Multiple Object Tracking as ID Prediction

Ruopeng Gao[1] Ji Qi[2] Limin Wang[1,3,✉]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]China Mobile (Suzhou) Software Technology Co., Ltd. [3]Shanghai AI Lab
ruopenggao@gmail.com, lmwang@nju.edu.cn

Abstract

*Multi-Object Tracking (MOT) has been a long-standing challenge in video understanding. A natural and intuitive approach is to split this task into two parts: object detection and association. Most mainstream methods employ meticulously crafted heuristic techniques to maintain trajectory information and compute cost matrices for object matching. Although these methods can achieve notable tracking performance, they often require a series of elaborate handcrafted modifications while facing complicated scenarios. We believe that manually assumed priors limit the method's adaptability and flexibility in learning optimal tracking capabilities from domain-specific data. Therefore, we introduce a new perspective that treats **M**ultiple **O**bject **T**racking as an in-context **I**D **P**rediction task, transforming the aforementioned object association into an end-to-end trainable task. Based on this, we propose a simple yet effective method termed **MOTIP**. Given a set of trajectories carried with ID information, MOTIP directly decodes the ID labels for current detections to accomplish the association process. Without using tailored or sophisticated architectures, our method achieves state-of-the-art results across multiple benchmarks by solely leveraging object-level features as tracking cues. The simplicity and impressive results of MOTIP leave substantial room for future advancements, thereby making it a promising baseline for subsequent research. Our code and checkpoints are released at https://github.com/MCG-NJU/MOTIP.*

1. Introduction

The objective of multiple object tracking (MOT) is to accurately locate all objects of interest within a video stream while consistently maintaining their respective identities throughout the sequence. As an essential problem in computer vision, it is crucial for many downstream tasks, such as action recognition [9] and trajectory prediction [24]. In practical applications, it has also played a substantial role in various fields, including autonomous driving [66], sports event analysis [10, 48, 58], animal behavior research [71], and so on. Consequently, the challenges and advancements in multiple object tracking (MOT) have long garnered attention from the community.

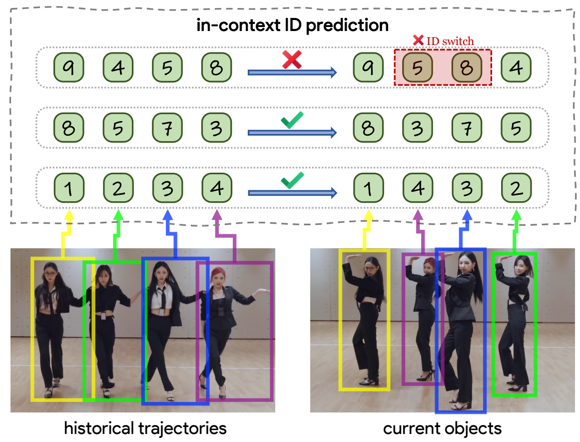

Figure 1. Diagram of the in-context ID prediction process. Different colored bounding boxes represent targets corresponding to different trajectories. We provide two valid ID prediction results, shown in the two lines below. This indicates that each trajectory only needs to predict the corresponding label based on the historical ID information, rather than being assigned a fixed label.

In the early stages of research within multi-object tracking area, the application scenarios and benchmarks were largely concentrated on pedestrian tracking [26, 40]. In this scenario, the characteristics of pedestrians are primarily linear motion and distinguishable appearance. Therefore, at that time, the methods [3, 69] predominantly relied on the Kalman filter [55] to model trajectories and predict their locations in the current frame, subsequently employing manually-designed algorithms for target matching. Subsequent research [54, 56, 68] introduced additional re-identification modules to compute the similarity

✉: Corresponding author.

between trajectories and current objects, aiding in resolving long-term occlusions that are challenging for linear motion estimations. Despite achieving notable success in pedestrian tracking, these methods have struggled to keep up with the emergence of increasingly complex tracking scenarios [10, 52, 71]. In these scenarios, irregular movements and similar appearances deviate from heuristic priors, reducing the effectiveness of fixed matching rules then weakening tracking performance. Although some of the latest heuristic algorithms [21, 36, 63] can gradually adapt to these cases, the compromise is that each improvement requires substantial human analysis as well as meticulous tuning of rules and hyperparameters.

In recent years, some scholars have proposed end-to-end trainable MOT methods [4, 39, 60, 67, 72] that directly learn tracking capabilities from the given training data to pursue the optimal solution. Among these, the methods [16, 39, 49, 51, 67] based on extending DETR [6, 30, 74] to MOT have garnered significant attention and research. They propagate track queries across video frames to represent different trajectories. Despite achieving impressive results on multiple benchmarks, particularly on some highly challenging ones, these methods still leave some concerning issues. The most notable is that simultaneously using different types of queries for detection and tracking can cause conflicts within the unified decoding process [61, 65], thus impairing either detection or tracking performance. Some studies [70] have found that using an additional independent detector [18] to decouple multi-object tracking can effectively mitigate this issue. Other studies [61, 65] have also shown that handling detection and tracking within the same module can lead to conflicts in the allocation of supervision signals. Reflecting on the above, a natural question arises: *Can we maintain the decoupling nature of the multi-object tracking problem while discarding heuristic algorithms in favor of an end-to-end pipeline to fully unleash the model's potential?*

Since there are already many mature end-to-end frameworks for object detection [6, 18, 74], we primarily focus on the formulation of object association. Intuitively, it resembles a classification problem, as different trajectories are annotated with distinct labels. However, considering the generalization to unseen trajectories during inference, *i.e.*, new ID labels, classification prediction cannot be directly applied to object association. This is why, despite some ReID-based methods [68] using label classification for supervision, cosine similarity is employed during inference to calculate the affinity matrix, thereby determining the association results. We reflect on this generalization dilemma, which arises because the labels of trajectories differ from traditional classification tasks [46]. Although a unique number annotates each trajectory as its ID, this does not imply that it can only be represented as this label. On the contrary, it is considered acceptable as long as a trajectory is predicted with the same ID label at all time steps. Therefore, we consider treating the object association problem as an in-context ID prediction problem, as illustrated in Fig. 1. Specifically, for a target in the current frame, we only need to predict its ID label based on the ID information carried by the corresponding historical trajectory, rather than predicting a globally fixed label as in traditional classification tasks. This ensures the generalization ability while facing unseen identities during inference. In this way, the target association is formulated into a novel framework, maintaining consistency and end-to-end in both training and inference.

