Contrastive Learning for Multi-Object Tracking with Transformers

Pierre-François De Plaen*[1] Nicola Marinello*[1] Marc Proesmans[1,3] Tinne Tuytelaars[1]
Luc Van Gool[1,2,3]
[1] ESAT-PSI, KU Leuven, Belgium [2] CVL, ETH Zürich, Switzerland [3] TRACE vzw
{pdeplaen,nicola.marinello,marc.proesmans,tinne.tuytelaars,luc.vangool}@esat.kuleuven.be

Abstract

The DEtection TRansformer (DETR) opened new possibilities for object detection by modeling it as a translation task: converting image features into object-level representations. Previous works typically add expensive modules to DETR to perform Multi-Object Tracking (MOT), resulting in more complicated architectures. We instead show how DETR can be turned into a MOT model by employing an instance-level contrastive loss, a revised sampling strategy and a lightweight assignment method. Our training scheme learns object appearances while preserving detection capabilities and with little overhead. Its performance surpasses the previous state-of-the-art by +2.6 mMOTA on the challenging BDD100K dataset and is comparable to existing transformer-based methods on the MOT17 dataset.

1. Introduction

In recent years, transformer networks [8, 22, 37] have gained momentum in computer vision. The original transformer [37] was designed for sequence modeling and transduction tasks. Its architecture is notable for using attention modules, which allow for capturing long-range contextual relationships between word tokens.

Likewise, the DEtection TRransformer (DETR) [4] models object detection as a translation task, converting image features into object-level representations. Its cross-attention mechanism enables extracting relevant features for detection without the need for anchors, hand-crafted feature extraction methods, or local prediction biases inherent in CNN architectures [12, 29, 30]. The self-attention mechanism, coupled with bipartite matching [19, 27], helps object representations reach a consensus and eliminate redundancy in predictions.

Multi-Object Tracking (MOT) is a task that requires both object detection and object association. Object association links detected objects across frames by considering their appearance, position, size, or other characteristics to determine which detections correspond to the same real-world object. The algorithm must be robust against (partial) occlusion, loss of sight, missed detection, appearance variation, etc. In particular, the re-identification of objects after occlusions can be challenging.

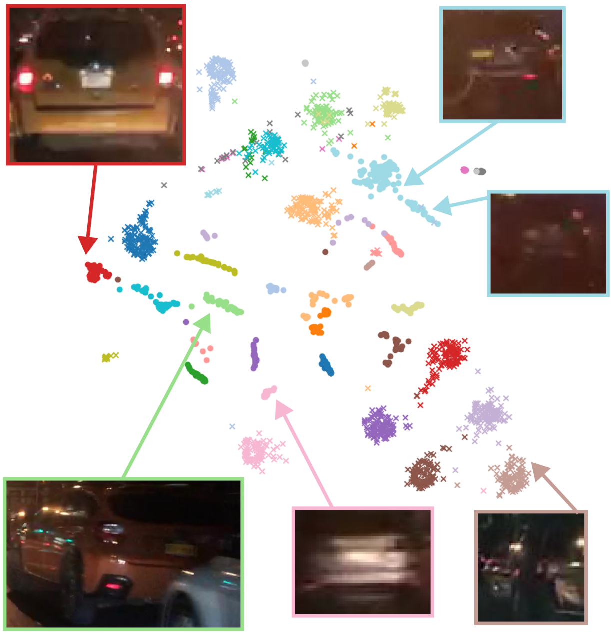


Figure 1. T-SNE projection of the predicted embeddings for the first 40 ground-truth objects in video *b23f7012-fab06dac* of the BDD100K validation set. Each color-symbol pair represents a ground-truth tracking ID assigned with DETR's bipartite matching. Even during nighttime, the method can discriminate similar objects.

The object-level representations from DETR could be potentially used to re-identify objects across different frames. However, while each of these embeddings corresponds to a particular object in a specific frame, they do not provide a fine-grained description of the target appearance as needed for MOT, as the model is only trained to regress the bounding box and to classify the detected object. As a

[*] These authors contributed equally.

result, they are insufficient for object re-identification.

Our model, named Contrastive TRansformer (ContrasTR), takes advantage of the representations produced by DETR to encode discriminative identity-level features for seamless object re-identification. This can be accomplished with little overhead using an instance-level contrastive loss and a revised sampling strategy. During training, the batch of images is built with sets of non-consecutive frames from multiple videos. It is important to select frames that are distant in time to increase the diversity of the appearance of objects, and to select images from several videos to have a wide diversity of negative examples for the contrastive loss. This sampling approach increases robustness under large object motion and adverse conditions, as illustrated in Figure 4.

Unlike competing DETR-like transformers [3, 18, 25, 33, 40, 43, 48, 51], we model MOT as a multi-task learning problem and introduce a model capable of performing joint detection and association with a single set of internal object representations. This simple yet effective approach removes the need for expensive additional transformer layers to update tracks [33, 43, 48, 51] or to perform object association [3, 40].

Inspired by advances in supervised contrastive learning for image classification [17], we propose a supplementary pre-training strategy on object detection datasets that does not require annotated tracking instance IDs and videos. This method leverages the size and diversity of object detection datasets. We show that it improves the tracking embedding space and boosts performance even when fine-tuning on large tracking datasets. We evaluate our method on the MOT17 [26] benchmark and the more diverse and challenging BDD100K [42] dataset.

