COOLer: Class-Incremental Learning for Appearance-Based Multiple Object Tracking

Zhizheng Liu*[0009−0006−9426−3718], Mattia Segu*[0000−0002−9107−531X], and Fisher Yu(✉)[0000−0001−8829−7344]

ETH Zürich, 8092 Zürich, Switzerland
{liuzhi, segum}@ethz.ch, i@yf.io

Abstract. Continual learning allows a model to learn multiple tasks sequentially while retaining the old knowledge without the training data of the preceding tasks. This paper extends the scope of continual learning research to class-incremental learning for multiple object tracking (MOT), which is desirable to accommodate the continuously evolving needs of autonomous systems. Previous solutions for continual learning of object detectors do not address the data association stage of appearance-based trackers, leading to catastrophic forgetting of previous classes' re-identification features. We introduce COOLer, a COntrastive- and cOntinual-Learning-based tracker, which incrementally learns to track new categories while preserving past knowledge by training on a combination of currently available ground truth labels and pseudo-labels generated by the past tracker. To further exacerbate the disentanglement of instance representations, we introduce a novel contrastive class-incremental instance representation learning technique. Finally, we propose a practical evaluation protocol for continual learning for MOT and conduct experiments on the BDD100K and SHIFT datasets. Experimental results demonstrate that COOLer continually learns while effectively addressing catastrophic forgetting of both tracking and detection. The project page is available at https://www.vis.xyz/pub/cooler.

Keywords: Continual learning · Multiple object tracking · Re-Identification.

1 Introduction

Continual learning aims at training a model to gradually extend its knowledge and learn multiple tasks sequentially without accessing the previous training data [5]. Since merely finetuning a pre-trained model on the new task would result in forgetting the knowledge learned from previous tasks - a problem known in literature as catastrophic forgetting [20]- ad-hoc continual learning solutions are required. As data distributions and practitioners' needs change over time, the practicality of continual learning has made it popular in recent years.

* Equal contribution.

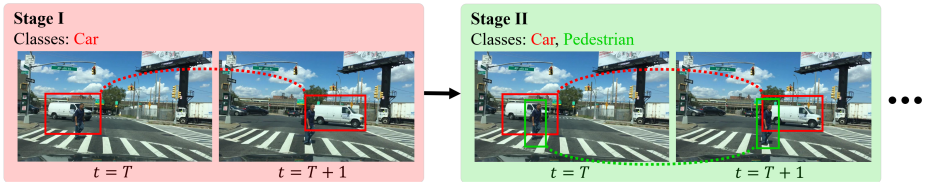

Fig. 1: Illustration of the class-incremental learning problem for multiple object tracking. In a first stage, an MOT model can only track cars (red). When given annotations only for the novel class 'pedestrian' (green), the objective is learning to track the new class without forgetting the previous one.

This paper addresses class-incremental learning for multiple object tracking (MOT), an important yet novel research problem that, to the best of our knowledge, has not been studied in previous literature. MOT tracks multiple objects simultaneously from a video sequence and outputs their location and category [2]. While prior work [27] explored domain adaptation of MOT to diverse conditions, continual learning for MOT would provide a flexible and inexpensive solution to incrementally expand the MOT model to new classes according to the changing necessities. For example, as illustrated in Fig. 1, one can train an MOT model to track cars and then expand its functionality to track pedestrians with new training data only annotated for pedestrians.

Following the tracking-by-detection paradigm [23], most MOT systems first detect object locations and classes via an object detector, and then associate the detected instances across frames via a data association module. State-the-art trackers often use a combination of motion and appearance cues in their association module [1,30,35]. While motion cues are straight-forward to use with simple heuristics, appearance cues are used for object re-identification (Re-ID) and are more robust to complex object motion and large object displacement across adjacent frames. Appearance-based association typically requires a Re-ID module [8,21,31] for learning Re-ID features. However, it is crucial to make such learned appearance representations flexible to incrementally added categories. Training the appearance extractors only on the new classes would indeed results in catastrophic forgetting of Re-ID features for older classes, and degrade the association performance (Tab. 2, Fine-tuning). Although previous work [22,28,37] explores class-incremental learning of object detectors, these approaches are sub-optimal for MOT by not addressing the data association stage.