Based on the perspective above, we propose our method, **MOTIP**, by treating **M**ultiple **O**bject **T**racking as an **I**D **P**rediction problem. Specifically, we opted for Deformable DETR [74] as our detector because it can directly provide object-level embeddings while detecting targets, without the need to consider various feature extraction techniques such as RoI, hierarchical structures, or feature pooling. To represent the identity information for each trajectory, we store a set of learnable ID embeddings, which are attached to specific trajectory tokens as needed. As for the crucial ID prediction module, we simply use a standard transformer decoder [53], composed of multiple layers of alternating self-attention and cross-attention. Despite our minimalist and straightforward design, without employing tailored and sophisticated network structures, it demonstrates impressive state-of-the-art tracking performance across multiple benchmarks. Therefore, we believe that framing multi-object tracking as an ID prediction problem still holds significant untapped potential, which can be further explored in future research.

2. Related Work

Tracking-by-Detection is the most widely used paradigm for multi-object tracking in the community. These methods [3, 47, 69] employ post-processing strategies to associate detection results with historical trajectories, thereby achieving online multiple object tracking frame by frame. Most of them [3, 69] rely on Kalman filter [55] to handle linear pedestrian motion [19, 26, 40] and leverage ReID features [7, 17, 36, 37, 54, 56, 68] to incorporate object appearance cues. In recent years, many methods [1, 5, 8, 12, 12, 14, 15, 21, 32, 38, 50, 62–64] have adopted more complicated modeling and matching approaches or introduced additional multimodal information to mitigate the limitations of manual algorithms in complex scenarios [10, 52]. Our proposed MOTIP also structurally decouples detection and association, but it relies on learnable models rather than heuristic algorithms. While some modern approaches [22, 34, 35, 43, 59] also utilize learnable modules to capture motion patterns, they still depend on handcrafted decisions to accomplish object association. In contrast, our method incorporates the decision-making process into the end-to-end pipeline, which significantly highlights the uniqueness of our approach.

Tracking-by-Propagation is a recently popular end-to-end multiple object tracking paradigm. Inspired by query-based detection models [6, 30, 74], they [39, 67] extended the detect queries to the tracking task, using track queries to represent tracked targets and propagate them through the video sequence. There are also some methods [4, 25] that employ tailored query-based model variants. Due to the flexibility of end-to-end trainable models, they are more adaptable to tracking in complex scenarios [52]. Subsequent works [4, 16, 49] have focused on long-term modeling, further enhancing the training performance. Nevertheless, some studies [61, 65, 70] have pointed out irreconcilable conflicts in the process of joint detection and tracking and attempted to alleviate this issue. Although our MOTIP also utilizes a query-based detection model [74], we do not fall into this paradigm because our object detection and association are performed sequentially in two separate modules, rather than simultaneously.

3. Method

In this section, we detail our proposed method, MOTIP, which treats multiple object tracking as an in-context ID prediction task. Firstly, in Sec. 3.1, we introduce a novel perspective on how to formulate object association in MOT as an ID prediction problem. Subsequently, in Sec. 3.2, we provide a detailed explanation of each component within MOTIP. Finally, in Sec. 3.3 and Sec. 3.4, we further illustrate the processes of training and inference.

3.1. In-context ID Prediction

In multiple object tracking data, different trajectories are annotated with distinct ID labels. Therefore, some works [68] adopt classification loss to directly supervise the model in distinguishing different identities. However, during inference, the model will encounter unseen trajectories, which means it needs to predict out-of-distributed labels, leading to generalization issues. As a result, additional post-processing steps must be employed to complete the inference, such as using cosine similarity to determine object-matching results.

Upon deep reflection, we believe this is due to the difference between ID labels in MOT and traditional classification tasks [46]. In MOT, the labels of the trajectories are actually used to indicate a certain consistency rather than specific semantic information. In other words, for a trajectory, as long as the ID label remains consistent across each frame, it is acceptable and does not need a specific label. For example, in Fig. 1, the ID labels of these four objects are marked as *1 2 3 4* in the ground truth file. However, we can also use *8 5 7 3* to represent them. As long as the labels remain consistent in subsequent frames (as shown on the right), it will be considered a correct result.

Based on the above analysis, MOT can be regarded as a special label prediction problem where target labels are determined by historical trajectory identity information. Let $\mathcal{T}_{t-1}$ represent the historical trajectories, where $\mathcal{T}_{t-1} = \{\mathcal{T}_{t-1}^1, \mathcal{T}_{t-1}^2, \cdots, \mathcal{T}_{t-1}^M\}$ with each $\mathcal{T}_{t-1}^m$ representing a trajectory with a consistent identity. Simultaneously, we randomly assign an ID label k_m to each trajectory $\mathcal{T}_{t-1}^m$ ensuring $1 \leq k_m \leq K$. When a new frame I_t is input, for any detected object o_t, if it belongs to the m-th trajectory $\mathcal{T}^m$, its correct ID label prediction result should be k_m. Since this prediction objective is based on the identity information k attached to the historical trajectories, we refer to it as *in-context ID prediction*, where the assigned label k serves as an in-context prompt. In Fig. 1, we have provided some examples. Due to this formulation, the ID prediction results for any unseen trajectories will remain within the distribution of the training procedure, *i.e.*, $1 \leq k \leq K$, thereby addressing the generalization dilemma.

3.2. MOTIP Architecture

The overall architecture of MOTIP is surprisingly simple, as shown in Fig. 2. It contains three main components, which we will detail below: *a DETR [74] detector* to detect objects and extract their object-level features, *a learnable ID dictionary* to represent different in-context identity information, and *an ID Decoder* to predict ID labels based on historical trajectories.