Our main contributions can be summarized as follows:

- We show how to turn a DETR-like object detection model into a tracking model with little overhead by learning to discriminate instances. Our approach reaches performance on par or higher than more complicated architectures. Furthermore, it improves SOTA by + 2.6 mMOTA on the BDD100K dataset.
- We highlight the required components for learning robust joint detection and association features through an extensive ablation study.
- We present an additional pre-training scheme for MOT that takes advantage of the size and diversity of object detection datasets and is scalable to large object detection datasets.

2. Related Work

This section briefly overviews joint detection and tracking methods, contrastive learning, and the use of DETR-like transformers for MOT.

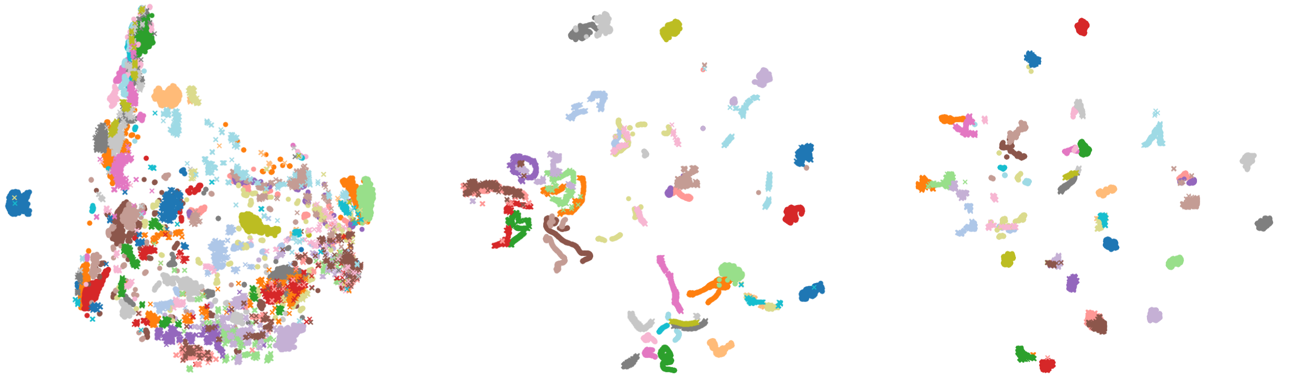
Joint-detection-and-tracking. Several works are focused on the implementation of models capable of performing detection and tracking [10, 23, 28, 35, 36, 38, 47, 50]. In particular, some of them [23, 28, 36, 38, 47] train the detection model to extract appearance embeddings used to re-identify objects across frames.

JDE [38] introduced a CNN-based joint detection and tracking model that prioritizes speed. Specifically, a detection model is trained to generate discriminative feature embeddings using a cross-entropy loss with one category per unique instance. FairMOT [47] presented a set of design improvements to mitigate the negative effect of anchors and high dimensional re-ID features when training a joint detection and tracking model. Furthermore, FairMOT introduced a scheme for training a tracker using individual images from detection datasets, bypassing the requirement for sequences. However, the scalability of their cross-entropy loss-based training procedure is limited for massive datasets like BDD100K, requiring a classification head with a significant number of parameters (approximately 230 million for the projection head of FairMOT on the BDD100K detection set). In contrast, our training approach decouples the tracking model size from the dataset size, ensuring scalability for large-scale datasets.

Contrastive Learning. Self-supervised contrastive learning has become a popular technique for learning robust and discriminative feature representations in computer vision [6, 13, 15, 16, 34]. For example, [6] learns representations by augmenting each image in a batch twice and encouraging the similarity between views of the same image while encouraging dissimilarity with views of different images. Later, supervised contrastive learning for image classification [17] has been proposed as a more effective method for learning feature representations, where labeled data is used to guide the contrastive loss function. In addition to the augmented views, images from the same class are also considered positive examples. This improves classification accuracy by a significant margin.

Contrastive learning also finds applications in tracking. QDTrack [28] learns appearance with a contrastive loss on object regions by sampling pairs of images from neighboring frames, whereas we sample multiple frames per video and from different videos, allowing for a much larger diversity of positive and negative examples for contrastive learning. MTrack [41] predicts a set of feature vectors for each object in an image. Contrastive learning is then used to push an object's inter-frame and intra-frame feature vectors to be similar and feature vectors from other objects to be dissimilar. Since multiple feature vectors are generated per object and frame, directly using contrastive learning would be costly. MTrack instead pushes feature vectors towards their corresponding trajectory centers.

Transformers for Multi-Object Tracking. DETR



(a) Pre-trained on detection without a contrastive loss. In this case, tracking embeddings correspond to the object embeddings.

(b) Pre-trained on detection with our contrastive loss (no tracking ID annotations required).

(c) Pre-trained on detection with our contrastive loss, then trained for tracking on MOT17 with a contrastive loss.

Figure 2. t-SNE visualization of the tracking embeddings of video 4 of MOT17. Each color-symbol pair represents a unique tracking ID, assigned with DETR's bipartite matching. All models are pre-trained on the CrowdHuman dataset [32] and evaluated on the validation set of MOT17 [26]. Without the contrastive loss, Deformable-DETR's embeddings are not clustered per instance id.

