at an object’s expected position in the image during partial or full occlusion are not describing the actual object they aim to represent increases this even further. Humans naturally reach decisions in such scenarios by considering the available information jointly and over multiple time steps; we are able cherry-pick frames that provide helpful information, ignore the ones that prove unhelpful for the task, and predict if, how and where an object is expected to re-appear in the current or a future frame. Intuitively, MO3STR follows a similar approach.

Our framework learns the temporal behavior of objects jointly with the rest of the model through a Transformer-based component [28] that we nickname temporal Transformer. For any tracked object \( m \) at time \( t \), the temporal Transformer casts attention over all embeddings contained in the object’s track history \( T^m_{t-1} = \{z^m_1, \ldots, z^m_t\} \). Leveraging the information from previous frames encoded within these embeddings, it then predicts an expected object representation \( \mathbf{z}_t^m \) for the current time step \( t \). In other words, it predicts an object’s state – a condensed version of all the information required to reason about the object’s most-likely location, identity and other properties to allow tracking.

While the architecture of a Transformer [28] is particularly suited for this purpose due to its ability to process and relate elements of sets like our track history, it is permutation-invariant in nature – a property that is undesired in cases like ours where the indexing of the set plays a crucial role. We thus supplement all embeddings \( \{z^m_1, \ldots, z^m_{t-1}\} \) that form part of an object’s track history \( T^m_{t-1} \) by adding positional encodings [28] to represent their relative time in the sequence and provide the Transformer in this way with the ability to leverage information from the time axis of the track. We denote the time-encoded track history by \( T^m_{\text{pe}} \) and individual positional time encodings for time \( t \) as \( p_e \in \mathbb{R} \). Passing the request for an embedding estimate of the current time step \( t \) in form of the positional time encoding \( p_e \) as a query to the Transformer \(^2\) and providing \( T^m_{t-1} \) as basis for keys and values, we retrieve the predicted object embedding

\[
\mathbf{z}_t^m = \Psi \left( \frac{1}{\sqrt{d_z}} q^m(p_e) k^m(T^m_{t-1})^T v^m(T^m_{t-1}) \right),
\]

(2)

where \( \Psi \) represents the \( \text{softmax} \) operator, \( q^m(\cdot) \), \( k^m(\cdot) \) and \( v^m(\cdot) \) are learnt query, key and value functions of the

2 Note that this method allows to predict embeddings for any future time step, and could thus be easily extended to further applications like trajectory forecasting, or similar.
temporal Transformer, respectively. The dimension of the object embeddings is denoted by $d_z \in \mathbb{R}$.

In other words, the predicted representation $\tilde{z}_m^t$ of object $m$ is computed through a dynamically weighted combination of all its previous embeddings. This allows the temporal Transformer to

- i) incorporate helpful and ignore irrelevant or faulty information from previous time steps, and
- ii) predict upcoming occlusions and create appropriate embeddings that might focus more on conveying important positional rather than visual information in such cases.

While these tasks resemble those usually performed via heuristics and manual parameter tuning during track management, MO3TR learns these dependencies end-to-end without the need of such heuristics.

In practice, the prediction of object representations introduced for the example of one tracked object in (2) is performed in a batched-parallel manner for the entire set of existing tracks $T_{t-1}$ over multiple Transformer layers, resulting in the output set $\tilde{Z}_t$ of the temporal Transformers that is passed as input to the spatial Transformers (Fig. 2). Note that the size of this set is dynamic and depends on the number of tracked objects. Details on how the temporal Transformer is trained are provided in Section 3.5 and architectural details in Section 4.1.

### 3.3 Learning Spatial Interactions

Multiple pedestrians that are present in the same environment not only significantly influence each others movements, but also their respective visual appearance through occluding each other when perceived from a fixed viewpoint. In this section, we introduce how MO3TR learns to incorporate these dependencies into the object representations. Starting from how detection and track initiation are performed within the concept of Transformers, we then detail the refinement of object embeddings by including the interaction between objects, followed by the interaction between objects and the input image.

**Initiation of New Tracks.** For a new and previously untracked object $m$ spawning at any time $t$, a corresponding track history $T_{t-1}^m$ does not yet exist and hence, no predicted embedding is passed from the temporal to the spatial Transformer (Fig. 2). To allow initiation of new tracks for such objects, we build upon [30] and learn a fixed set of initiation queries $Z_Q$ (in other words, a fixed set of learnt embeddings that acts as input to the decoder). Intuitively, these queries learn to propose embeddings that lead the spatial Transformer to check for objects with certain properties and at certain locations in the visual input data. Importantly, these queries are considered jointly with the ones propagated from the temporal Transformer to avoid duplicate tracks. Note that while our main method MO3TR is based around learning a fixed set of initiation queries, these can also be predicted for each image via a modified encoder – an extension we discuss in Section 4.2.

**Interaction Between Tracked Objects.** We use self-attention [28] to capture the influence tracked objects have onto each other’s motion behavior and appearance. This interaction aspect is incorporated into the object embeddings by computing an updated version of the representation set

$$Z'_t = \Psi \left( \frac{1}{\sqrt{d_z}} q^e(\tilde{z}_t^e) k^e(\tilde{z}_t^e)^T \right) v^e(\tilde{z}_t^e),$$

where $q^e(\cdot)$, $k^e(\cdot)$ and $v^e(\cdot)$ are all learnt functions of the concatenated object embedding set $\tilde{Z}_t = \{\tilde{Z}_t, Z_Q\}$. The dimension of the embeddings is denoted by $d_z$ and $\Psi$ represents the $\text{softmax}$ operator. Relating this approach to the classical Transformer formulation, the learnt functions conceptually represent the queries, keys and values introduced in [28], respectively.

**Interaction Between Objects and the Input Image.** While the previous step of modeling the relationships between different objects can be interpreted as a refined way of proposing *where to look* and *what to look for* regarding existing and new objects, these proposals naturally have to be related to the actual input data of the current time step to draw conclusions about potential initiation, termination or further tracking of any given object. This relationship between the set of object embeddings and the input image is modeled through encoder-decoder attention (*aka* cross-attention) to relate all object representations to the encoded visual information of the current image (i.e. measurement). Evaluating this interaction results in the computation of a second update to the set of object representations

$$Z''_t = \Psi \left( \frac{1}{\sqrt{d_z}} q^{cr}(Z'_t) k^{cr}(\mathbf{x}_t)^T \right) v^{cr}(\mathbf{x}_t),$$

where $q^{cr}(\cdot)$ is a learnt function of the pre-refined object embeddings $Z'_t$ and $k^{cr}(\cdot)$ and $v^{cr}(\cdot)$ are learnt functions of the image embedding $\mathbf{x}_t$ produced by a CNN backbone and a Transformer encoder. The embedding dimension is denoted by $d_z$ and $\Psi$ represents the $\text{softmax}$ operator.

**Combining Interactions for Refined Embeddings.** In practice, the two previously described update steps are performed consecutively with (4) taking as input the result of (3), and are iteratively repeated over several layers of the Transformer architecture. This sequential incorporation of updates into the representation is inspired by DETR [30], where self-attention and cross-attention modules are similarly deployed in a sequential manner. Using both introduced concepts of *Object-to-Object* and *Object-to-Input* attention allows our model to globally reason about all tracked objects via their pair-wise relationships, while employing the current image as context information to retrieve the final set of updated object representations $Z_t$.