DETR Detector. We use Deformable DETR [74], an end-to-end object detection model, as our detector. Starting from an input image I_t, the CNN [20] backbone and transformer encoder extract and enhance the image features. Subsequently, the transformer decoder generates the output embeddings from learnable detect queries. They are decoded into bounding boxes and classification confidence by the *bbox* and *cls* head, as illustrated in Fig. 2. This approach further simplifies our method, as we can directly use the decoded output embedding as the target feature f_t^n, eliminating the need for complicated feature extraction techniques such as RoI, hierarchical methods, *etc.*

ID Dictionary. As discussed in Sec. 3.1, for the model's generalization on unseen trajectories, we require additional signifiers to represent the identity information of trajectories, which are used as in-context prompts. Since identity is discrete information, a naïve approach would be to use one-hot encoding. However, we believe this is not a good idea. Firstly, one-hot encoding is not conducive to neural network training. Secondly, this encoding scheme limits the number of ID labels to the vector dimensions that the model can handle, which is unfavorable for subsequent expansion and generalization. Therefore, we create an ID dictionary $\mathcal{I}$ that consists of $K+1$ learnable words to represent different

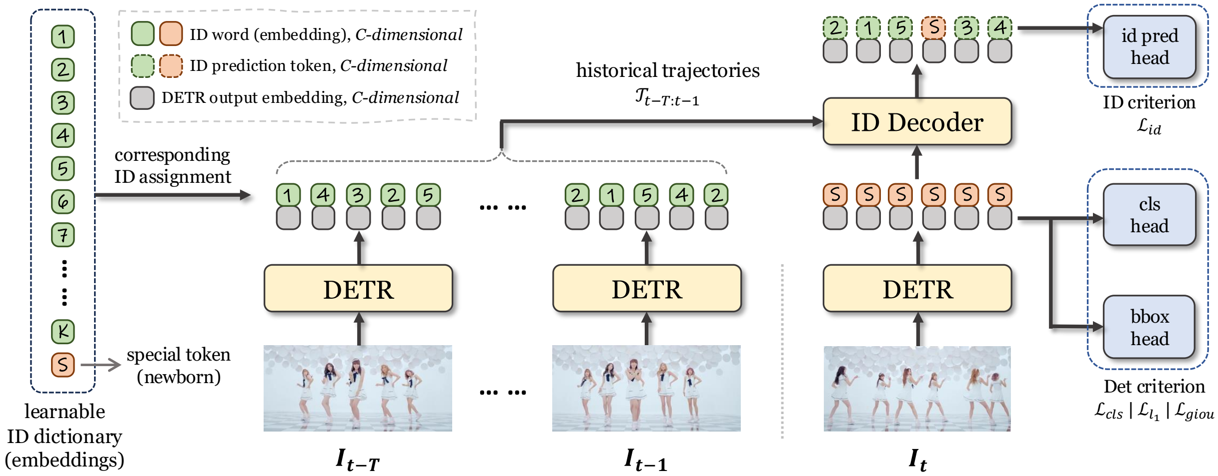

Figure 2. **Overview of MOTIP.** There are three primary components: a DETR detector detects objects, a learnable ID dictionary represents different identities, and an ID Decoder predicts the ID labels of current objects, as we detailed in Sec. 3.2. We combine object features with their corresponding ID embeddings to form the historical trajectories $\mathcal{T}_{t-T:t-1}$. Subsequently, the ID tokens are regarded as identity prompts, and the ID Decoder performs in-context ID prediction based on them, as discussed in Sec. 3.1 and Sec. 3.2.

identities, as follows:

$$\mathcal{I} = \{i^1, i^2, \cdots, i^K, i^{spec}\}, \tag{1}$$

where each word i^k is a learnable C-dimensional embedding. In detail, the first K tokens $\{i^1, i^2, \cdots, i^K\}$ are regular tokens that represent specific identities, while the last word i^{spec} is a *special* token that stands for newborn objects.

Tracklet Formation. In MOTIP, we only use object-level features as tracking cues. Therefore, for the m-th trajectory, we retain all target features from the past T frames, denoted as $\mathcal{F}_{t-T:t-1} = \{f^m_{t-T}, \cdots, f^m_{t-1}\}$, and randomly assign a unique ID label k_m. Then, we fuse the corresponding ID words i^{k_m} with the target features f^m_t, so that the tracklets carry both tracking cues and in-context identity prompts needed for ID prediction, as discussed in Sec. 3.1. Here, we simply use concatenation to achieve this, as shown below:

$$\tau^{m,k_m}_t = concat(f^m_t, i^{k_m}). \tag{2}$$

Here, f^m_t is the C-dimensional output embedding from DETR, and i^{k_m} is a C-dimensional token obtained from the dictionary Eq. (1). Ultimately, this results in a $2C$-dimensional tracklet representation τ^{m,k_m}_t. According to this, we denote all historical trajectories as $\mathcal{T}_{t-T:t-1} = \{\cdots, \mathcal{T}^m_{t-T:t-1}, \cdots\}$, where $\mathcal{T}^m_{t-T:t-1} = \{\tau^{m,k_m}_{t-T}, \cdots, \tau^{m,k_m}_{t-1}\}$. For the sake of consistency, we apply the same construction form from Eq. (2) to the targets in the current frame. However, since there is no trajectory identity yet, we use the special token i^{spec} instead of i^{k_m}, denoted as $\tau^n_t = concat(f^n_t, i^{spec})$.

ID Decoder. Due to the variable length and number of historical trajectories, we use a standard transformer decoder structure [53] as our ID Decoder to handle the variable-length inputs. This component uses all historical tracklets τ^{m,k_m} as *Key* and *Value* to decode all active detection tracklets τ^n_t in the current frame, as illustrated in Fig. 2. We use a simple linear classification head to predict the ID label for the decoded output embeddings. If a detection τ^n_t belongs to the m-th trajectory, the classification head should predict it as k_m, since i^{k_m} corresponds to the in-context ID information for that trajectory. This way, the entire object association process can be formulated as a classification task, allowing for direct supervision using cross-entropy loss.

3.3. Training

Loss Function. As previously discussed, we transform the object association in MOT into an end-to-end learnable $K+1$ classification problem through in-context ID prediction. Consequently, we can use the standard cross-entropy loss function as supervision, denoted as $\mathcal{L}_{id}$. Since DETR [6, 74] can also be trained end-to-end, the entire MOTIP model can utilize a unified loss function $\mathcal{L}$ for supervision:

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{giou}\mathcal{L}_{giou} + \lambda_{id}\mathcal{L}_{id}, \tag{3}$$

where $\mathcal{L}_{cls}$ is the focal loss [29]. $\mathcal{L}_{L1}$ and $\mathcal{L}_{giou}$ denote the L1 loss and the generalized IoU loss [44], respectively. λ_{cls}, λ_{L1} and λ_{giou} are their corresponding weight coefficients, and λ_{id} is the weight coefficient of ID loss $\mathcal{L}_{id}$.