To address this problem, we introduce COOLer, a COntrastive- and cOntinual-Learning-based multiple object tracker. Building on the state-of-the-art appearance-based tracker QDTrack [21], COOLer represents the first comprehensive approach for continual learning for appearance-based trackers by addressing class-incremental learning of both the building blocks of an MOT system, *i.e.* object detection and data association. To continually learn to track new categories while preventing catastrophic forgetting, we propose to combine the available ground truth labels from the newly added categories with the association pseudo-labels

and the temporally-refined detection pseudo-labels generated by the previous-stage tracker on the new training data. Furthermore, adding classes incrementally without imposing any constraint may cause overlapping instance representations from different classes, blurring the decision boundaries and leading to misclassifications. While traditional contrastive learning can disentangle the representations of different classes, they undermine the intra-class discrimination properties of the instance embeddings for data association. To this end, we propose a novel contrastive class-incremental instance representation learning formulation that pushes the embedding distributions of different classes away from each other while keeping the embedding distributions of the same class close to a Gaussian prior. To assess the effectiveness of continual learning strategies for MOT, we propose a practical and comprehensive evaluation protocol and conduct extensive experiments on the BDD100K [34] and SHIFT [29] datasets.

We demonstrate that COOLer can alleviate forgetting of both tracking and detection, while effectively acquiring incremental knowledge. Our key contributions are: (i) we introduce COOLer, the first comprehensive method for class-incremental learning for multiple object tracking; (ii) we propose to use the previous-stage tracker to generate data association pseudo-labels to address catastrophic forgetting of association of previous classes and leverage the temporal information to refine detection pseudo-labels; (iii) we introduce class-incremental instance representation learning to disentangle class representations and further improve both detection and association performance.

2 Related Work

Continual learning aims at learning new knowledge continually while alleviating forgetting. Various continual learning strategies have been proposed, including model growing [26], regularization [14,16], parameter isolation [19], and replay [24]. We here discuss related literature in continual learning for object detection, unsupervised Re-ID learning, and contrastive representation learning.

Continual Learning for Object Detection. Shmelkov et al. [28] propose the first method for continual learning for object detection. It uses the old model as the teacher model which generates pseudo labels for the classification and bounding box regression outputs to prevent forgetting. Later works [17,22] follow this diagram by incorporating the state-of-the-art detectors such as Faster R-CNN [25] and Deformable DETR [38]. While our work also uses detection pseudo-labels, we refine them temporally by leveraging a multiple-object tracker.

Unsupervised Re-ID Learning. As annotating instance IDs is laborious and time-consuming, unsupervised Re-ID learning proposes to learn data association from video sequences without annotations given only a pre-trained detector [13]. Most unsupervised Re-ID learning approaches generate pseudo-identities to train the association module from a simple motion-based tracker [13], image clustering [9,15,32] or contrastive learning of instance representation under data augmentation [27]. In contrast, our class-incremental instance representation learning