**Updating the Track History.** After each frame is processed by the entire framework, the final set of embeddings $Z_t$ of objects identified to be present in the frame is added to the track history $T_{t-1}$, creating the basis for the next prediction of embeddings by the temporal Transformer (Fig. 2). We consistently append new embeddings from the right-hand side, followed by right-aligning the entire set of embeddings. Due to the different lengths of tracks for different objects, this procedure aligns embeddings representing identical time steps, a method that we found to help stabilize training and improve the inference of the temporal Transformer (see ablation studies in Section 4.4 and Table 7).
3.4 Removing the Need for Explicit Data Association

In the introductory section of this paper, we briefly noted that our proposed MO3TR performs tracking without any explicit data association module. To elaborate what we mean by that and how multi-object tracking (MOT) without an explicitly formulated data association task is feasible, let us reconsider the actual definition of the MOT problem: Finding a mapping from any given input data, e.g. an image sequence, to the output data, i.e. a set of object states over time. In any learning scheme, given a suitable learning model, this mapping function can theoretically be learned end-to-end without the requirement for solving any additional auxiliary task, as long as the provided inputs and outputs are clearly defined. The firmly established task of performing data association between detections and objects, usually via a minimum cost assignment solved by using the Hungarian algorithm, is nothing more than such an auxiliary task that has originally been created to solve tracking based on the tracking-by-detection paradigm. An end-to-end learning model, however, can learn to directly infer implicit correspondences – thus rendering the explicit formulation of this task obsolete.

Precisely speaking, our end-to-end tracking model learns to relate the visual input information to the internal states of the objects via a self-supervised attention scheme. We realize this through the introduced combination of Transformers [28] to distill the available spatial and temporal information into representative object embeddings (i.e. the object states), making the explicit formulation of any auxiliary data association strategy unnecessary.

3.5 Training MO3TR

The training procedure of MO3TR (Fig. 2) is composed of two key tasks: (i) creating a set of suitable tracklets that can be used as input $T_{t-1}$ to the temporal Transformer, and (ii) assigning the predicted set of $M$ output embeddings $Z_t = \{z_i^t\}_{i=1}^{M}$ to corresponding ground truth labels of the training set, and applying a corresponding loss to facilitate training. With the number of output embeddings being by design larger than the number of objects in the scene, matching occurs either with trackable objects or the background (no-object) class.

**Constructing the Input Tracklet Set.** The input to the model at any given time $t$ is defined as the track history $T_{t-1}$ and the current image $I_t$. To construct a corresponding $T_{t-1}$ for any $I_t$ sampled from the dataset during training, we first extract the ordered set of $K$ directly preceding images $\{k_i\}_{i=1}^{K}$ from the training sequence. Passing these images without track history to MO3TR causes the framework to perform track initiation for all identified objects in each frame by using the trainable embeddings $Z_{k_i}$, returning an ordered set of output embedding sets $\{Z_{k_i}\}_{i=1}^{K}$. Each output embedding set $Z_{k_i}$ contains a variable number of $M_i$ embeddings representing objects in the respective frame $k_i$. We use multilayer perceptrons (MLPs) to extract corresponding bounding boxes $b_k^i$ and class scores $\hat{c}_k^i$ from each of these object embeddings $z_k^i \in Z_{k_i}$, resulting in a set of $M_i$ object-specific pairs denoted as $\{\hat{g}_{m_i}^k, \hat{a}_{m_i}^k\}_{m_i=1}^{M_i}$ for each frame $k_i$. The pairs are then matched with the ground truth $\{g_{k_i}^i\}_{i=1}^{M_i}$ of the respective frame through computing a bipartite matching between these sets following [30]. The permutation $\hat{s}_k$ of the $M_k$ predicted elements with lowest pair-wise matching cost $C_{\text{matching}}$ is determined by solving the assignment problem

$$\hat{s}_k = \arg \min_{\sigma \in \mathcal{S}} \sum_{i=1}^{M_k} C_{\text{matching}}(\hat{y}_k^i, \hat{y}_k^{\sigma(i)})$$

through the Hungarian algorithm [25], with the matching cost taking both the probability of correct class prediction $\hat{p}_{\text{giou}}(\hat{c}_k^i)$ and bounding box similarity into account

$$C_{\text{matching}} = -\hat{p}_{\text{giou}}(\hat{c}_k^i) + C_{\text{bbox}}(\hat{b}_k^i, \hat{b}_k^{\sigma(i)})$$.

We follow [30] and use a linear combination of $L1$ distance and the scale-invariant generalized intersection over union [68] cost $C_{\text{giou}}$ to mitigate any possible scale issues arising from different box sizes. The resulting bounding box cost with weights $g_{\text{giou}} \in \mathbb{R}^+$ is then defined as

$$C_{\text{bbox}} = g_{\text{giou}} \left\| b_k^i - \hat{b}_k^{\sigma(i)} \right\|_1 + g_{\text{giou}} C_{\text{giou}}(\hat{b}_k^i, \hat{b}_k^{\sigma(i)})$$.

The identified minimum cost matching between the output and ground truth sets is used to assign all embeddings classified as objects their respective identities annotated in the ground truth labels. The objects of all $K$ frames are accumulated, grouped regarding their assigned identities and sorted in time-ascending order to form the overall set of previous object tracks $T_{t-1}$ serving as input to our model.

**Beyond Two Frames.** Most recent tracking works attempt to learn track initiation and termination by sampling two frames out of a short sequence. This is however only a sparse and often poor approximation of the actual complex tracking scenario and thus frequently requires significant post-processing or dataset specific tuning of the augmentation process (e.g. adjusting the false positive rate) due to the gap between the scenarios encountered during training and at inference time. In this work, we attempt to take a step towards closing this gap and use our model to infer the results for up to 30 consecutive frames during training. We then employ the same algorithm underlying the CLEAR metric [69] to match our predicted tracklets and ground truth annotations. This procedure better replicates the scenarios encountered during inference already during the training process, allowing for complex scenarios to naturally unfold and for errors to accumulate over multiple frames – thus helping to learn the characteristics of the tracking problem from the data itself rather than based on heuristic choices of dataset-specific augmentation parameters.

**Losses.** Given the created input set of tracks $T_{t-1}$ and the image $I_t$, MO3TR predicts an output set of object embeddings $Z_t = \{z_1^t, z_2^t, \ldots, z_M^t\}$ at time $t$. Similar to before, we extract bounding boxes and class scores for each embedding in the set. However, embeddings that possess a track history already have unique identities associated to them and are thus directly matched with the respective ground truth elements. Only newly initiated embeddings without track history are then matched with remaining unassigned ground truth labels as previously described. Elements that could not be matched are assigned the background class.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Finally, we re-use (6) and (7) for \( k = t \) and apply them as our loss to the matched elements of the output set.

**Data Augmentation.** Most datasets are highly imbalanced regarding the occurrence of occlusion, initiation and termination scenarios. To facilitate learning of correct tracking behaviour and aid our previously described multi-frame training procedure, we propose to further mitigate the imbalance problem by modelling similar effects through augmentation:

1. We randomly drop a certain number of embeddings in the track history to simulate cases where the object could not be identified for some frames, aiming to increase robustness. If the most recent embedding is dropped, the model can learn to re-identify objects.
2. Random false positive examples are inserted into the history to simulate false detection and faulty appearance information due to occlusion. This aims at encouraging the model to learn to ignore unsuited representations through its attention mechanism.
3. We randomly select the sequence length used to create the track history during training to increase the model’s capability to deal with varying track lengths.