Trajectory Augmentation. In multiple object tracking, we often face numerous challenges, such as target occlusion,

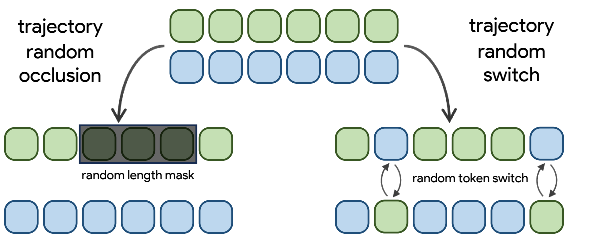

Figure 3. Illustration of trajectory augmentation: *trajectory random occlusion* (left) and *trajectory random switch* (right). Two different colors represent two distinct trajectories.

blurriness, and high similarity between targets. This can potentially lead to partial errors in ID assignment during online inference, which in turn can reduce the reliability of historical trajectories in subsequent processes. However, such errors do not occur during training because we use ground truth for supervision and trajectory construction. We argue that the oversimplification during training may prevent the model from acquiring sufficiently generalized and robust tracking capabilities. To mitigate this issue, we propose two trajectory augmentation techniques to be used during the training phase. Firstly, considering that occlusion is a challenging problem faced by MOT, we randomly drop tokens from each trajectory with a probability of λ_{occ}, as shown on the left of Fig. 3. Secondly, considering the potential ID assignment errors during inference, we randomly swap the ID tokens of two trajectories within the same frame with a probability of λ_{sw} to simulate the model assigning incorrect IDs to similar targets, as shown on the right of Fig. 3.

3.4. Inference

As discussed in Sec. 3.1 and Sec. 3.2, during the inference stage, we can randomly assign an ID label k_m to each trajectory $\mathcal{T}^m$ and use the corresponding ID embedding i^{k_m}, as long as the labels are unique across different trajectories. *I.e.*, for any two trajectories $\mathcal{T}^m$ and $\mathcal{T}^n$, $k_m \neq k_n$. In the implementation, we sequentially assign ID labels from 1 to K to represent the trajectories. For some longer video sequences, as trajectories expire and new ones appear, there may be more than K trajectories. To address this, we recycle the ID labels of the concluded trajectories for reuse.

In practice, for all output embeddings decoded by DETR, we first filter them using a detection confidence threshold λ_{det}. After that, all active detections are fed into the ID Decoder to predict the probability of each ID label. Similar to traditional classification tasks [20, 46], for each object, we select the ID with the highest probability ($> \lambda_{\text{id}}$) as the final result. Subsequently, if an object is not assigned a valid ID and its detection confidence is greater than λ_{new}, it will be marked as a newborn target and assigned a new identity. This makes our inference process very simple and straightforward. Notably, due to the restriction against duplicate ID predictions in the evaluation process of MOT tasks, when two targets in the same frame are predicted to the same ID label, only the one with the highest confidence score is retained. The pseudocode and additional details are provided in Appendix B.2.

From past experience, more complex or advanced ID assignment strategies, such as Hungarian algorithm or multi-stage matching [69], might offer some improvements. However, to validate the robustness and generalization of the model itself, we do not focus on these approaches.

4. Experiments

4.1. Datasets and Metrics

Datasets. To evaluate MOTIP, we select a variety of challenging benchmarks. DanceTrack [52] is a multi-person tracking dataset composed of 100 videos of various types of group dances. SportsMOT [10] is a dataset focused on athlete tracking, composed of 240 sports broadcast videos. BFT [71] is a high-maneuverability target tracking dataset that includes 22 bird species from around the world, consisting of 106 video clips. These benchmarks feature numerous serious challenges commonly face in multi-object tracking, such as frequent occlusions, irregular movements, high-speed motion, and similar appearance. This will help us fully verify the robustness and generalization ability of MOTIP in different scenarios.

Metrics. We mainly use the Higher Order Tracking Accuracy (HOTA) [33] to evaluate our method since it provides a balanced way to measure both object detection accuracy (DetA) and association accuracy (AssA). We also list the MOTA [2] and IDF1 [45] metrics in our experiments.

4.2. Implementation Details

Network. In practice, we select Deformable DETR [74] with a ResNet-50 [20] backbone as our default DETR detector because it is a versatile option for downstream tasks [42, 67]. Similar to previous work [16, 61, 65], we also utilize the COCO [28] pre-trained weights as initialization. We apply relative position encoding in the ID Decoder because tracking focuses more on relative temporal relationships rather than absolute timestamps. To minimize unnecessary additional modules, the hidden dimension throughout the entire model is kept consistent with Deformable DETR, which is $C = 256$. Since the ID dictionary can be reused, it is only necessary to ensure that K is not less than the maximum number of targets per frame. Here, we set K to 50 for simplicity.

Training. As in the prior work [16, 70], we use several common data augmentation methods, such as random resize, crop, and color jitter. The shorter and longer side of the input image is resized to 800 and 1440, respectively.

In each training iteration, we randomly sample $T + 1$

Methods	HOTA	DetA	AssA	MOTA	IDF1
w/o extra data:					
FairMOT [68]	39.7	66.7	23.8	82.2	40.8
CenterTrack [72]	41.8	78.1	22.6	86.8	35.7
TraDeS [57]	43.3	74.5	25.4	86.2	41.2
TransTrack [51]	45.5	75.9	27.5	88.4	45.2
ByteTrack [69]	47.7	71.0	32.1	89.6	53.9
GTR [73]	48.0	72.5	31.9	84.7	50.3
QDTrack [41]	54.2	80.1	36.8	87.7	50.4
MOTR [67]	54.2	73.5	40.2	79.7	51.5
OC-SORT [5]	55.1	80.3	38.3	92.0	54.6
StrongSORT [13]	55.6	80.7	38.6	91.1	55.2
C-BIoU [62]	60.6	81.3	45.4	91.6	61.6
Hybrid-SORT [63]	62.2	/	/	91.6	63.0
DiffMOT [35]	62.3	**82.5**	47.2	**92.8**	63.0
MeMOTR [16]	63.4	77.0	52.3	85.4	65.5
CO-MOT [61]	65.3	80.1	53.5	89.3	66.5
MOTIP (ours)	**69.6**	80.4	**60.4**	90.6	**74.7**
with extra data:					
MOTRv3 [65]	68.3	/	/	91.7	70.1
CO-MOT [61]	69.4	82.1	58.9	91.2	71.9
MOTRv2 [70]	69.9	**83.0**	59.0	**91.9**	71.7
MOTIP (ours)	**72.0**	81.8	**63.5**	**91.9**	**76.8**

Table 1. Performace comparison with state-of-the-art methods on the DanceTrack [52] test set. The best result is shown in **bold**.