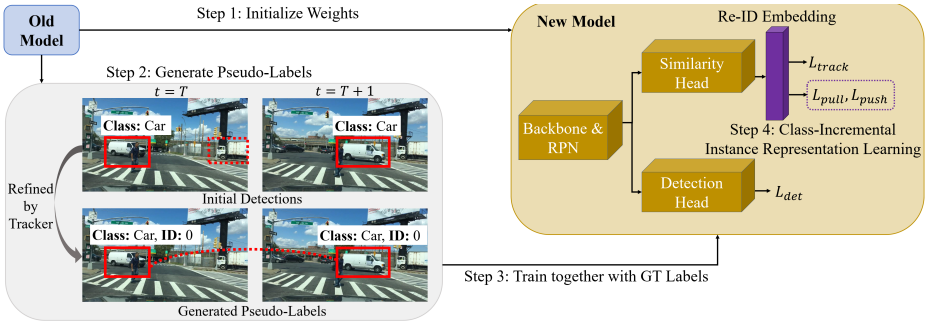

Fig. 2: Pipeline of COOLer for class-incremental learning. 1). Initialize the new model with weights from the old model. 2). Use the previous tracker to refine the initial detections and generate detection and association pseudo-labels. 3). Append them to the ground truth labels to train the new model jointly. 4). During training, apply class-incremental instance representation learning.

approach handles a combination of labeled and unlabeled Re-ID data to continually learn Re-ID of new categories' instances without forgetting the old ones.

Contrastive Representation Learning. Contrastive learning [4,10] aims to attract representations of similar samples and push away representations of dissimilar ones. Previous continual learning methods leverage contrastive learning at a category level. Mai et al. [18] propose supervised contrastive replay and use a nearest-class-mean classifier instead of the softmax classifier. Co2L [3] shows that contrastively-learned representations are more robust to catastrophic forgetting than ones trained with cross-entropy. OWOD [12] introduces a memory queue for updating the class mean prototype vector during training to help contrastive learning. However, such class-contrastive formulations often collapse intra-class representations, hindering Re-ID-based data association in MOT. Our class-incremental instance representation learning approach maintains intra-class variability with a contrastive loss that estimates standard deviation prototype vectors for each class and keeps the class distribution close to a prior.

3 Method

We define the continual learning problem for MOT (Sec. 3.2). We then provide an overview of COOLer (Sec. 3.2) and introduce its two key components, namely continual pseudo-label generation for detection and data association (Sec. 3.3), and class-incremental instance representation learning (Sec. 3.4).

3.1 Problem Definition

We define continual learning for MOT as a class-incremental learning (CIL) problem over a sequence of B training stages $\{\mathcal{S}^0, \mathcal{S}^1, ..., \mathcal{S}^{B-1}\}$, where at each

stage b a set of categories Y_b is introduced. $\mathcal{S}^b = \{\mathcal{X}^b, \mathcal{D}^b, \mathcal{T}^b\}$ is the set of training videos $\mathcal{X}^b$, detection labels $\mathcal{D}^b$, and tracking labels $\mathcal{T}^b$ for a set of categories Y_b at stage b. Although typical CIL assumes no overlapping classes in different tasks b and b', it is common in real-world applications to observe old classes in new stages [33]. Thus, we assume that categories from another stage b' may occur again at b despite not being in the annotation set, i.e. $Y_b \cap Y_{b'} = \emptyset$. The goal is continually learning an MOT model that can track Y_b without forgetting to track $\bar{Y}_{b-1} = Y_0 \cup ... \cup Y_{b-1}$. During each stage b, only data $\mathcal{S}^b$ can be accessed. After each training stage b, the model is evaluated over all seen classes $\mathcal{Y}_b = Y_0 \cup ... \cup Y_b$.

3.2 COOLer

Architecture. COOLer's architecture is based on the representative appearance-based tracker QDTrack [21], which consists of a Faster R-CNN [25] object detector and a similarity head to learn Re-ID embeddings for data association.
Base Training. Given the data $\mathcal{S}^0 = \{\mathcal{X}^0, \mathcal{D}^0, \mathcal{T}^0\}$ from the first stage $b = 0$, we train the base model ϕ^0 following QDTrack. Let $\hat{\mathcal{D}}^0$ be the detector predictions and $\hat{\mathcal{V}}^0$ their corresponding Re-ID embeddings. QDTrack is optimized end-to-end with a detection loss $\mathcal{L}_{\text{det}}$ to train the object detector, and a tracking loss $\mathcal{L}_{\text{track}}$ to learn the Re-ID embeddings for data association. $\mathcal{L}_{\text{det}}$ is computed from $\mathcal{D}^0$ and $\hat{\mathcal{D}}^0$ as in Faster R-CNN [25]. As for the tracking loss $\mathcal{L}_{\text{track}}$, QDTrack first samples positive and negative pairs of object proposals in adjacent frames using $\mathcal{D}^0$, $\mathcal{T}^0$, and $\hat{\mathcal{D}}^0$. Then, $\mathcal{L}_{\text{track}}$ is computed from a contrastive loss using the Re-ID embeddings $\hat{\mathcal{V}}^0$ of the sampled proposals to cluster object embeddings of the same IDs and separate embeddings of different instances. Refer to the original QDTrack paper [21] for more details. The final loss is:

$$\mathcal{L}^0 = \mathcal{L}_{\text{det}}(\hat{\mathcal{D}}^0, \mathcal{D}^0) + \mathcal{L}_{\text{track}}(\hat{\mathcal{D}}^0, \hat{\mathcal{V}}^0, \mathcal{D}^0, \mathcal{T}^0). \tag{1}$$

Continual Training. Given the old model ϕ^{b-1} trained up to the stage $b-1$, and the new data $\mathcal{S}^b$ for the stage b, COOLer is the first tracker to incrementally learn to track the new classes Y_b without forgetting the old ones $\bar{Y}_{b-1}$. We propose a continual pseudo-label generation strategy for MOT (Sec. 3.3) that uses the previous tracker ϕ^{b-1} to generate pseudo-labels $\{\bar{\mathcal{D}}^b_{\text{old}}, \bar{\mathcal{T}}^b_{\text{old}}\}$ for the old classes $\bar{Y}_{b-1}$, and combine them with the ground-truth labels $\{\mathcal{D}^b_{\text{new}}, \mathcal{T}^b_{\text{new}}\}$ for the new ones Y_b to train the new model ϕ^b. To further disentangle the Re-ID embedding space for different classes and instances, we propose a novel class-incremental instance representation learning approach (Sec. 3.4). See Fig. 2 for an overview.

3.3 Continual Pseudo-label Generation for Tracking

While training with detection pseudo-labels generated by the previous object detector has proven effective against catastrophic forgetting in CIL of object detection [22,36], detection pseudo-labels lack the instance association information, which is crucial to learn the Re-ID module in appearance-based MOT. We instead propose to use the MOT model ϕ^{b-1} from the previous stage $b-1$ to

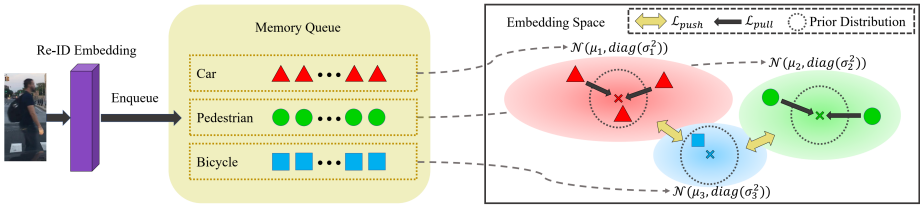

Fig. 3: Illustration of our class-incremental instance representation learning. We keep a memory queue to update the class embedding distributions. The contrastive loss includes an inter-class pushing loss and an intra-class pulling loss.