The importance of these augmentation strategies is investigated and clearly demonstrated during our ablation studies in Section 4.4 and Table 7, leading to significant improvements compared to non-augmented training.

### 4 Experiments and Discussion

In this section, we demonstrate and discuss the performance of MO3TR by comparing against other multi-object tracking methods on popular MOT benchmarks and evaluate different aspects of our contribution in detailed ablation studies. We further provide all implementation details required to reproduce our findings.

**Datasets.** We use the widely established MOT16 and MOT17 datasets as well as the rather new MOT20 dataset from the MOTChallenge benchmarks to evaluate and compare MO3TR with other state of the art models.

Both MOT16 and MOT17 contain seven training and test sequences each, capturing crowded indoor or outdoor areas via moving and static cameras from various viewpoints. MOT20 contains eight sequences (four training and four test), focusing on heavily crowded scenes with a very high number of people in both day and night scenarios. Pedestrians are often heavily occluded by other pedestrians or background objects across all three datasets, making identity-preserving tracking challenging. Three sets of public detections are provided with MOT17 (DPM [9], FRCNN [12] and SDP [13]), one with MOT16 (DPM) and one with MOT20 (FRCNN). For ablation studies, we combine sequences of the new MOT20 benchmark [79] and 2DMOT15 [80] to form a diverse validation set covering both indoor and outdoor scenes at various pedestrian density levels.

**Evaluation Metrics.** To evaluate MO3TR and compare its performance to other state-of-the-art tracking approaches, we use the standard set of metrics recognized by the tracking community [69], [81].

1. **IDF1.** One widely used metric to summarize the tracking performance of a tracking algorithm is the identity F1 score (IDF1). MOTA combines the measures for the three error sources of false positives, false negatives and identity switches into one compact measure, and a higher MOTA score implies better performance of the respective tracking approach. The IDF1 represents the ratio of correctly identified detections over the average number of ground-truth and overall computed detections. We additionally report our results on the recently proposed higher order tracking accuracy metric (HOTA), which aims to combine the effects of accurate detection, association and localization, and better aligns with human perception of tracking performance [82].

All reported results are computed using the official evaluation code of the MOTChallenge benchmark. For further details, please refer to the supplementary material, available online.

### 4.1 Implementation Details of MO3TR

**Multi-Stage Training Details.** To implement the general training strategy introduced in Section 3.5, we employ a multi-stage training concept to train MO3TR end-to-end. First, our COCO [83] pretrained ResNet50 [84] backbone is, together with the encoder and spatial Transformers, trained on the combination of the CrowdHuman [85], ETH [86] and CUHK-SYSU [87] datasets for 300 epochs on a pedestrian detection task. This training procedure is similar to the one introduced in DETR [30]. Afterwards, we engage our temporal Transformer and train the entire model end-to-end using the respective dataset (e.g., MOT17) for another 100 epochs. The initial learning rate for both training tasks is 1e-4, and is dropped by a factor of 10 at 0.3 and 0.4 times the total number of epochs (i.e. at 90/120 and 60/80 for first and second training stage, respectively). Relative weights of our loss are the same as described in [30], and we choose the number of initiation queries to \( Z_0 = 100 \). The input sequence length representing object track histories is \( len(T_{m-1}) \) for any object \( m \) varies randomly from 1 to 30 frames – meaning that we can expect our model to re-identify objects that have been occluded for at most this number of frames. To reduce overall memory consumption, we thus drop the tracklet for any object \( m \) from our set of active tracks \( T \) if the respective object has not been successfully identified for 30 frames. Note that this cut-off is a practical design choice given our limited computational resources, and does not affect the capability of our approach to track across longer sequences. To enhance the learning of temporal encoding, we predict 10 future frames instead of one and compute the total loss. Please note that while we choose a multi-stage training strategy for increased training stability and reduced initial memory footprint, our approach is fully differentiable without the use of non-differentiable heuristics and can thus be trained entirely ‘end-to-end’. We provide additional experiments and discussion on using a...
TABLE 1
Results on the MOT16 Benchmark [70] Test Set Using public Detections

| Method                  | MOTA | IDFI | HOTA | MT | ML | FP | FN | IDS |
|-------------------------|------|------|------|----|----|----|----|-----|
| Offline                 |      |      |      |    |    |    |    |     |
| eHAF [53]               | 47.2 | 52.4 | 40.3 | 18.6 | 42.8 | 12,586 | 83,107 | 542  |
| NOTA [72]               | 49.8 | 55.3 | 40.7 | 17.9 | 37.7 | 7,428   | 83,614 | 614  |
| MPNTrack [50]           | 58.6 | 61.7 | 48.9 | 27.3 | 34.0 | 4,949   | 70,252 | 354  |
| LPC_MOT [55]            | 58.8 | 67.6 | 51.7 | 27.3 | 35.0 | 6,167   | 68,432 | 435  |
| Lif_T [51]              | 61.3 | 64.7 | 50.8 | 27.0 | 34.0 | 4,844   | 65,401 | 389  |
| ApLift [52]             | 61.7 | 66.1 | 51.3 | 34.3 | 31.2 | 9,168   | 60,180 | 499  |
| Online                  |      |      |      |    |    |    |    |     |
| EAMTT [73]              | 38.8 | 42.4 | 32.5 | 7.9  | 49.1 | 8,114   | 102,452 | 965  |
| DMAN [74]               | 46.1 | 54.8 | 40.3 | 17.4 | 42.7 | 7,909   | 89,874 | 532  |
| AMIR [16]               | 47.2 | 46.3 | 35.1 | 14.0 | 41.6 | 2,681   | 92,856 | 774  |
| MOTDT17 [75]            | 47.6 | 50.9 | 38.4 | 15.2 | 38.3 | 9,253   | 85,431 | 792  |
| STRN [76]               | 48.5 | 53.9 | 39.7 | 17.0 | 34.9 | 9,038   | 84,178 | 747  |
| UMA [77]                | 50.5 | 52.8 | -    | 17.8 | 33.7 | 7,587   | 81,924 | 685  |
| Tracktor++ [21]         | 54.4 | 52.5 | 42.3 | 19.0 | 36.9 | 3,280   | 79,149 | 682  |
| DeepMOT-T [78]          | 54.8 | 53.4 | 42.2 | 19.1 | 37.0 | 2,955   | 78,765 | 645  |
| Tracktor++v2 [21]       | 56.2 | 54.9 | 44.6 | 20.7 | 35.8 | 2,394   | 76,844 | 617  |
| TINF [71]               | 57.6 | -    | -    | -    | -    | 30.0    | 12,121 | 733  |
| TM0H [20]               | 63.2 | 63.5 | 50.7 | 27.0 | 31.0 | 3,122   | 63,376 | 635  |
| MO3TR (Res50, IoU)      | 63.5 | 60.3 | 49.6 | 27.4 | 22.4 | 6,987   | 56,691 | 880  |
| MO3TR (Res50, CD)       | 64.2 | 60.6 | 50.3 | 31.6 | 18.3 | 7,620   | 56,761 | 929  |

**Bold and underlined numbers indicate best and second best result, respectively. In case the two versions of our method fill both spots, we additionally use dotted underline to indicate the next best result. Metrics that were not available for comparison are indicated via a hyphen (–). More detailed results of our approach across the individual sequences of the benchmark are provided in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3213073. † MT and ML in [71] were reported without fractional digit.**

Single-stage joint training approach in the supplementary material, available online.