[4] introduced a simple one-stage framework for object detection, removing the need to create anchor boxes manually and for Non-Maximum Suppression (NMS). It models object detection as a sequence-to-sequence translation task: it first extracts image features, which then serve as keys and values to update a set of learnable object queries through a series of transformer decoder blocks. Multiple works have built trackers on top of DETR by exploiting the object-level internal representations for class-agnostic MOT [18] and MOT [3, 25, 33, 40, 43, 51].

TrackFormer [25] and MOTR [43] introduced track queries to model instances over time and preserve identities. Newborn objects are detected with object queries, and their hidden states produce track queries for the next frame. Both query types are fed into the transformer decoder. TrackFormer is trained on consecutive frame pairs, which prevents its application for long-range occlusions. MOTR instead learns long-range dependencies through a cross-attention temporal aggregation network and a video-level loss. Both approaches require track augmentations and face challenges in detecting newborn objects. MOTRv2 [48] solved the poor performance on newborn objects by using an additional YOLOX [11] as a proposal network, but at the cost of running this additional detection model.

Similarly, MeMOT [3] uses a memory aggregation module to encode temporal object information from a memory of previous embeddings into track embeddings through attention modules. The model is then trained to predict uniqueness scores so that only new objects are added to the track embeddings.

TransTrack [33] instead uses a unique set of object queries to detect objects. Additionally, a parallel transformer decoder autoregressively updates tracked objects. The association is performed by matching the outputs of both decoders with an IoU cost, which leads to poor association performance. In contrast, our approach avoids a parallel decoder, opting for learning instance-level embeddings that are more robust for re-identification.

3. Learning Identity-Level Representations

We introduce a method that can turn any DETR-like object detector into a model that can perform joint detection and MOT. We describe how such a model can learn identity-level features with the aid of an additional loss term and a revisited sampling strategy. We further introduce a pre-training scheme that allows us to learn better representations by exploiting large-scale object detection datasets.

3.1. Preliminaries

Given an input image I, DETR extracts image features with a backbone and a transformer encoder. These features then serve as keys and values to update a set of N learnable object queries through a series of transformer decoder blocks. Each output object embedding $\{\hat{x}_1^I, \hat{x}_2^I, \cdots, \hat{x}_N^I\}$ represents a different possible object in the image. Class probabilities and bounding box positions are then predicted for each output embedding through Feed-Forward Network (FFN) heads.

The method uses the Hungarian algorithm [19, 27] to match ground truth objects with predictions, forming an optimal bipartite matching $\hat{\sigma}^I$. The assignment minimizes the global matching cost, which is a linear combination of classification and localization costs. The matched predictions are then improved with object-specific losses (i.e. classification and localization losses), and the remaining predictions are trained to predict the background class.

3.2. MOT as a multi-task learning problem

Our method exploits the object-level representations from DETR for re-identification. As illustrated in Figure 2a, these representations do not provide a fine-grained description of the objects' appearance. We, therefore, introduce an additional loss term to force the model to encode identity-level features besides the bounding box and class information. More specifically, we build an embedding space for tracking in which different views of an object are positioned close to one another, whereas other objects lie further in the embedding space, see Figure 2c.

Tracking projection head. We pass each output embedding of the decoder to three shared FFNs that predict respectively class probabilities, bounding box positions and *tracking embeddings* $\left\{\hat{z}^I_1, \hat{z}^I_2, \cdots, \hat{z}^I_N\right\}$. The projection into *tracking embeddings* is more convenient for performing object re-identification.

Contrastive learning. Our framework relies on the bipartite matching $\hat{\sigma}^I$ to associate output object embeddings to ground-truths objects and, therefore, the corresponding tracking embeddings to the annotated tracking IDs. As a result, each tracking embedding $\hat{z}^I_i \in \mathbb{R}^D$ that was associated with a ground truth will be associated with an object instance ID: $\left\{(\hat{z}^I_{\hat{\sigma}^I(1)}, ID^I_1), (\hat{z}^I_{\hat{\sigma}^I(2)}, ID^I_2), \cdots, (\hat{z}^I_{\hat{\sigma}^I(M)}, ID^I_M)\right\}$ where M is the number of ground truth objects in the image. For simplicity of notations, the set of tracking embeddings matched with an object over the full batch is denoted by:

$$\tilde{\boldsymbol{z}} = \left[\hat{z}^1_{\hat{\sigma}^I(1)}, \hat{z}^1_{\hat{\sigma}^I(2)}, \cdots, \hat{z}^1_{\hat{\sigma}^I(M_1)}, \hat{z}^2_{\hat{\sigma}^I(1)}, \cdots, \hat{z}^B_{\hat{\sigma}^I(M_B)}\right] \quad (1)$$

where M_b is the total number of ground truth objects in video b and B is the batch size.