simultaneously generate temporally-refined detection pseudo-labels $\bar{\mathcal{D}}^b_{\text{old}}$ and instance association pseudo-labels $\bar{\mathcal{T}}^b_{\text{old}}$ for the old classes $\bar{Y}_{b-1}$ in the new stage b. We then train the new tracker ϕ^b on the union of the pseudo-labels $\{\bar{\mathcal{D}}^b_{\text{old}}, \bar{\mathcal{T}}^b_{\text{old}}\}$ for old classes $\bar{Y}_{b-1}$ and ground-truth labels $\{\mathcal{D}^b_{\text{new}}, \mathcal{T}^b_{\text{new}}\}$ for new classes Y_b:

$$\mathcal{L}^b_{\text{pseudo}} = \mathcal{L}_{\text{det}}(\hat{\mathcal{D}}^b, \bar{\mathcal{D}}^b_{\text{old}} \cup \mathcal{D}^b_{\text{new}}) + \mathcal{L}_{\text{track}}(\hat{\mathcal{D}}^b, \hat{\mathcal{V}}^b, \bar{\mathcal{D}}^b_{\text{old}} \cup \mathcal{D}^b_{\text{new}}, \bar{\mathcal{T}}^b_{\text{old}} \cup \mathcal{T}^b_{\text{new}}). \quad (2)$$

It is worth noticing that, unlike detection pseudo-labels in [22,36], our detection pseudo-labels are temporally refined by the tracking algorithm, resulting in a reduced number of false positives and in recovery of initially missed detections. Moreover, the pseudo-identities $\bar{\mathcal{T}}^b_{\text{old}}$ alleviate catastrophic forgetting in data association by training the similarity head on old classes $\bar{Y}_{b-1}$.

3.4 Class-Incremental Instance Representation Learning

Our class-incremental learning strategy based on tracking pseudo-labels (Sec. 3.3) enforces that each instance must be well-separated from others in the embeddings space, but does not constrain where Re-ID features for each class are projected, potentially leading to entangled class distributions that hurt both detection and tracking performance. Previous CIL approaches [3,12,18] ensure separation of class distributions by applying class-contrastive losses during incremental learning. However, naively applying contrastive learning on the instance embedding space would cause the distribution of a class' embeddings to collapse to a single point, undermining the intra-class discrimination properties of the learned Re-ID embeddings necessary for effective data association.

To this end, we introduce a novel contrastive loss for class-incremental instance representation learning that disentangles embeddings of different classes while maintaining the intra-class variability of the embeddings (Fig. 3).

Class Prototype Vectors. First, we model variability of instance embeddings within each class c by approximating each class' embedding distribution as a Gaussian $\mathcal{N}(\boldsymbol{\mu}_c, \text{diag}(\boldsymbol{\sigma}_c^2))$, whose class mean prototype vector $\boldsymbol{\mu}_c$ and class standard deviation prototype vector $\boldsymbol{\sigma}_c$ are approximated online as the exponential moving average of a memory queue with limited size N_{queue} that stores exemplary class embeddings. See Supplement Sec. A for details on the memory queue.

Contrastive Loss. Our contrastive loss consists of a pushing term $\mathcal{L}_{\text{push}}$ that pushes distributions of different classes away from each other, and a pulling term $\mathcal{L}_{\text{pull}}$ that keeps the class distribution close to a prior, ensuring intra-class variability. We derive such losses from the Bhattacharyya distance D_B, which measures the similarity between distributions $\mathcal{N}(\boldsymbol{\mu}_{c_1}, \text{diag}(\boldsymbol{\sigma}_{c_1}^2))$ and $\mathcal{N}(\boldsymbol{\mu}_{c_2}, \text{diag}(\boldsymbol{\sigma}_{c_2}^2))$ of two classes c_1 and c_2. Their Bhattacharyya distance is:

$$D_B(c_1, c_2) = \frac{1}{8}(\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2})^T \boldsymbol{\Sigma}_{c_1,c_2}^{-1}(\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2}) + \frac{1}{2}\ln\frac{\det \boldsymbol{\Sigma}_{c_1,c_2}}{\sqrt{\det \boldsymbol{\Sigma}_{c_1} \det \boldsymbol{\Sigma}_{c_2}}}, \quad (3)$$