**Model Components.** We use an 8-head 3-layer self-attention module as temporal Transformer (Section 3.2) and 6 pairs of 8-head 6-layer modules performing self- and cross-attention in an alternating manner as spatial Transformer (Section 3.3). Both Transformers use 256 d embeddings, and we use 1D and 2D sinusoidal positional encodings to supplement the input data for the temporal and spatial Transformers, respectively. Classification and bounding box regression are performed via one and three layer MLPs.

**Computational Complexity for Training.** We train our MO3TR model using 4 GTX 1080ti GPUs with 11 GB memory each with a total batch size of 32 during the first training stage (detection task), and on one single GPU with a batch size of 4 in the second stage. It is to be noted that these computational requirements are significantly lower than for other recently published approaches in this field. We expect the performance of our model to further increase through bigger backbones and longer sequence lengths as well as an increased number of objects per frame.

**Public Detection.** We evaluate the tracking performance using the public detections provided by the MOTChallenge. Since our method is based on embeddings and thus not able to directly produce tracks from these provided detections, we follow [24] in filtering our initiations by the public detections using bounding box center distances (CD), and only allow initiation of matched and thus publicly detected tracks. We additionally report our results achieved by using the strategy of using the public detections to filter track initiations via the Intersection over Union (IoU) recently introduced in [33].

**Private Detection.** To facilitate better comparison to other methods, we additionally report the performance on tracking benchmark using our method’s own, unfiltered detections – commonly referred to as private detections.

### 4.2 Extending MO3TR by Predicting Initiation Queries

Observing that current set-based detectors like DETR [30] with ResNet [84] backbone are still often significantly outperformed by non-set based versions like [88], we explore the possibility of extending our MO3TR setup via the use of a more powerful multi-scale encoder to improve the detection capabilities of our framework. For our investigations, we choose the multi-scale Deformable DTR [89] with the multi-scale DarkNet CNN backbone [88] (initialized with COCO-pretrained [83] weights) and train this new encoder on the CrowdHuman [85] and MOT20 [70] datasets using the detection head and losses proposed in [88], following their training strategy and hyperparameter setup. Given our computational resource constraints, we then freeze the backbone and head, and continue to train our tracking framework for 50 epochs on the respective MOT dataset. Instead of learning a fixed set of initiation queries like with our base MO3TR version, we investigate the advantage of dynamic query generation via the encoder and directly retrieve our desired set of initiation queries $Q_0$ from the encoder. In detail, we use the classifier part of the detection head to select the top $k = 100$ embeddings from the Darknet output as initiation queries, concatenate them as previously introduced with the output set of our temporal Transformer and pass it to the spatial Transformer module as described in Sections 3.2 and 3.3. We refer to this extended method as MO3TR-PIQ (due to its Predicted Initiation Queries).
TABLE 2
Results on the MOT17 Benchmark [70] Test Set Using public Detections

| Method               | Backbone | PD | MOTA↑ | IDF1↑ | HOTA↑ | MT↑ | ML↓ | FP↓ | FN↓ | IDs↓ |
|----------------------|----------|----|-------|-------|-------|-----|-----|-----|-----|-----|
| Offline              |          |    |       |       |       |     |     |     |     |     |
| jCC [90]             | –        | IN | 51.2  | 54.5  | 42.5  | 20.9| 37.0| 25,937| 247,822| 1,802|
| NOTA [72]            | –        | IN | 51.3  | 54.7  | 42.3  | 17.1| 35.4| 20,148| 252,531| 2,285|
| eHAF [53]            | –        | IN | 51.8  | 54.7  | 43.4  | 23.4| 37.9| 33,212| 236,772| 1,834|
| JBNOT [91]           | –        | IN | 52.6  | 50.8  | 41.3  | 19.7| 35.8| 31,572| 232,659| 3,050|
| TT [46]              | –        | IN | 54.9  | 63.1  | 48.4  | 24.4| 38.1| 20,236| 233,295| 1,088|
| MFPNTrack [50]       | –        | IN | 58.8  | 61.7  | 49.0  | 28.8| 33.5| 17,413| 213,594| 1,185|
| LPC_MOT [55]         | –        | IN | 59.0  | 66.8  | 51.5  | 29.9| 33.9| 23,102| 206,948| 1,122|
| Lif_T [51]           | –        | IN | 60.5  | 65.6  | 51.3  | 27.0| 33.6| 14,966| 206,619| 1,189|
| ApLift [52]          | –        | IN | 60.5  | 65.6  | 51.5  | 33.9| 30.9| 30,609| 190,670| 1,709|
| Online               |          |    |       |       |       |     |     |     |     |     |
| GMPHD [92]           | –        | IN | 39.6  | 36.6  | 30.3  | 8.8 | 43.3| 50,903| 284,228| 5,811|
| EAMT1 [73]           | ResNet50 | IN | 42.6  | 41.8  | 34.0  | 12.7| 42.7| 30,711| 288,474| 4,488|
| SORT1T [3]           | VGG16    | IN | 43.1  | 39.8  | 34.0  | 12.5| 42.3| 28,398| 238,582| 4,852|
| DMAN [74]            | ResNet50 | IN | 48.2  | 55.7  | 42.5  | 19.3| 38.3| 26,218| 263,608| 2,194|
| MOTDT17 [75]         | GoogLeNet| IN | 58.0  | 56.5  | 42.6  | 20.1| 37.0| 27,532| 246,924| 2,593|
| STRN [76]            | ResNet50 | IN | 50.9  | 57.5  | 42.3  | 19.3| 34.9| 25,937| 232,659| 1,802|
| DeepMOT-T [78]       | ResNet101| IN | 53.7  | 53.8  | 42.4  | 19.4| 36.6| 17,413| 234,477| 1,947|
| FAMNet [95]          | ResNet101| IN | 50.9  | 48.7  | 42.0  | 19.1| 33.4| 20,148| 234,477| 1,947|
| UMA [77]             | AlexNet  | IN | 53.1  | 54.4  | –     | 21.5| 31.8| 21,589| 236,534| 2,251|
| Tracktor++ [21]      | ResNet101| IN | 55.5  | 55.1  | 44.8  | 21.1| 35.3| 18,201| 224,047| 2,072|
| Tracktor++v2 [21]    | ResNet101| IN | 56.5  | 55.1  | 44.8  | 21.1| 35.3| 18,201| 224,047| 2,072|
| CenterTrack [24]1    | DLA [97] | IoU| 60.5  | 55.7  | –     | 26.4| 33.0| 11,599| 208,577| 2,540|
| TrackformerV1 [33]   | ResNet101| IN | 59.7  | 59.0  | –     | 29.6| 31.5| 30,724| 194,320| 2,579|
| TrackformerV3 [33]   | ResNet50 | IN | 62.3  | 57.6  | –     | 29.2| 27.1| 16,591| 192,123| 4,018|
| MO3TR (ours)         | ResNet50 | IN | 62.7  | 59.9  | 49.4  | 28.4| 32.3| 20,075| 182,860| 2,320|
| CenterTrack [24]     | DLA [97] | CD | 61.5  | 59.6  | 48.2  | 26.4| 31.9| 14,076| 200,672| 2,583|
| TrackformerV1 [33]   | ResNet101| IN | 61.8  | 59.8  | –     | 35.4| 21.1| 35,226| 177,270| 2,982|
| TrackformerV3 [33]   | ResNet50 | IN | 63.4  | 60.0  | –     | –   | –   | –    | –    | –    |
| MO3TR (ours)         | ResNet50 | IN | 63.2  | 60.2  | 49.6  | 31.9| 19.2| 21,966| 182,860| 2,841|

**Bold** and underlined numbers indicate best and second best result, respectively. Metrics that were not available for comparison are indicated via a hyphen (–). Public detections (PD) are processed in different ways by different methods based on their algorithmic structure: directly as input (IN), or used for filtering of track initialization proposals via the center distance (CD) as proposed in [24] or the intersection over union (IoU) as proposed in [33]. More detailed results of our approach are provided in the supplementary material, available online. 1 Result reported in [33].