Given a tracking embedding $\tilde{z}_i$ and its corresponding ground truth instance id ID_i, we define the set of positives $\mathcal{P}(i)$ as embeddings in the batch that are associated with objects from the same video with the same ground truth tracking id ID_i. The remaining are negatives $\mathcal{N}(i)$. Following previous work on supervised contrastive learning [17], the contrastive loss for tracking embeddings $\tilde{z}_i$ and $\tilde{z}_j$ corresponding to two views from the same instance (i.e. such that $ID_i = ID_j$) takes the following form.

$$l_{\text{contr}}\left(\tilde{z}_i, \tilde{z}_j, \mathcal{N}(i)\right) = \\ -\log\left(\frac{e^{sim(\tilde{z}_i,\tilde{z}_j)/\tau}}{e^{sim(\tilde{z}_i,\tilde{z}_j)/\tau} + \sum_{k\in\mathcal{N}(i)} e^{sim(\tilde{z}_i,\tilde{z}_k)/\tau}}\right) \quad (2)$$

where $\tau \in \mathbb{R}^+$ is the temperature hyper-parameter.

Note that only the similarity of the target positive pair (i, j) and of the negative examples $\mathcal{N}(i)$ is considered in the denominator so that the loss value for a given pair is invariant to the other positive pairs. The contrastive loss is then averaged over the set of positive pairs:

$$\mathcal{L}_{\text{contr}}\left(\tilde{z}_i, \mathcal{P}(i), \mathcal{N}(i)\right) = \frac{1}{|\mathcal{P}(i)|}\sum_{j\in\mathcal{P}(i)} l_{\text{contr}}\left(\tilde{z}_i, \tilde{z}_j, \mathcal{N}(i)\right) \quad (3)$$

where $|\mathcal{P}(i)|$ denotes the number of positives for ground truth object i in the batch.

Such a loss forces the tracking embeddings associated with the same identity to be pulled as close as possible while tracking embeddings of different identities to be pushed as far as possible.

We then reformulate the overall loss as a weighted sum of per-object classification, localization, and contrastive loss terms:

$$\begin{aligned}\mathcal{L}_{\text{train}}(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) =& \lambda_{\text{class}}\mathcal{L}_{\text{class}}(c_i, \hat{c}_{\hat{\sigma}(i)}) \\ &+ \mathbb{1}_{\{c_i\neq\varnothing\}}\lambda_{\text{box}}\mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \\ &+ \mathbb{1}_{\{c_i\neq\varnothing\}}\lambda_{\text{contr}}\mathcal{L}_{\text{contr}}\left(\tilde{z}_i, \mathcal{P}(i), \mathcal{N}(i)\right)\end{aligned} \quad (4)$$

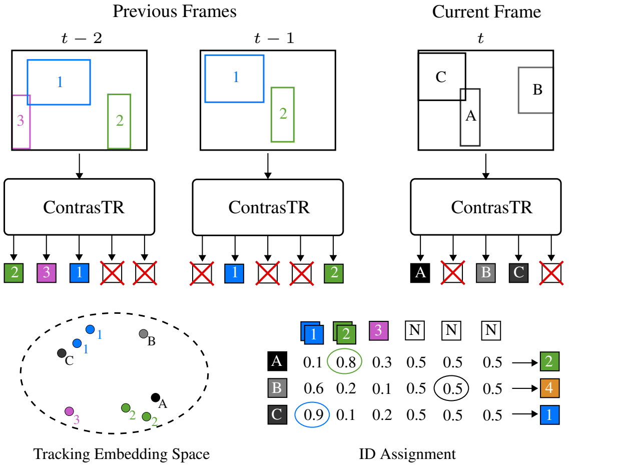
where $\mathcal{L}_{\text{class}}$, $\mathcal{L}_{\text{box}}$ and $\mathcal{L}_{\text{contr}}$ are respectively the classification, localization and contrastive loss terms with their corresponding weighting hyper-parameters λ, c_i is the target class label, $b_i \in [0, 1]^4$ is the target bounding box in coordinates relative to the image size.

3.3. Sampling strategy

The default data sampling strategy to build training batches for object detection is to sample images from the training set with a uniform distribution. This allows each mini-batch to be representative of the training set. However, such a way of sampling images dramatically reduces the probability of having multiple views of the same ground truth object in different images. This implies a very small, if not non-existent, set of positive pairs within each batch. Yet, contrastive learning usually benefits from having many positive pairs [17]. We, therefore, design an alternative sampling strategy with the following properties:

- a high probability of having many positives by sampling multiple frames from each video in the batch;
- variation in objects' appearance by sampling non-consecutive frames;
- a large diversity in the negative examples by sampling from multiple videos.

We build training batches by sampling with a uniform distribution N_v videos and then sampling N_f frames from each of these videos again with a uniform distribution. This sampling strategy increases the probability of including the same identities from different frames, potentially including more variations of the same object. At the same time, it also ensures a diverse enough selection of objects with different identities and contexts, as the frames also come from different videos. Besides, such a sampling strategy practically



	1	2	3	N	N	N	
A	0.1	0.8	0.3	0.5	0.5	0.5	→ 2
B	0.6	0.2	0.1	0.5	0.5	0.5	→ 4
C	0.9	0.1	0.2	0.5	0.5	0.5	→ 1

Figure 3. **ContrasTR.** Our framework during the *inference* phase. ID assignment maximizes the global cosine similarity between the predictions and previous instances. A *new instance* (N) entry is added to the cost matrix for each prediction.

preserves the batch variety needed not to compromise the model's object detection capabilities.

3.4. Learning MOT from Object Detection datasets

Most tracking methods are pre-trained on object detection datasets, which are usually larger and more diverse than MOT datasets. However, MOT requires features to be informative enough to discriminate different instances of the same class, and object detectors don't offer the granularity level required for object association. We show how to learn an embedding space that discriminates between instances during the object detection pre-training. We then empirically verify that this pre-training phase offers a better initialization for the MOT training phase.