where $\boldsymbol{\Sigma}_{c_1,c_2} = \frac{\boldsymbol{\Sigma}_{c_1}+\boldsymbol{\Sigma}_{c_2}}{2}$. As it is hard to back-propagate gradients for the prototype mean and standard deviation $\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c$, we additionally introduce the per-batch embedding mean $\bar{\boldsymbol{\mu}}_c$ and standard deviation $\bar{\boldsymbol{\sigma}}_c$ for class c:

$$\bar{\boldsymbol{\mu}}_c = \frac{1}{N_c}\sum_{i=1}^{N_c} \boldsymbol{v}_{c,i}, \bar{\boldsymbol{\sigma}}_c = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}(\boldsymbol{v}_{c,i} - \boldsymbol{\mu}_c)^2}, \quad (4)$$

where N_c denotes the number of embedding vectors for class c in the current batch and $\boldsymbol{v}_{c,i}$ is the i_{th} embedding vector for class c.

Pushing Loss. For the pushing loss, the distance between the two class distributions is derived from the first term of Eqn. 3 as:

$$D_{\text{push}}(c_1, c_2) = \sqrt{(\bar{\boldsymbol{\mu}}_{c_1} - \boldsymbol{\mu}_{c_2})^T \boldsymbol{\Sigma}_{c_1,c_2}^{-1}(\bar{\boldsymbol{\mu}}_{c_1} - \boldsymbol{\mu}_{c_2})}. \quad (5)$$

We use the following hinge-based pushing loss to separate the two distributions:

$$\mathcal{L}_{\text{push}} = \frac{1}{C(C-1)}\sum_{c_1=1}^{C}\sum_{\substack{c_2=1 \\ c_2 \neq c_1}}^{C}[\Delta_{\text{push}} - D_{\text{push}}(c_1, c_2)]_+^2, \quad (6)$$

where C is the number of classes, Δ_{push} is the hinge factor, and $[x]_+ = \max(0, x)$.

Pulling Loss. For the pulling loss, we introduce a prior Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_c, \text{diag}(\boldsymbol{\sigma}_p^2))$ for each class, which has the same class mean prototype vector $\boldsymbol{\mu}_c$ while the standard deviation $\boldsymbol{\sigma}_p$ is fixed. We derive the distance between the class distribution and the prior distribution from the second term of Eqn. 3 as:

$$D_{\text{pull}}(c, p) = \frac{1}{2}\left(\sum_{j=1}^{N_d}\ln\left(\frac{\bar{\sigma}_{c,j}^2 + \sigma_{p,j}^2}{2}\right) - \sum_{j=1}^{N_d}\ln(\bar{\sigma}_{c,j}\sigma_{p,j})\right), \quad (7)$$

where N_d is the dimension of the embedding. We find that directly applying Eqn. 7 as the pulling loss will lead to numerical instability during optimization, as the logarithm operator is non-convex. We propose the following surrogate based on the $\mathcal{L}_2$ distance for a smoother optimization landscape as follows:

$$\mathcal{L}_{\text{pull}} = \frac{1}{C}\sum_{c=1}^{C}\sum_{j=1}^{N_d}(\bar{\sigma}_{c,j} - \sigma_{p,j})^2. \quad (8)$$

Total Loss. Finally, we extend Eqn. 2 with our pulling and pushing contrastive losses to learn the tracking model ϕ^b at stage b:

$$\mathcal{L}^b = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{track}} + \beta_1\mathcal{L}_{\text{pull}} + \beta_2\mathcal{L}_{\text{push}}, \quad (9)$$

where β_1 and β_2 are weights for the pushing and the pulling loss respectively.

4 Evaluation Protocol

We introduce a protocol for evaluating algorithms for class-incremental MOT.

Datasets. We use the BDD100K [34] and SHIFT [29] tracking datasets for evaluation. BDD100K is a large-scale real-world driving dataset for MOT with 8 classes. SHIFT is a large-scale synthetic driving dataset for MOT with 6 classes. Because of the size of the SHIFT dataset, training multiple stages on it is not feasible with modest computational resources. To ensure practicality for all researchers', we propose to only use its clear-daytime subset. The detailed class statistics for each dataset are reported in Tab. 1. Note that other popular MOT datasets are unsuitable for our setting. The MOT20 dataset [7] has very few categories. While TAO [6] has hundreds of classes, due to the scarcity of annotations it is intended as an evaluation benchmark and is not suitable for CIL.