4.3 Comparison With the State of the Art

**Public Detections.** We evaluate MO3TR on the challenging MOT16 and MOT17 benchmark test datasets [70] using the provided public detections and report our results in Tables 1 and 2, respectively. Despite not using any heuristic track management to filter or post-process, we achieve competitive performance across both benchmarks. We outperform most competing methods regarding several metrics, achieving new state of the art results for MOTA, FN, MT and ML metrics on MOT16. On MOT17, we set new benchmarks regarding MOTA, HOTA and ML when we incorporate the public detections via the intersection over union (IoU) as proposed in [33], and regarding IDF1, HOTA and ML when using the center distance incorporation proposed in [24].

As shown by its competitive IDF1 scores on both datasets, MO3TR is capable of identifying objects and maintaining their identities over long parts of the track, in many cases for more than 80% of the objects’ lifespans as evidenced by the comparably high MT and low ML results. We attribute this to MO3TR’s access to the track history through the temporal Transformers. This information allows our method to jointly reason over existing tracks, possible initiation and the input data via our spatial Transformers, a capability that helps to learn discerning occlusion from termination (Fig. 3), and thus to avoid false termination as is evidenced by the low FN and state of the art ML numbers achieved on both MOT datasets. These values further indicate that MO3TR learns to fill in gaps due missed detections or occlusions, which has additional influence on reducing FN and ML while increasing IDF1 and MT. Using its joint reasoning over the available information also helps MO3TR to reduce false track initiations (FN) considerably while keeping incorrect track initiations (FPs) at a reasonably low levels. The combination of superior IDF1, low FN and reasonable FP allows MO3TR to reach new state of the art MOTA results on both MOT16 (Table 1) and MOT17 (Table 2) datasets.

Comparing our results specifically to the concurrently developed method Trackformer [33], we achieve comparable or better results across all metrics while using a significantly smaller backbone then both Trackformer versions (cf. Table 2). Some of the improvements from TrackformerV1 towards its second version can be attributed to the switch to a multi-scale set detection concept [89], while we employ a single scale [30] in our method (similar to TrackformerV1). It is to be noted that Trackformer [33] exploits information only across two frames, while our MO3TR considers longer-term information. To make processing of our embedding sequences computationally tractable for any track length given constrained computational resources, we opted for a smaller backbone (ResNet50) for MO3TR. We expect our method to
yield further improvements with the incorporation of bigger backbones and multi-scale detection concepts like [89], however at higher computational costs.

Private Detections. To allow for broader comparability of our approach to other recent methods, we also evaluate MO3TR on the MOT17 benchmark using our method’s own private detections, achieving competitive results (cf. Table 3).

MO3TR With Predicted Initiation Queries. As introduced in Section 4.2, we additionally investigate how modifying the encoder of our approach as described (multi-scale backbone, additional auxiliary detection loss) and dynamically predicting the initiation queries $Z_Q$ conditioned on each input image instead of learning a fixed set of queries will influence the performance of our approach. We refer to this method as MO3TR-PIQ (Predicted Initiation Queries).

The results achieved on the MOT17 benchmark (Table 3) show significant gains in performance across all metrics, including 13.7% on MOTA and 10.4% on HOTA compared to our standard method – setting new state of the art results across severall metrics (MOTA of 77.6%). Evaluations conducted on the recently proposed and rather complex MOT20 benchmark underpin the performance capabilities by outperforming competing methods on several metrics (Table 4, please refer to the supplementary material, available online).

| Method                  | MOTA | IDF1 | HOTA | FP   | FN  | IDs |
|------------------------|------|------|------|------|-----|-----|
| TubeTK [98]            | 63.0 | 58.6 | 48.0 | 27,060 | 177,483 | 4,137 |
| CTracker [99]          | 66.6 | 57.4 | 52.2 | 22,284 | 160,491 | 5,529 |
| CenterTrack [24]       | 67.8 | 64.7 | 49.0 | 18,498 | 160,332 | 3,039 |
| TrackformerV2 [33]     | 65.0 | 63.9 | —    | 70,443 | 123,552 | 3,528 |
| TrackformerV3 [33]     | 74.1 | 68.0 | —    | 34,602 | 108,777 | 2,829 |
| TraDeS [58]            | 69.1 | 63.9 | 52.7 | 20,892 | 150,060 | 3,555 |
| PermaTrack [59]        | 73.8 | 68.9 | 55.5 | 28,998 | 115,104 | 3,699 |
| GSDDT-V2 [57]          | 73.2 | 66.5 | 55.2 | 26,397 | 120,666 | 3,891 |
| SOTMOT [61]            | 71.0 | 71.9 | —    | 39,537 | 118,983 | 5,189 |
| CorrTracker [62]       | 76.5 | 73.6 | —    | 29,808 | **99,510** | 3,369 |
| FairMOTv1 [56]         | 67.9 | 68.1 | 56.3 | 32,571 | 144,126 | 4,293 |
| FairMOTv2 [56]         | 73.7 | 72.3 | 59.3 | 27,507 | 117,477 | 3,030 |
| QDTrack [63]           | 68.7 | 66.3 | 53.9 | 26,859 | 146,643 | 3,378 |
| MO3TR (ours)           | 63.9 | 60.5 | 49.9 | 23,358 | 177,684 | 2,847 |
| MO3TR-PIQ (ours)       | **77.6** | 72.9 | **60.3** | 21,045 | 102,531 | 2,547 |

*Bold* and underlined numbers indicate best and second best result, respectively. More detailed results of our approach are provided in the supplementary material, available online.
for further results using public detections). We mainly attribute this improvement on the benchmarks to the fact that set-based detection methods like DETR [30] have just very recently been introduced to the community, and are still often significantly outperformed by non-set-based ones like [88]. While MO3TR is entirely set-based and built upon the detection concept of DETR, MO3TR-PIQ is incorporating the more powerful multi-scale detection setup of [88] into our tracking method – demonstrating potential for future combinations with other detection methods.

**Inference Speed versus Accuracy.** We provide details regarding the inference speed-accuracy comparison in Table 5. While center-based methods like CenterTrack [24] and Fair-MOT [56] seem to generally provide faster inference (possibly due to their point-based interpretation of the tracking problem), both our models show inference times that are comparable to the other two Transformer-based methods. While being comparable in performance (MOTA and IDF1), our base MO3TR method is notably faster than TrackformerV2 [33] (11.9 FPS versus 7.4 FPS). Our extended MO3TR-PiQ approach is slightly slower than the base version and is positioned directly between TrackformerV2/V3 [33] and TransTrack [34] in terms of inference speed while outperforming both in terms of MOTA and IDF1.

### 4.4 Ablation Studies

In this section, we evaluate different components of MO3TR on our validation set using private detections and show the individual contributions of the key components and strategies to facilitate learning.