Methods	HOTA	DetA	AssA	MOTA	IDF1
w/o extra data:					
FairMOT [68]	49.3	70.2	34.7	86.4	53.5
QDTrack [41]	60.4	77.5	47.2	90.1	62.3
ByteTrack [69]	62.1	76.5	50.5	93.4	69.1
TrackFormer [39]	63.3	66.0	61.1	74.1	72.4
OC-SORT [5]	68.1	**84.8**	54.8	**93.4**	68.0
MeMOTR [16]	68.8	82.0	57.8	90.2	69.9
MOTIP (ours)	**72.6**	83.5	**63.2**	92.4	**77.1**
with extra data:					
GTR [73]	54.5	64.8	45.9	67.9	55.8
CenterTrack [72]	62.7	82.1	48.0	90.8	60.0
ByteTrack [69]	62.8	77.1	51.2	94.1	69.8
TransTrack [51]	68.9	82.7	57.5	92.6	71.5
OC-SORT [5]	71.9	86.4	59.8	94.5	72.2
DiffMOT [35]	72.1	86.0	60.5	94.5	72.8

Table 2. Performace comparison with state-of-the-art methods on the SportsMOT [10] test set. The best is shown in **bold**. The results of existing methods are from prior work [10, 16, 49].

frames of images with random intervals, perform ID prediction on the subsequent T frames, and supervise them with Eq. (3). To reduce computational costs, we only backpropagate the gradients for the DETR on four of these frames, while the remaining $T-3$ frames are processed in the no-gradient mode using *torch.no_grad()*. By decoupling the detection and association problems and allowing the ID Decoder to use attention masks [53] to ensure future invisibility, our method can achieve high parallelism and be GPU-friendly. As a result, MOTIP can be efficiently trained using 8 NVIDIA RTX 4090 GPUs. For instance, training on DanceTrack [52] takes less than one day. More details and discussions about the training setups can be found in Appendix B.1.

Hyperparameters. In our experiments, the supervision weight coefficients λ_{cls}, λ_{L1}, λ_{giou} and λ_{id} are set to 2.0, 5.0, 2.0 and 1.0. The maximum temporal length T is set to 29, 59, and 19 for DanceTrack, SportsMOT, and BFT, respectively. The inference thresholds λ_{det}, λ_{new}, and λ_{id} are set to 0.3, 0.6, 0.2. For the training augmentation parameters, we set $\lambda_{\text{occ}} = \lambda_{\text{sw}} = 0.5$. Although fine-tuning some hyperparameters on different datasets may yield better results, for simplicity, we strive to maintain their consistency.

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

We compare MOTIP with numerous previous methods on the DanceTrack [52], SportsMOT [10], and BFT [71] benchmarks, as shown in Tab. 1, Tab. 2, and Tab. 3, respectively. For recent tracking-by-query methods [67, 70] that also use DETR, studies [16, 49, 65] have shown that the choice of different DETR [30, 74] and backbone [20, 31] networks can significantly impact performance. Therefore, we chose Deformable DETR [74] with a ResNet-50 [20] backbone as the competing platform to ensure a fair comparison. Some methods [10, 61, 65, 70] use extra detection datasets to simulate video clips for joint training. We argue this approach is detrimental to the robustness of end-to-end, especially long-term modeling methods, as detailed and discussed in Appendix C and other research [16, 49]. Therefore, we primarily compare results without using additional datasets and still demonstrate superior performance.

DanceTrack. The complex scenarios of frequent occlusions and irregular motion pose a severe challenge to heuristic algorithms [68, 69]. Methods such as Hybrid-SORT [63], C-BIoU [62], and others [13, 35], despite utilizing a more powerful detector [18], more intricate manual designs, and additional tracking cues to enhance performance, are still significantly outperformed by MOTIP. Compared to the strong competitor CO-MOT [61], which also uses Deformable DETR [74], we achieve a new state-of-the-art result with a notable lead of 4.3 HOTA and 6.9 AssA, even surpassing some outstanding results [61, 65] that using additional datasets for training (as shown in the lower of Tab. 1). Such impressive performance demon-

Methods	HOTA	DetA	AssA	MOTA	IDF1
JDE [54]	30.7	40.9	23.4	35.4	37.4
CSTrack [27]	33.2	47.0	23.7	46.7	34.5
FairMOT [68]	40.2	53.3	28.2	56.0	41.8
TransCenter [60]	60.0	66.0	61.1	74.1	72.4
SORT [3]	61.2	60.6	62.3	75.5	77.2
ByteTrack [69]	62.5	61.2	64.1	**77.2**	**82.3**
TrackFormer [39]	63.3	66.0	61.1	74.1	72.4
CenterTrack [72]	65.0	58.5	54.0	60.2	61.0
OC-SORT [5]	66.8	65.4	68.7	77.1	79.3
MOTIP (ours)	**70.5**	**69.6**	**71.8**	77.1	82.1

Table 3. Performace comparison with state-of-the-art methods on the BFT [71] test set. The best performance is shown in **bold**. The results of existing methods are derived from [71] and [49].

self	*hung*	*aug*	HOTA	DetA	AssA	MOTA	IDF1
			57.7	**76.3**	43.9	85.5	56.1
	✓		59.7	75.8	47.2	**85.6**	59.9
	✓	✓	60.8	75.5	49.3	83.7	62.5
		✓	60.2	75.4	48.2	82.0	61.3
✓			59.5	75.3	47.2	85.6	61.1
✓	✓		59.9	75.1	47.9	85.6	62.2
✓	✓	✓	**62.2**	75.2	**51.8**	85.4	**65.6**
✓		✓	**62.2**	75.3	51.5	85.2	64.8

Table 4. Evaluate the impact of different components and strategies. Let *self*, *hung*, and *aug* symbolize the self-attention layer, Hungarian algorithm, and trajectory augmentation, respectively. The gray background is the choice for our final experiment.

strates the considerable potential of our approach in extremely challenging scenarios.