We follow a similar approach to self-supervised contrastive learning on image classification [6, 15, 16, 34]. For a set of N randomly sampled images, we build a batch of $2N$ images by applying two different sets of augmentations to each image. We then train the model with a supervised contrastive loss in which representations of the same ground truth object are pushed to be similar, whereas other object representations are pushed to be dissimilar.

3.5. Object association with maximal similarity

At inference time, frames are processed online. Detections with a confidence score above the *objectness threshold* are collected into a memory queue m that allows up to T most recent frames in the past. It works essentially as a first-in-first-out (FIFO) queue. This memory allows the model to re-identify objects even after occlusions. The ID assignment process is the following:

1. The model generates object embeddings for the current frame and uses the projection head to generate tracking embeddings for each non-background prediction.
2. A similarity matrix is formed with new predictions as rows and previously predicted instances as columns. The embeddings from previous frames are grouped per instance, where the highest similarity value is kept.

$$s_{i,j} = \max_{t' \in [t-T-1, t-1]} sim(\hat{z}_i, m_j^{t'}) \tag{5}$$

where $m_j^{t'}$ is the memory of tracking embedding with assigned id j at frame t'. An additional *new instance* entry is added to the cost matrix for each predicted embedding, with a pre-defined *tracking threshold* score.
3. The Hungarian algorithm finds a bipartite matching between the predicted and the previous tracking embeddings. The solution maximizes the sum of the similarities of the assignment.

$$\hat{\mu} = \arg\max_{\mu \in \mathcal{M}} \sum_{i=1}^{K} s_{i,\mu(i)} \tag{6}$$

where K is the number of non-background predictions and $\mathcal{M}$ is the set of all possible assignments.
4. Each object is assigned the ID of the corresponding previous embedding. If an object is matched with a *new instance* entry, it is assigned a new unique ID.

The process is repeated frame-by-frame online and is illustrated on Figure 3.

It should be noted that the memory size T can be adjusted during inference time, based on the specific requirements of the target application and the video frame rate, without the need to re-train the model.

4. Experiments

In this section, we show the results of our method on the MOT17 [26] and BDD100K [42] benchmarks. We then conduct an ablation study to obtain more detailed insights about our framework.

4.1. Experimental setup

Datasets. MOT17 [26] is a widely used benchmark to evaluate MOT methods. It consists of training and test sets, each including seven sequences. Only pedestrians are evaluated in this benchmark. BDD100K [42] is a large-scale driving video dataset comprising 100K diverse video clips in different environmental and weather conditions. The dataset provides several types of annotations, such as object bounding boxes, semantic and instance segmentation, tracking IDs, etc. The object detection set consists of one annotated frame per video and the MOT set is a subset of 1600 videos, annotated at a lowered frame rate of 5 Hz.

Evaluation metrics. The most widely used metric to evaluate MOT methods is Multi-Object Tracking Accuracy (MOTA) [2]. However, the more recently introduced

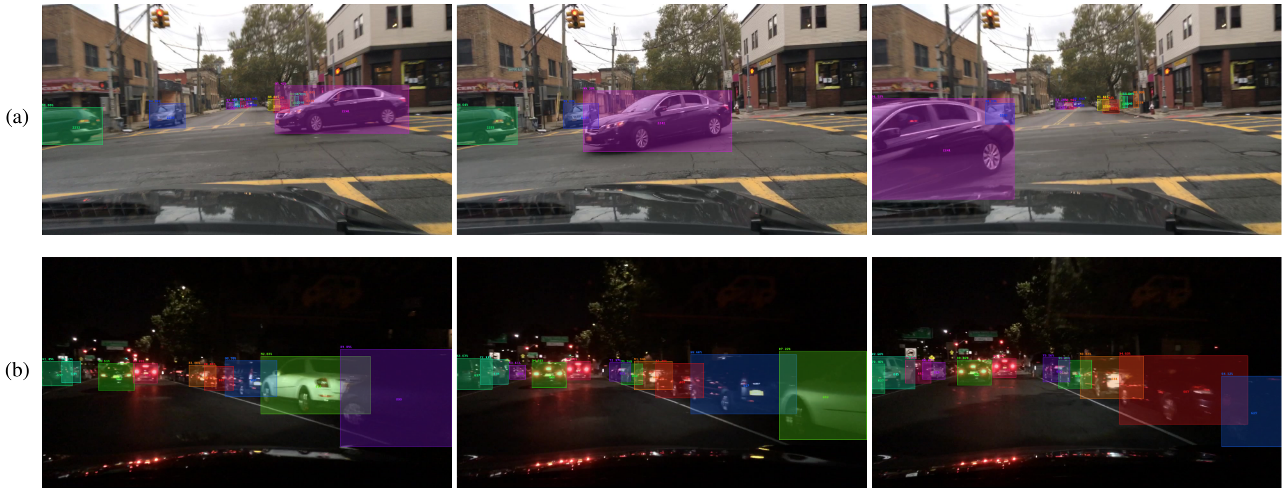


Figure 4. Predictions our model on the validation set of BDD100K, each color represents a different predicted ID. The method is robust to occlusions and in the nighttime.