**Effect of Track History Length.** The length of the track history describes the maximum number of embeddings from all the previous time steps of a certain identified object that our temporal Transformer has access to. To avoid overfitting to any particular history length that might be dominant in the dataset but not actually represent the most useful source of information, we specifically train our model with input track histories of varying and randomly chosen lengths. It is important to note that if the maximum track history length is set to one, the method practically degenerates to a two-frame based joint detection and tracking method such as Trackformer [33]. Our results reported in Table 6 however show that incorporating longer-term information is crucial to improve end-to-end tracking. Both MOTA and IDF1 can be consistently improved while FP can be reduced when longer term history, i.e., information from previous frames, is taken into account. This trend is also clearly visible throughout evaluation of our training strategies presented in Table 7, further discussed in the following.

**Training Strategies.** MOT datasets are highly imbalanced when it comes to the occurrence of initialization and termination examples compared to normal propagation, making it nearly impossible for models to naturally learn initiation or termination of no longer existing tracks when trained in a naive way. As presented in Table 7, naive training without any augmentation shows almost double the number of false negatives (FN) compared to augmented approaches, basically failing to initiate tracks properly. Augmenting with FN as discussed in 3.5 shows significant improvements for both two-frame and longer-term methods. Additionally right-aligning the track history helps generally to stabilize training and greatly reduces false positives. At last, augmenting with false positives is most challenging to implement but crucial. As the results

| Method                  | MOTA | IDF1 | FPS |
|-------------------------|------|------|-----|
| GSDT-V2 [57]            | 67.1 | 67.5 | 53.6 |
| OUTrack [100]           | 68.6 | 69.4 | 56.2 |
| CrowdTrack [101]        | 70.7 | 68.2 | 55.0 |
| FairMOTv1 [56]          | 58.7 | 63.7 | 65.2 |
| FairMOTv2 [56]          | 61.8 | 67.3 | 54.6 |
| SOTMOT [61]             | 68.6 | 71.4 | 57.0 |
| CorrTracker [62]        | 65.2 | 69.1 | 57.0 |
| TrackformerV3 [33]      | 68.6 | 65.7 | 57.0 |
| MO3TR-PIQ (ours)        | 72.3 | 69.0 | 57.3 |

| Method                  | MOTA | IDF1 | FPS |
|-------------------------|------|------|-----|
| MO3TR (ours)            | 77.6 | 72.9 | 8.8 |

**TABLE 6**

| len(T<sub>max</sub>) | MOTA | IDF1 | MT | ML |
|-----------------------|------|------|----|----|
| 1                     | 55.4 | 48.4 | 114| 19 |
| 10                    | 56.8 | 49.0 | 115| 18 |
| 20                    | 57.8 | 50.1 | 115| 19 |
| 30                    | 58.9 | 50.6 | 114| 20 |

_Ablation studies were performed using our validation set combining indoor and outdoor sequences of MOT20 [79] and 2DMOT15 [80] to cover a representative range of different pedestrian density levels._

| Method                  | MOTA | IDF1 | FP | FN |
|-------------------------|------|------|----|----|
| MO3TR-PIQ (ours)        | 77.6 | 72.9 | 8.8 |

**TABLE 4**

Results on the MOT20 Benchmark [79] Test Set Using private Detections

| Method                  | MOTA | IDF1 | HOTA | FP | FN |
|-------------------------|------|------|------|----|----|
| GSJD-T-V2 [57]          | 67.1 | 67.5 | 53.6 | 31,507 | 135,395 | 3,230 |
| OUTrack [100]           | 68.6 | 69.4 | 56.2 | 36,816 | 123,208 | 2,223 |
| CrowdTrack [101]        | 70.7 | 68.2 | 55.0 | 21,928 | 126,333 | 3,198 |
| FairMOTv1 [56]          | 58.7 | 63.7 | 65.2 | –   | –    | 6,013 |
| FairMOTv2 [56]          | 61.8 | 67.3 | 54.6 | 103,440 | 88,901 | 5,243 |
| SOTMOT [61]             | 68.6 | 71.4 | 57.0 | 101,154 | 95,855 | 4,209 |
| CorrTracker [62]        | 65.2 | 69.1 | 57.0 | –   | –    | 5,183 |
| TrackformerV3 [33]      | 68.6 | 65.7 | 57.0 | –   | –    | 1,532 |
| MO3TR-PIQ (ours)        | 72.3 | 69.0 | 57.3 | 12,738 | 128,439 | 2,200 |

_Bold and underlined numbers indicate best and second best result, respectively. Note that evaluation on the MOT20 benchmark does not differentiate between public and private detections (see use our private ones here). More detailed results of our approach are provided in the supplementary material, available online._

---

For more details, refer to the supplementary material available online.
To provide some insight (Two Frames versus 30) and Augmentation Strategies: False Negatives (FN), False Positives (FP), Right-Aligned Insertion (RA)

| Training Strategies | MOTA ↑ | IDF1 ↑ | FP ↓ | FN ↓ |
|---------------------|--------|--------|------|------|
| Naive (Two Frames)  | 12.2   | 22.1   | 7,905| 26,848|
| FN (Two Frames)     | 42.0   | 22,609 | 11,671|
| FN+RA (Two Frames)  | 42.5   | 22,609 | 11,671|
| FN+RA+FP (Two Frames)| 48.4   | 3,927  | 17,912|
| FN                  | 42.5   | 19,353 | 11,693|
| FN+RA               | 48.1   | 12,265 | 12,002|
| FN+RA+FP            | 50.6   | 3,471  | 12,692|

Ablation studies were performed using our validation set combining indoor and outdoor sequences of MOT20 [79] and 2DMOT15 [80] to cover a representative range of different pedestrian density levels.

![Temporal attention maps averaged over 100 randomly selected objects from the MOT20 dataset [79].](image)

Fig. 4. Temporal attention maps averaged over 100 randomly selected objects from the MOT20 dataset [79]. The vertical axis represents the maximum track history length, the horizontal axis the different embedding positions in the history. The displayed attention related the current query at time t to all the previous embeddings. Every row sums up to 1.

demonstrate, it significantly reduces false positives by helping the network to properly learn the terminating of tracks.

**Analysing Temporal Attention.** To provide some insight into the complex and highly non-linear working principle of our temporal Transformers, we visualize the attention weights over the temporal track history for different track history lengths averaged for 100 randomly picked objects in our validation set (Fig. 4). Results for the first layer clearly depict most attention being paid to multiple of its more recent frames, decreasing with increasing frame distance. The second and third layers are harder to interpret due to the increasing non-linearity, and the model starts to increasingly cast attention over more distant frames. It is important to notice that even if an embedding is not available at time $t - k$, the model can still choose to pay attention to that slot and use the non-existence for reasoning.

## 5 Conclusion

We presented MO3TR, a truly end-to-end multi-object tracking framework that uses temporal Transformers to encode the history of objects while employing spatial Transformers to encode the interaction between objects and the input data, allowing it to handle occlusions, track termination and initiation. Demonstrating the advantages of long term temporal learning, we set new state of the art results regarding multiple metrics on the popular MOT16, MOT17 and MOT20 benchmarks.