SportsMOT. Sports broadcasts involve frequent camera movements, accompanied by athletes' high-speed movements and repeated interactions. OC-SORT [5] effectively handles sudden stops and starts by explicitly modeling non-linear movements, resulting in a significant improvement over its predecessor [69]. In experiments, our proposed MOTIP significantly outperforms all previous methods by a considerable margin while also surpasses competitors [16, 39] using the same detector [74]. To avoid introducing additional engineering challenges and intricate remedies, as elaborated in Appendix C, we have not provided the results with extra training datasets like [23, 35, 43]. However, our method, trained solely on the SportsMOT train set, still surpasses many joint training methods [5, 35, 51] especially on the association accuracy (AssA), as shown in the lower part of Tab. 2, demonstrating our commendable performance and potential.

BFT. Tracking birds differs in many ways from tracking humans [10, 11, 40, 48, 52]. On the one hand, birds have highly dynamic movements due to their three-dimensional activity space, compared to ground targets. On the other hand, their appearance is often more similar due to the absence of artificial distinctions such as clothing. Therefore, this presents a challenging new problem that is different from previous ones. Nevertheless, as shown in Tab. 3, our MOTIP has established a new state-of-the-art result with 70.5 HOTA and 71.8 AssA. This helps demonstrate the generalization ability of our method across different scenarios.

4.4. Ablations

We conduct our ablation experiments on DanceTrack [52] because it is challenging and offers a large-scale training set that better unlocks the model's potential. Unless otherwise stated, all trajectory augmentation techniques will not be used, *i.e.*, $\lambda_{\text{occ}} = \lambda_{\text{sw}} = 0.0$. More details and analyses will be elaborated in Appendix B.3.

Hungarian Algorithm. The Hungarian algorithm is a commonly used approach for finding global optimal solutions [5, 69]. However, by default, we do not use the Hungarian algorithm in our method, but rather opt for the more straightforward inference procedure described in Sec. 3.4. Nonetheless, we explore its impact on our MOTIP. As shown in the bottom half of Tab. 4, it does not provide considerable benefits to our method. We believe this is because our model inherently possesses the ability to find optimal solutions, which also indicates that our approach is far removed from traditional heuristic algorithms.

Self-Attention in ID Decoder. Earlier, we mentioned that MOTIP can find global optimal solutions. We believe this can be attributed to the self-attention layers in the ID Decoder. We perform an ablation study on this design in Tab. 4. Not surprisingly, using only the decoder layers can still achieve acceptable tracking performance. However, we argue that self-attention layers are crucial for better tracking. This is because they help the current objects exchange identity information during inference, thereby preventing confusion among similar targets. Therefore, the impact of the Hungarian algorithm is amplified, which is why you can observe a remarkable improvement. When trajectory augmentation is introduced, the performance gap between the approach without self-attention layers and the final MOTIP further widens, underscoring the critical role of self-attention layers.

Comparison with ReID Pipelines. Since our MOTIP also uses object-level features as tracking cues, it can easily be mistaken for a type of ReID method. In Tab. 5, we compare two ReID learning pipelines derived from [68] and [42], identified as *re-id* and *contra*, respectively. In the upper section (#1 to #3) of Tab. 5, we perform experiments using the

#	*Train*	*Form*	HOTA	AssA	IDF1
# 1	*Two-Stage*	*re-id*	29.4	11.5	22.1
# 2		*contra*	41.0	22.6	36.4
# 3		*id-pred*	55.4	41.1	55.7
# 4	*One-Stage*	*re-id*	41.0	22.5	35.0
# 5		*re-id*‡	50.6	34.7	50.9
# 6		*contra*	49.8	33.0	47.1
# 7		*contra*‡	52.6	37.3	54.0
# 8		$\star$*contra*	51.2	35.0	48.1
# 9		$\star$*contra*‡	54.5	40.3	55.5
# 10		*id-pred*	**59.5**	**47.2**	**61.1**

Table 5. Comparison with common ReID pipelines. As the tracking formulation (*Form*), *re-id* and *contra* represent training the model using the formula from two well-known ReID methods, [68] and [42], respectively, and inference is based on cosine similarity. The *id-pred* indicates our proposed MOTIP. ‡ and ⋆ represent the use of the Hungarian algorithm and the trajectory enhancement module, respectively.

frozen, well-trained DETR weights. These results clearly illustrate that our method shows significant advantages over the other two formulations under the same object features. In the remaining part of Tab. 5, we jointly train all network parameters in a one-stage manner. To eliminate the influence of introducing additional structures, we incorporate a trajectory enhancement module, identical to the structure of our ID Decoder, into some experiments, denoted as ⋆. The experimental results demonstrate that, whether utilizing a trajectory enhancement module or an advanced assignment strategy (Hungarian algorithm, refer to ‡), these methods still lag behind our MOTIP. This can be attributed to MOTIP's ability to manage historical tracklets with greater flexibility, as visualized and discussed further in Appendix D.3. In contrast, ReID methods [42, 54, 68] employ heuristic algorithms to integrate historical information and perform similarity calculation independently, which limits the model's adaptability. Furthermore, we emphasize that the incorporation and interaction of ID information can enhance the model's capability to distinguish similar trajectories in complex scenarios while facilitating better assignment decisions. Incidentally, the introduction of the ID field also enables the trajectory augmentation mentioned in Sec. 3.3, further boosting the tracking performance, as shown in Tab. 6.

Trajectory Augmentation. In Tab. 6, we explore the hyperparameters of the two different trajectory augmentation approaches mentioned in Sec. 3.3. The performance significantly improves when λ_{occ} is set to 0.5. However, if too many tokens are discarded ($\lambda_{\text{occ}} = 1.0$), it can undermine the results due to excessive difficulty. Set λ_{occ} to 0.5,

λ_{occ}	λ_{sw}	HOTA	DetA	AssA	MOTA	IDF1
0.0	0.0	59.5	75.3	47.2	85.6	61.1
0.5	0.0	60.7	75.0	49.4	85.2	62.7
1.0	0.0	58.6	**75.5**	45.7	**85.7**	59.8
0.5	0.2	61.6	75.4	50.5	85.4	64.1
0.5	0.5	**62.2**	75.3	**51.5**	85.2	**64.8**
0.5	0.8	59.8	75.0	47.9	82.7	61.8

Table 6. Exploration of the hyperparameters for the trajectory augmentation techniques mentioned in Sec. 3.3. The gray background is the choice for our final experiment.

when progressively increasing the λ_{sw} from 0.2 to 0.8, our method achieves the best performance while λ_{sw} is set to 0.5. Therefore, we use $\lambda_{\text{occ}} = \lambda_{\text{sw}} = 0.5$ to conduct experiments as the final results in Sec. 4.3. It should be noted that using different augmentation hyperparameters on different datasets can yield better performance. However, to avoid over-focusing on engineering tricks, we use this unified setting across all datasets.