Higher Order Tracking Accuracy (HOTA) [24] metric better balances detection and association. Since its introduction, many MOT benchmarks have adopted it, including MOT17 and BDD100K. Identity F1 Score (IDF1) [31] is also widely used to measure MOT systems and it focuses on object's identities association. While on MOT17 the main metrics are computed over all the classes together, BDD100K computes these metrics independently for each class and then computes an average. As a result, the main metrics are defined as mean Higher Order Tracking Accuracy (mHOTA), mean Multi-Object Tracking Accuracy (mMOTA) and mean Identity F1 Score (mIDF1).

4.2. Implementation details

Our method is built on top of Deformable-DETR [52] with a ResNet-50 [14] backbone. We also experiment with a more powerful Swin-L [22] backbone to demonstrate the scalability of our method. The architecture includes *mixed selection* and *look forward twice* refinements introduced in DINO [44]. Unless stated otherwise, we train the models with their default hyper-parameter sets. More details are provided in Appendix D. Our additional loss term is computed over every transformer decoder layer output, and the temperature value τ is set to 0.1. The *objectness threshold* is set to 0.5 during inference.

Our model is pre-trained on the object detection task with the procedure described in Section 3.4 with the default Deformable-DETR hyperparameters and augmentations. The contrastive loss coefficient is set to $\lambda_{\text{contr}} = 2$. We did not use the contrastive pre-training when pre-trained on COCO [21].

MOT17. We follow a similar procedure described in [50] by pre-training the model on CrowdHuman (CH) [32]. However, we do not simulate tracking frames. For simplicity of the training procedure, we skip the joint MOT17+CH dataset fine-tuning stage and only use MOT17. We pre-train the model for 50 epochs with a batch size of 8 and augment each image twice for the contrastive loss, leading to a total batch size of 16. In the training stage, the batches are built by sampling $N_v = 2$ videos and $N_f = 8$ frames, and λ_{contr} is set to 2. In addition, we allow past detected objects to be kept in memory for a maximum of $T = 20$ frames, and we evaluate our method under the private detection protocol.

BDD100K. Our model is pre-trained on the BDD100K detection dataset for 36 epochs. We use a batch size of 24 and again augment each image twice, leading to a total batch size of 48. Afterwards, we train it on the tracking set for 10 epochs and decrease the learning rate by a factor of 10 at epoch 8. Each training batch samples $N_v = 4$ videos and $N_f = 10$ frames. The contrastive loss coefficient is set to $\lambda_{\text{contr}} = 1$. In addition, we allow past detected objects to be kept in memory for a maximum of $T = 9$ frames.

4.3. Results

MOT17. We report quantitative results on the test set of MOT17 in Table 1. Our method outperforms comparable DETR-like models (bottom section) on HOTA, suggesting that our model has the best balance between detection and association. It also outperforms comparable DETR-like models on IDF1 which shows the effectiveness of the object representations learned through the instance-level contrastive loss.

BDD100K. For BDD100K, we report results on the test set in Table 2. Our method outperforms all methods by a significant margin on the mMOTA metric, including unpublished methods[1]. We further report results on the validation set (Table 3) to compare with MOTR [43] and

[1] The full leaderboard, including unpublished methods, is available on *eval.ai* [1].

Method	Backbone	Pre-train	Train	HOTA↑	MOTA↑	IDF1↑	IDS↓
CNN-based:							
CenterTrack [50]	DLA-34	CH	MOT17	52.2	67.8	64.7	3039
QDTrack [28]	ResNet-50	COCO	MOT17	—	68.7	66.3	3378
MTrack [41]	DLA-34	CH	MOT17	—	72.1	73.5	2028
FairMOT [47]	DLA-34	COCO	MOT17+CH+CP+ETHZ+CS+CT+PRW	59.3	73.7	72.3	3303
Swin-JDE [36]	ResNet-50	CH	MOT17+CP+ETHZ+CS+CT+PRW	57.2	71.7	71.3	2784
ByteTrack [46]	YOLOX-X	COCO	MOT17+CH+CP+ETHZ	63.1	80.3	77.3	2196
SUSHI [5]	YOLOX-X	COCO	MOT17+CH+CP+ETHZ	66.5	81.1	83.1	1149
Transformer-based:							
MO3TR PIQ-SST [51]	DarkNet	COCO	MOT17+CH	60.5	78.6	72.4	2808
TransCenter [40]	PVTv2	COCO	MOT17+CH	—	76.2	65.5	5394
MOTRv2 [48]	YOLOX-X+ResNet-50	COCO	MOT17+CH*+CP+ETHZ	62.0	78.6	75.0	2619
MO3TR [51]	ResNet-50	COCO	MOT17+CH+ETHZ+CS	49.9	63.9	60.5	2847
MeMOT [3]	ResNet-50	COCO	MOT17+CH*	56.9	72.5	69.0	2724
TrackFormer [25]	ResNet-50	CH	MOT17+CH	57.3	74.1	68.0	2829
MOTR [43]	ResNet-50	COCO	MOT17+CH*	57.8	73.4	68.6	**2439**
TransTrack [33]	ResNet-50	CH	MOT17+CH	—	**74.5**	63.9	3663
TransTrack [33]	ResNet-50	CH	MOT17	—	68.4	—	3942
ContrasTR (ours)	ResNet-50	CH	MOT17	**58.9**	73.7	**71.8**	2619

Table 1. Results on the MOT17 test set using private detections. The second group shows DETR-like models, all based on Deformable-DETR except MO3TR, which is based on DETR. The best results among DETR-like models using the ResNet-50 backbone are highlighted in **bold**. Datasets abbreviations refer to the following works ETHZ [9], CityPersons [45], CS [39], CT [7], PRW [49].