**Future Work.** It is known to the tracking community that most popular datasets and metrics overemphasize detection performance (see supplementary), available online. This problem has recently been tackled by the introduction of the new HOTA metric in [82], which is designed to better balance the contributions. While developing novel metrics is one valuable approach, we propose to potentially break with this tradition in the future and instead reconsider the data we use as basis to evaluate tracking. Creating data with significantly easier detection (medium/large object sizes) but much harder tracking tasks (complex movements including frequent occlusions and interactions) could help to better decouple the contributions of tracking approaches and motivate the community to focus on more diverse aspects of tracking beyond detection as well as broaden the access due to the reduced computational complexity.

## Acknowledgments

Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond and Ian Reid were associated with the Australian Centre for Robotic Vision.

## References

[1] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2011, pp. 1265–1272.
[2] A. Andriyenko, K. Schindler, and S. Roth, “Discrete-continuous optimization for multi-target tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 1926–1933.
[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in Proc. IEEE Int. Conf. Image Process., 2016, pp. 3464–3468.
[4] S. S. Blackman and R. Popoli, Design and Analysis of Modern Tracking Systems. Norwood, MA, USA: Artech House, 1999.
[5] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Sonar tracking of multiple targets using joint probabilistic data association,” IEEE J. Ocean. Eng., vol. 8, no. 3, pp. 173–184, Jul. 1983.
[6] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, “Joint probabilistic data association revisited,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 3047–3055.
[7] J. Smith, F. Particke, M. Hiller, and J. Thielecke, “Systematic analysis of the PMBM, PHD, JPD and GNN multi-target tracking filters,” in Proc. IEEE 22th Int. Conf. Inf. Fusion, 2019, pp. 1–8.
[8] R. L. Streit and T. E. Luginbuhl, “Maximum likelihood method for probabilistic multihypothesis tracking,” in Signal and Data Processing of Small Targets, vol. 2235. Bellingham, WA, USA: Int. Soc. Opt. Photonics, 1994.
[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[10] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6575–6583.
[11] J. Ren et al., “Accurate single stage detector using recurrent rolling convolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 752–760.
[12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
[13] F. Yang, W. Choi, and Y. Lin, “Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2129–2137.
[14] H-N. Hu et al., “Joint monocular 3D vehicle detection and tracking,” in Proc. IEEE/ICVF Int. Conf. Comput. Vis., 2019, pp. 5389–5398.
[15] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, “Learning by tracking: Siamese CNN for robust target association,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2016, pp. 418–425.
[16] A. Sadeghian, A. Alahi, and S. Savarese, “Tracking the untrackable: Learning to track multiple cues with long-term dependencies,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 300–311.
[17] Z. Wang, L. Zheng, Y. Liu, and S. Wang, “Towards real-time multi-object tracking,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 107–122.

[18] S. Wang, H. Sheng, Z. Zhang, Y. Wu, and Z. Xiong, “A general recurrent tracking framework without real data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 31399–31308.

[19] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, “Multiple hypothesis tracking and re-identification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 369–378.

[20] L. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.

[21] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Res. Logistics Quart., vol. 2, pp. 83–97, 1955.

[22] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Detect to track and track to detect,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 3057–3065.

[23] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, “Deep affinity access for multi-object tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 104–119, Jan. 2021.

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zhang, “DETR: A general multi-object tracking with single object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 3677–3686.

[25] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.

[26] R. E. Kalman, “A new approach to linear filtering and prediction problems,” Trans. ASME–J. Basic Eng., vol. 82, pp. 35–45, 1960.

[27] A. Vascwani et al., “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6001.

[28] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” 2021, arXiv:2101.01169.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zhang, “DETR: A general multi-object tracking with single object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 3677–3686.

[30] H. H. Kuhn, “The hungarian method for the assignment problem,” Naval Res. Logistics Quart., vol. 2, pp. 83–97, 1955.

[31] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.

[32] C. Feichtenhofer, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 369–378.

[33] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Res. Logistics Quart., vol. 2, pp. 83–97, 1955.

[34] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.

[35] C. Feichtenhofer, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 474–490.

[36] A. Hornakova, T. Kaiser, P. Svoboda, M. Rolinek, B. Rosenhahn, and P. Svoboda, “Lifted disjoint paths with application in multi-object tracking,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 4364–4375.

[37] A. Hornakova, T. Kaiser, P. Svoboda, M. Rolinek, B. Rosenhahn, and P. Svoboda, “Lifted disjoint paths with application in multi-object tracking,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 4364–4375.

[38] C.-H. Kuo and R. Nevatia, “How does person identity recognition help multi-person tracking?,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1217–1224.

[39] F. Particke, M. Hiller, L. Patino-Studencki, J. Thielecke, C. Sippl, and C. Feist, “Improvements in pedestrian movement by a generalized potential field approach,” in Mobil. TUM, Intell. Transport. System, vol. 14, 2010, pp. 389–404.

[40] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Res. Logistics Quart., vol. 2, pp. 83–97, 1955.

[41] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.

[42] C. Feichtenhofer, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 474–490.

[43] F. Particke, M. Hiller, C. Feist, and J. Thielecke, “Improvements in pedestrian movement prediction by considering multiple intentions in a multi-hypotheses filter,” in Proc. IEEE/ION Position, Location Navigation Symp., 2018, pp. 209–212.

[44] K. Fang, Y. Xiang, X. Li, and S. Savarese, “Recurrent autoregressive networks for multi-object tracking,” in Proc. IEEE Winter Conf. Comput. Vis. Pattern Recognit., 2018, pp. 468–475.

[45] Y. Liang and Y. Zhou, “LSTM multiple object tracker combining multiple cues,” in Proc. IEEE Int. Conf. Image Process., 2018, pp. 2351–2355.

[46] Y. Zhang et al., “Long-term tracking with deep tracklet association,” IEEE Trans. Image Process., vol. 29, pp. 6694–6706, May 2020.

[47] F. Saleh, S. Aliakbarian, H. Rezaotafghi, M. Salzmann, and S. Gould, “Probabilistic tracklet scoring and inpainting for multi-object tracking,” 2020, arXiv:2012.02337.

[48] D. Stadler and J. Beyerer, “Improving multiple pedestrian tracking by track management and occlusion handling,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 10953–10962.

[49] T. Meinhardt, C. Feichtenhofer, and J. Thielecke, “Improvements in pedestrian movement by a generalized potential field approach,” in Proc. Sensor Data Fusion: Trends, Solutions, 2021, pp. 13199–13208.

[50] ZHU ET AL.: LOOKING BEYOND TWO FRAMES: END-TO-END MULTI-OBJECT TRACKING USING SPATIAL AND TEMPORAL... 12795

[51] F. Particke, M. Hiller, L. Patino-Studencki, C. Sippl, C. Feist, and J. Thielecke, “Multiple intention tracking by a generalized potential field approach,” in Proc. Sensor Data Fusion: Trends, Solutions, 2017, pp. 1–5.

[52] F. Particke, M. Hiller, L. Patino-Studencki, C. Sippl, C. Feist, and J. Thielecke, “Multiple intention tracking by a generalized potential field approach,” in Proc. Sensor Data Fusion: Trends, Solutions, 2017, pp. 1–5.

[53] F. Particke, M. Hiller, C. Feist, and J. Thielecke, “Improvements in pedestrian movement prediction by considering multiple intentions in a multi-hypotheses filter,” in Proc. IEEE/ION Position, Location Navigation Symp., 2018, pp. 209–212.

[54] K. Fang, Y. Xiang, X. Li, and S. Savarese, “Recurrent autoregressive networks for multi-object tracking,” in Proc. IEEE Winter Conf. Comput. Vis. Pattern Recognit., 2018, pp. 468–475.