5. Limitations and Discussions

Although we achieved new state-of-the-art results across numerous datasets, our method still has considerable room for improvement, and several noteworthy limitations remain. As discussed in Sec. 3.2, our approach is simple and intuitive, adhering to the philosophy *less is more*. Therefore, our primary objective is to verify the feasibility of treating MOT as an in-context ID prediction process, rather than delving into highly customized model designs. This leaves ample room for future research to explore enhancements and customizations, such as tailored ID Decoder layers, additional tracking cues (*e.g.* motion, depth, *etc.*), and more sophisticated trajectory modeling techniques. Another limitation is that the capacity K of the ID dictionary may not be enough in crowded scenarios. We have shown that in most cases, the token utilization rate is below 40%. If necessary, K can be adjusted upwards for extreme scenarios. Just like DETRs set the number of detect queries to 300 by default, our setting is also for general scenarios.

6. Conclusion

We have introduced treating multiple object tracking as an in-context ID prediction task, which simplifies both the training and tracking processes. Based on this, we proposed MOTIP, a simple yet effective baseline design. Surprisingly, our method surpassed the state-of-the-art on all benchmarks. This demonstrates the tremendous potential of our pipeline and method, suggesting it can serve as a viable inspiration for future research.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2022ZD0160900), Jiangsu Frontier Technology R&D Program (No. BF2024076), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and Nanjing University-China Mobile Communications Group Co., Ltd. Joint Institute. Besides, Ruopeng Gao would like to thank Yunzhe Lv for the kind discussion and Muyan Yang for the social support.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *CoRR*, abs/2206.14651, 2022. 2

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008, 2008. 5

[3] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016. 1, 2, 7

[4] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *CVPR*, pages 8080–8090. IEEE, 2022. 2, 3

[5] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric SORT: rethinking SORT for robust multi-object tracking. *CoRR*, abs/2203.14360, 2022. 2, 6, 7

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV (1)*, pages 213–229. Springer, 2020. 2, 3, 4

[7] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, pages 1–6. IEEE Computer Society, 2018. 2

[8] Sijia Chen, En Yu, Jinyang Li, and Wenbing Tao. Delving into the trajectory long-tail distribution for muti-object tracking. In *CVPR*, pages 19341–19351. IEEE, 2024. 2

[9] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV (4)*, pages 215–230. Springer, 2012. 1

[10] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. In *ICCV*, 2023. 1, 2, 5, 6, 7

[11] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian D. Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *CoRR*, abs/2003.09003, 2020. 7

[12] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? In *NeurIPS*, 2022. 2

[13] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Trans. Multim.*, 25:8725–8737, 2023. 6

[14] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljosa Osep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *ICCV*, pages 10829–10839. IEEE, 2021. 2

[15] Jianjun Gao, Yi Wang, Kim-Hui Yap, Kratika Garg, and Boon Siew Han. Occlutrack: Rethinking awareness of occlusion for enhancing multiple pedestrian tracking. *CoRR*, abs/2309.10360, 2023. 2

[16] Ruopeng Gao and Limin Wang. MeMOTR: Long-term memory-augmented transformer for multi-object tracking. In *ICCV*, pages 9901–9910, 2023. 2, 3, 5, 6, 7

[17] Ruopeng Gao, Yuyao Wang, Chunxu Liu, and Limin Wang. History-aware transformation of reid features for multiple object tracking. *arXiv preprint arXiv:2503.12562*, 2025. 2

[18] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 2, 6

[19] Shengyu Hao, Peiyuan Liu, Yibing Zhan, Kaixun Jin, Zuozhu Liu, Mingli Song, Jenq-Neng Hwang, and Gaoang Wang. Divotrack: A novel dataset and baseline method for cross-view multi-object tracking in diverse open scenes. *Int. J. Comput. Vis.*, 132(4):1075–1090, 2024. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 3, 5, 6

[21] Cheng Huang, Shoudong Han, Mengyu He, Wenbo Zheng, and Yuhao Wei. Deconfusetrack: Dealing with confusion for multi-object tracking. In *CVPR*, pages 19290–19299. IEEE, 2024. 2

[22] Hsiang-Wei Huang, Cheng-Yen Yang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Exploring learning-based motion models in multi-object tracking. *CoRR*, abs/2403.10826, 2024. 2

[23] Hsiang-Wei Huang, Cheng-Yen Yang, Jiacheng Sun, Pyong-Kun Kim, Kwang-Ju Kim, Kyoungoh Lee, Chung-I Huang, and Jenq-Neng Hwang. Iterative scale-up expansioniou and deep features association for multi-object tracking in sports. In *WACV (Workshops)*, pages 163–172. IEEE, 2024. 7

[24] Oluwafunmilola Kesa, Olly Styles, and Victor Sanchez. Multiple object tracking and forecasting: Jointly predicting current and future object locations. In *WACV (Workshops)*, pages 560–569. IEEE, 2022. 1

[25] Bruno Korbar and Andrew Zisserman. End-to-end tracking with a multi-query transformer. *CoRR*, abs/2210.14601, 2022. 3

[26] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*, abs/1504.01942, 2015. 1, 2

[27] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multi-object tracking. *CoRR*, abs/2010.12138, 2020. 7

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, pages 740–755. Springer, 2014. 5

[29] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007. IEEE Computer Society, 2017. 4

[30] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: dynamic anchor boxes are better queries for DETR. In *ICLR*. OpenReview.net, 2022. 2, 3, 6

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11966–11976. IEEE, 2022. 6

[32] Zelin Liu, Xinggang Wang, Cheng Wang, Wenyu Liu, and Xiang Bai. Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. *CoRR*, abs/2306.05238, 2023. 2