Method	Backbone	Pre-train	mHOTA↑	mMOTA↑	mIDF1↑	IDS↓
Yu et al. [42]	ResNet-101	—	—	26.3	44.7	14674
QDTrack [28]	ResNet-50	BDD100K	41.8	35.6	52.4	**10790**
TETer [20]	ResNet-50	BDD100K	—	37.4	53.3	—
ByteTrack [46]	YOLOX-X	COCO	—	40.1	55.8	15466
SUSHI [5]	YOLOX-X	COCO	**48.2**	40.2	**60.0**	13626
ContrasTR (ours)	Swin-L	BDD100K	46.1	**42.8**	56.5	10793

Table 2. Results on BDD100K test split, with an objectness threshold of 0.4. The best results are shown in **bold**.

MOTRv2 [48], since the methods were not evaluated on the test set. MOTRv2 outperforms our method when using an additional YOLOX [11] detector for object proposals, which highly reduces the number of misses. Nevertheless, our model with a ResNet-50 backbone outperforms MOTR [43] and MORTv2 [48]. To make a fair comparison, we also pre-train our model on COCO. Although we don't use contrastive pre-training in that setting, we still exceed MOTRv2's performance by +0.3 mMOTA and +1.1 mIDF1.

Limitations. Yet, our method does not achieve the same level of performance as the state-of-the-art tracking methods on MOT17. As shown in the table, differences can be attributed to extra training data, more powerful backbones, extra post-processing and more elaborate association strategies.

4.4. Ablation study

We conduct the ablation study on BDD100K [42] as the dataset is large and offers adequate variation. We pre-train our models on the detection set of BDD100K without the contrastive loss unless specified, and then we train on $1/8$ of the tracking training data by randomly subsampling 25% of the videos and 50% of the frames. The entire validation set is used for evaluation. We use a batch size of 40, $T = 5$ previous frames and an *objectness threshold* of 0.5. Due to limited computational resources, the ablations have not been run with the optimal parameters.

Sampling strategy. In this ablation, we investigate different variations of the number of sampled videos N_v and sampled frames N_f to show the effect of increasing one variable and simultaneously decreasing the other. For the experiment, we used a batch size of $N_v N_f = 40$ and a contrastive loss weighting of $\lambda_{\text{contr}} = 2$. Results are given in Table 4. As the number of sampled videos N_v decreases and the number of sampled frames per video N_f increases, the probability of sampling larger sets of positive examples rises. This explains the increasing tendency of tracking metrics that emphasize association: the contrastive loss can leverage a greater variety of positive and negative examples. However, when sampling all 40 frames within one video, the scenery variability decreases and so the variation in the set of negative examples dramatically drops, harming performance.

Method	Backbone	Pre-train	mHOTA↑	mMOTA↑	mIDF1↑	IDS↓
MOTR [43]	ResNet-50	COCO	—	32.0	43.5	**3493**
MOTRv2 [48]	ResNet-50	COCO	—	35.5	48.2	—
ContrasTR w/o CPT (ours)	ResNet-50	COCO	40.2	36.4	48.1	6067
ContrasTR (ours)	ResNet-50	BDD100K	**40.8**	**36.7**	**49.2**	6695
MOTRv2 [48]	YOLOX-X+ResNet-50	BDD100K+COCO	—	**43.6**	**56.5**	—
ContrasTR (ours)	Swin-L	BDD100K	44.4	41.7	52.9	6363

Table 3. Results on the BDD100K validation set, with an objectness threshold of 0.4. The best results are highlighted in **bold**.

N_v	N_f	mHOTA	mMOTA	mIDF1	mAP	IDSw
20	2	35.0	33.3	40.8	33.8	10116
8	5	35.6	33.3	41.6	33.8	8724
4	10	35.9	**33.7**	42.3	33.7	7869
2	20	**36.2**	33.6	**42.9**	33.7	7483
1	40	35.4	32.1	41.3	**34.0**	**7191**

Table 4. Influence of the number of frames sampled per video on BDD100K validation set. The batch size $N_v N_f$ is fixed at 40.

Contrastive loss weight. The contrastive loss parameter λ_{contr} also impacts performance. We experimented with different values and found values close to 0.5 to offer the best results. When the coefficient is too high, detection performance suffers, so overall tracking performance decreases. In practice, we found a slight shift in the optimal value when using contrastive pre-training and the whole training data.

λ_{contr}	mHOTA	mMOTA	mIDF1	mAP	IDSw
0.25	36.4	34.9	42.9	**34.8**	10105
0.5	**36.6**	**35.0**	**43.1**	34.7	9216
1	36.5	34.5	**43.1**	34.4	8502
2	35.9	33.7	42.3	33.7	7869
3	35.0	32.2	40.9	33.2	7693
4	34.3	31.0	39.8	32.7	**7550**

Table 5. Influence of the contrastive loss weighting on object detection and MOT metrics on BDD100K validation set. Classification and localization coefficients are kept fixed. A higher coefficient reduces the number of identity switches but comes at the cost of a lower mAP. All models have been trained with $N_f = 10$, $N_v = 4$.