[55] Y. Liang and Y. Zhou, “LSTM multiple object tracker combining multiple cues,” in Proc. IEEE Int. Conf. Image Process., 2018, pp. 2351–2355.

[56] Y. Zhang et al., “Long-term tracking with deep tracklet association,” IEEE Trans. Image Process., vol. 29, pp. 6694–6706, May 2020.

[57] F. Saleh, S. Aliakbarian, H. Rezaotafghi, M. Salzmann, and S. Gould, “Probabilistic tracklet scoring and inpainting for multi-object tracking,” 2020, arXiv:2012.02337.
P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shi, “Stand-alone self-attention in vision models,” in Proc. Adv. Neural Inf. Process. Syst., 2019, Art. no. 7.

A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” Int. Conf. Learn. Representations, 2021, pp. 1–21.

H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savaresi, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 658–666.

K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” EURASIP J. Image Video Process., vol. 2008, 2008, Art. no. 1.

T. A. P. Milan, L. Leal-Taixe, I. Reid, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” 2016, arXiv:1603.00831.

Y. Tian, A. Dehghan, and M. Shah, “On detection, data association and segmentation for multi-target tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 9, pp. 2146–2160, Sep. 2019.

L. Chen, H. Ai, R. Chen, and Z. Zhuang, “Aggregate tracklet appearance features for multi-object tracking,” IEEE Signal Process. Lett., vol. 26, no. 11, pp. 1613–1617, Nov. 2019.

R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, “Online multi-target tracking with strong and weak detections,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 84–99.

J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, Online multi-object tracking with dual matching attention modules,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 379–396.

C. Long, A. Haizhou, Z. Zijie, and S. Chong, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification,” in Proc. IEEE Int. Conf. Multimedia Expo, 2018, pp. 1–6.

J. Xu, Y. Cao, Z. Zhang, and H. Hu, “Spatial-temporal relational networks for multi-object tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 3987–3997.

J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, “A unified object motion and affinity model for online multi-object tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6677–6676.

Y. Xu, A. Osep, Y. Ban, R. Horaud, L. Leal-Taixé, and X. Alameda-Pineda, “How to train your deep multi-object tracker,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6796–6795.

P. Dendorfer et al., “MOT20: A benchmark for multi object tracking in crowded scenes,” 2020, arXiv:2003.09003.

L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, “MOTChallenge 2015: Towards a benchmark for multi-target tracking,” 2015, arXiv:1504.01942.

F. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 17–35.

J. Luiten et al., “HOTA: A higher order metric for evaluating multi-object tracking,” Int. J. Comput. Vis., vol. 129, pp. 548–578, 2021.

T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

S. Shao et al., “CrowdHuman: A benchmark for detecting human in a crowd,” 2018, arXiv:1805.00123.

A. Ess, B. Leibe, K. Schindler, and L. Van Gool, “A mobile vision system for robust multi-person tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2008, pp. 1–8.

T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint detection and identification feature learning for person search,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3376–3385.

Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in Proc. IJCAI, Learn. Representations, 2020, pp. 145–161.

M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, “Motion segmentation & multiple object tracking by correlation co-clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 1, pp. 140–153, Jan. 2020.

R. Henschel, Y. Zou, and B. Rosenhahn, “Multiple people tracking using body and joint detections,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2019, pp. 770–779.

T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora, “Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill., 2017, pp. 1–5.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

P. Chu and H. Ling, “Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6171–6180.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

F. Yu, D. Wang, E. Shellhammer, and T. Darrell, “Deep layer aggregation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 2403–2412.

B. Pang, Y. Li, Y. Zhang, M. Li, and C. T. Lu, “Adopting tubes to track multi-object in a one-step training model,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6307–6317.

J. Peng et al., “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” 2021, arXiv:2105.07945.

Q. Liu et al., “Online multi-object tracking with unsupervised re-identification learning and occlusion estimation,” Neurocomputing, vol. 483, pp. 333–347, 2022.

D. Stadler and J. Beyerer, “On the performance of crowd-specific detectors in multi-pedestrian tracking,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Survell., 2021, pp. 1–12.

Tianyu Zhu received the bachelor of engineering and science degree from Monash University with first class Honours, in 2017. He is currently working toward the PhD degree in computer vision with Monash University since 2018. He was interning as an applied scientist with Amazon, in 2022. His current research interests are fully learnable multi object tracking frameworks and weakly-supervised learning.

Markus Hiller received the bachelor of science and master of science degrees from FAU, Germany, in 2015 and 2017, respectively. He is currently working towards the PhD degree with the School of Computing and Information Systems, University of Melbourne, Australia. Prior to this, he has worked for 2.5 years on a research project studying environment perception and modelling for mobile robots. His current research interests include representation learning, learning from limited data and under distribution shifts, geometry, as well as various applications in computer vision.

Maha Ehsanpour is currently working toward the PhD degree in computer science with The University of Adelaide and the Australian Institute for Machine Learning (AIML) since 2018. Her research interest lies with the intersection of deep learning applied to computer vision applications with a specific focus on understanding humans in motion within video sequences. More specifically, her research encompasses fine-grained human understanding such as human pose, body surface and trajectory prediction as well as coarse-grained human understanding such as comprehending human actions, interactions with other individuals (social groups) and social activities.
Rongkai Ma received the bachelor’s degree in electrical and computer systems engineering (ECSE) from Monash University. He is currently working toward the PhD degree with Monash University and Australian Research Council Centre of Excellence for Robotic Vision since 2019. His research interest focuses on learning adaptively with few observations for computer vision applications.

Tom Drummond is the Melbourne Connect chair of Digital Innovation for Society with the University of Melbourne. Between 2010 and 2021, he was a professor of engineering and from 2016, the head of the Department of Electrical and Computer Systems Engineering, Monash University. From 2001 to 2010, he was a lecturer in engineering with the University of Cambridge. His research interests include robotic vision, augmented reality, machine learning and efficient algorithms. He has published more than 150 peer reviewed papers and serves on the program committees of numerous national and international conferences and was an area editor for IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010-2017.

Ian Reid is the head of the School of Computer Science, University of Adelaide. He is a fellow of Australian Academy of Science and fellow of the Academy of Technological Sciences and Engineering, and held an ARC Australian Laureate Fellowship 2013-2018. Between 2000 and 2012 he was a professor of engineering science with the University of Oxford. His research interests include robotic vision, SLAM, visual scene understanding, and human motion analysis. He serves on the program committees of various national and international conferences, and was the general chair of the International Symposium of Mixed and Augmented Reality ISMAR, in 2008.

Hamid Rezatofighi received the PhD degree from Australian National University, in 2015. He is a lecturer with the Faculty of Information Technology, Monash University, Australia. Before that, he was an Endeavour research fellow with the Stanford Vision Lab (SVL), Stanford University, and a senior research fellow with the Australian Institute for Machine Learning (AIML), the University of Adelaide. He has published more than 60 top tier papers in computer vision, AI and machine learning, robotics, and has been involved as primary investigator in several successful grant fundings (totally over $11.0 M), including two DARPA grants and one ARC discovery grant. He served as the publication chair in ACCV 2018 and as the area chair in WACV 2021, CVPR 2020 and CVPR 2022-2023. His research interests include computer vision, machine learning and robotic vision, esp. the visual perception problems that are required for an autonomous robot to navigate or interact in a human crowded environment.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.