[33] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021. 5

[34] Run Luo, Zikai Song, Lintao Ma, Jinlin Wei, Wei Yang, and Min Yang. Diffusiontrack: Diffusion model for multi-object tracking. In *AAAI*, pages 3991–3999. AAAI Press, 2024. 2

[35] Weiyi Lv, Yuhang Huang, Ning Zhang, Ruei-Sung Lin, Mei Han, and Dan Zeng. Diffmot: A real-time diffusion-based multiple object tracker with non-linear prediction. In *CVPR*, pages 19321–19330. IEEE, 2024. 2, 6, 7

[36] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. *arXiv preprint arXiv:2302.11813*, 2023. 2

[37] Nima Mahmoudi, Seyed Mohammad Ahadi, and Mohammad Rahmati. Multi-target tracking using cnn-based features: CNNMTT. *Multim. Tools Appl.*, 78(6):7077–7096, 2019. 2

[38] Gianluca Mancusi, Aniello Panariello, Angelo Porrello, Matteo Fabbri, Simone Calderara, and Rita Cucchiara. Trackflow: Multi-object tracking with normalizing flows. In *ICCV*, pages 9531–9543, 2023. 2

[39] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixé, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *CVPR*, pages 8834–8844. IEEE, 2022. 2, 3, 6, 7

[40] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016. 1, 2, 7

[41] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, pages 164–173. Computer Vision Foundation / IEEE, 2021. 6

[42] Pierre-François De Plaen, Nicola Marinello, Marc Proesmans, Tinne Tuytelaars, and Luc Van Gool. Contrastive learning for multi-object tracking with transformers. In *WACV*, pages 6853–6863. IEEE, 2024. 5, 7, 8

[43] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, pages 17939–17948. IEEE, 2023. 2, 7

[44] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666. Computer Vision Foundation / IEEE, 2019. 4

[45] Ergys Ristani, Francesco Solera, Roger S. Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops (2)*, pages 17–35, 2016. 5

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 2, 3, 5

[47] Leonardo Saraceni, Ionut Marian Motoi, Daniele Nardi, and Thomas A. Ciarfuglia. Agrisort: A simple online real-time tracking-by-detection framework for robotics in precision agriculture. In *ICRA*, pages 2675–2682. IEEE, 2024. 2

[48] Atom Scott, Ikuma Uchida, Ning Ding, Rikuhei Umemoto, Rory P. Bunker, Ren Kobayashi, Takeshi Koyama, Masaki Onishi, Yoshinari Kameda, and Keisuke Fujii. Teamtrack: A dataset for multi-sport multi-object tracking in full-pitch videos. In *CVPR Workshops*, pages 3357–3366. IEEE, 2024. 1, 7

[49] Mattia Segù, Luigi Piccinelli, Siyuan Li, Yung-Hsu Yang, Bernt Schiele, and Luc Van Gool. Samba: Synchronized set-of-sequences modeling for multiple object tracking. *CoRR*, abs/2410.01806, 2024. 2, 3, 6, 7

[50] Jenny Seidenschwarz, Guillem Brasó, Victor Castro Serrano, Ismail Elezi, and Laura Leal-Taixé. Simple cues lead to a strong multi-object tracker. In *CVPR*, pages 13813–13823. IEEE, 2023. 2

[51] 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. *CoRR*, abs/2012.15460, 2020. 2, 6, 7

[52] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, pages 20961–20970. IEEE, 2022. 2, 3, 5, 6, 7

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 2, 4, 6

[54] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV (11)*, pages 107–122. Springer, 2020. 1, 2, 7, 8

[55] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 1, 2

[56] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017. 1, 2

[57] 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*, pages 12352–12361. Computer Vision Foundation / IEEE, 2021. 6

[58] Tao Wu, Runyu He, Gangshan Wu, and Limin Wang. Sportshhi: A dataset for human-human interaction detection in sports videos. In *CVPR*, pages 18537–18546. IEEE, 2024. 1

[59] Changcheng Xiao, Qiong Cao, Zhigang Luo, and Long Lan. Mambatrack: A simple baseline for multiple object tracking with state space model. In *ACM Multimedia*, pages 4082–4091. ACM, 2024. 2

[60] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *CoRR*, abs/2103.15145, 2021. 2, 7

[61] Feng Yan, Weixin Luo, Yujie Zhong, Yiyang Gan, and Lin Ma. Bridging the gap between end-to-end and non-end-to-end multi-object tracking, 2023. 2, 3, 5, 6

[62] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *WACV*, pages 4788–4797. IEEE, 2023. 2, 6

[63] Mingzhan Yang, Guangxin Han, Bin Yan, Wenhua Zhang, Jinqing Qi, Huchuan Lu, and Dong Wang. Hybrid-sort: Weak cues matter for online multi-object tracking. *CoRR*, abs/2308.00783, 2023. 2, 6

[64] Sisi You, Hantao Yao, Bing-Kun Bao, and Changsheng Xu. UTM: A unified multiple object tracking model with identity-aware feature enhancement. In *CVPR*, pages 21876–21886. IEEE, 2023. 2

[65] En Yu, Tiancai Wang, Zhuoling Li, Yuang Zhang, Xiangyu Zhang, and Wenbing Tao. Motrv3: Release-fetch supervision for end-to-end multi-object tracking. *CoRR*, abs/2305.14298, 2023. 2, 3, 5, 6

[66] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2633–2642. Computer Vision Foundation / IEEE, 2020. 1

[67] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: end-to-end multiple-object tracking with transformer. In *ECCV (27)*, pages 659–675. Springer, 2022. 2, 3, 5, 6

[68] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129(11):3069–3087, 2021. 1, 2, 3, 6, 7, 8

[69] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV (22)*, pages 1–21. Springer, 2022. 1, 2, 5, 6, 7

[70] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors, 2022. 2, 3, 5, 6

[71] Guangze Zheng, Shijie Lin, Haobo Zuo, Changhong Fu, and Jia Pan. Nettrack: Tracking highly dynamic objects with a net. In *CVPR*, pages 19145–19155. IEEE, 2024. 1, 2, 5, 6, 7

[72] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV (4)*, pages 474–490. Springer, 2020. 2, 6, 7

[73] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, pages 8761–8770. IEEE, 2022. 6

[74] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*. OpenReview.net, 2021. 2, 3, 4, 5, 6, 7