Contrastive learning, pre-training and projection head. We ablate the effect of our contributions on the tracking metrics in Table 6. All models are trained with $N_f = 10$, $N_v = 4$ and $\lambda_{\text{contr}} = 1$. Without the contrastive loss, the model is not trained for tracking and produces many ID switches (fourth row). When the model is trained without the FFN projection head, the contrastive loss is applied directly over the embeddings produced by the transformer decoder. In this setup, the embeddings produced by the transformer decoder are used to generate the bounding boxes and class predictions through the respective heads and as tracking embeddings to re-identify objects over time. The performance gap (between the second and third row) shows that this additional head provides more flexibility to the network to generate the tracking embeddings and preserves detection performance. Pre-training the model with the contrastive loss on object detection using the methodology outlined in Section 3.4 improves tracking metrics significantly (first row). Finally, the performance difference between Deformable-DETR (last row) and ContrasTR (first row) is small in terms of mMOTA. This is because mMOTA overemphasizes the effect of detection performance [24].

Contr.	Proj.	CPT	mHOTA	mMOTA	mIDF1	mAP
✓	✓	✓	**38.0**	**35.1**	**45.1**	34.3
✓	✓	✗	36.5	34.5	43.1	**34.4**
✓	✗	✗	35.4	33.8	41.4	33.6
✗	✗	✗	33.4	32.9	38.3	34.3

Table 6. Ablation study of different method components on BDD100K validation set: the contrastive loss and the sampling strategy (Contr.); the tracking projection head (Proj.); the contrastive pre-training on object detection (CPT).

5. Conclusion

This work presents a joint object detection and MOT method, ContrasTR. Our model exploits DETR's object-level embeddings to learn robust object representations through an instance-level contrastive loss and a revised sampling strategy. The proposed method preserves detection performance and can turn existing DETR-like object detectors into trackers without needing complicated architectural design or additional heavy modules. We also present a highly scalable pre-training scheme to exploit detection-only datasets to boost performance further.

It matches the performance of comparable DETR-like methods on the MOT17 benchmark while using less additional training data. Furthermore, its performance surpasses the previous state-of-the-art by +2.6 mMOTA on the challenging BDD100K dataset. Thereby demonstrating its robustness in adverse weather conditions and under significant object motion.

Acknowledgements The authors thankfully acknowledge support by Toyota via the TRACE project. All the authors are also affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

References

[1] Bdd100k multiple object tracking challenge. https://eval.ai/web/challenges/challenge-page/1836/leaderboard/4312/mMOTA, 2023 (accessed August 21, 2023). 6

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 5

[3] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022. 2, 3, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 3

[5] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22877–22887, June 2023. 7

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 5

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE conference on computer vision and pattern recognition*, pages 304–311. IEEE, 2009. 7

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. 1

[9] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 7

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3, 7

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[15] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020. 2, 5

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*. 2, 5

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 2, 4

[18] Bruno Korbar and Andrew Zisserman. End-to-end tracking with a multi-query transformer. *arXiv preprint arXiv:2210.14601*, 2022. 2, 3

[19] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1, 3

[20] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild. In *European Conference on Computer Vision*, pages 498–515. Springer, 2022. 7

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 6

[23] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14668–14678, 2020. 2

[24] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixe, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision (IJCV)*, 2020. 6, 8

[25] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022. 2, 3, 7

[26] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking.

arXiv:1603.00831 [cs], Mar. 2016. arXiv: 1603.00831. 2, 3, 5

[27] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 1, 3

[28] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 164–173, 2021. 2, 7

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[31] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 17–35, Cham, 2016. Springer International Publishing. 6

[32] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 3, 6

[33] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 2, 3, 7

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 2, 5

[35] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10860–10869, 2021. 2

[36] Chi-Yi Tsai, Guan-Yu Shen, and Humaira Nisar. Swin-jde: joint detection and embedding multi-object tracking in crowded scenes based on swin-transformer. *Engineering Applications of Artificial Intelligence*, 119:105770, 2023. 2, 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[38] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer, 2020. 2

[39] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3376–3385, 2017. 7

[40] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense representations for multiple-object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–16, 2022. 2, 3, 7

[41] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8834–8843, 2022. 2, 7

[42] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 5, 7

[43] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022. 2, 3, 6, 7, 8

[44] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2022. 6

[45] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4457–4465, 2017. 7

[46] 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 *Computer Vision – ECCV 2022*, pages 1–21, Cham, 2022. Springer Nature Switzerland. 7

[47] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 2, 7

[48] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22056–22065, June 2023. 2, 3, 7, 8

[49] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1367–1376, 2017. 7

[50] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European conference on computer vision*, pages 474–490. Springer, 2020. 2, 6, 7, 12

[51] Tianyu Zhu, Markus Hiller, Mahsa Ehsanpour, Rongkai Ma, Tom Drummond, Ian Reid, and Hamid Rezatofighi. Looking beyond two frames: End-to-end multi-object tracking using spatial and temporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3, 7

[52] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers

for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 6