Table 1: Class frequencies for the training splits of BDD100K [34] and SHIFT's [29] clear-daytime subset.

Dataset	Car	Ped	Truck	Bus	Bike	Rider	Motor	Train
BDD100K [34]	2098517	369798	149411	57860	25565	20107	12176	1620
SHIFT [29]	677580	749640	145940	65367	52267	-	74469	-

Protocol. The choice of the class ordering during incremental stages may largely impact results and observations. Object detection benchmarks typically add classes by their alphabetical order. However, in real-world MOT applications the annotation order often depends by (i) class frequency or (ii) semantic grouping. We hence propose the practical class splits to mirror the practitioners' needs. First, we propose two *frequency-based splits*. The Most→Least (**M→L**) split incrementally adds classes one-by-one from the most to the least frequent class according to Tab. 1. General→Specific (**G→S**) only evaluates one incremental step by dividing the classes into two groups: the first half of the most populated classes (General) and the remainder (Specific). Then, we propose a *semantic split*. We group classes into three super-categories according to their semantic similarity: vehicles, bikes, and humans. Therefore, we experiment on the Vehicle→Bike→Human (**V→B→H**) setting with two incremental steps.

Taking BDD100K as example, in the **M→L** setting the classes are added as follows: car→pedestrian→truck→bus→bike→rider→motor→train. In the **G→S** setting, the model is first trained on {car, pedestrian, truck, bus}, and then {bike, rider, motor, train} are added at once. In the **V→B→H** setting, the classes are added as follows: {car, truck, bus, train}→{bike, motor}→{pedestrian, rider}.

5 Experiments

5.1 Baselines

Since no prior work studied class-incremental learning for multiple objcet tracking, we compare COOLer with the following baseline methods:
Fine-tuning. In each incremental step, the model is trained only on the training data of the new classes, without addressing catastrophic forgetting.

Table 2: **Class-incremental Learning on BDD100K.** We conduct experiments on **M→L**, **G→S** and **V→B→H** settings. We compare COOLer with the Fine-tuning, Distillation, Det PL baselines and the oracle tracker.

Setting Stage (+New Classes)	Method	All Classes						
		mMOTA	mHOTA	mIDF1	MOTA	HOTA	IDF1	mAP
M→L Stage 0 (Car)		67.6	62.1	73.3	67.6	62.1	73.3	58.7
M→L Stage 1 (+Pedestrian)	Fine-tuning	15.6	21.8	27.7	4.5	19.1	14.1	19.9
	Distillation	46.4	51.7	63.0	61.8	59.0	70.1	47.1
	Det PL	46.7	49.5	59.2	56.3	53.9	61.8	46.8
	COOLer	**54.2**	**52.6**	**64.3**	**62.7**	**59.5**	**70.5**	**47.4**
	Oracle	57.4	53.6	65.9	65.1	59.9	71.5	48.3
M→L Stage 2 (+Truck)	Fine-tuning	-11.5	12.8	14.0	-2.2	13.3	6.2	11.7
	Distillation	27.3	47.8	57.3	56.9	56.6	67.4	**42.8**
	Det PL	34.9	47.1	55.6	57.3	55.5	65.1	42.5
	COOLer	**42.8**	**49.2**	**59.6**	**58.6**	**57.9**	**68.7**	42.6
	Oracle	49.8	50.8	62.1	63.2	58.9	70.4	45.0
M→L Stage 3 (+Bus)	Fine-tuning	-24.0	9.2	9.3	-2.0	8.6	2.5	9.9
	Distillation	-11.2	43.4	50.8	54.0	55.1	65.6	40.8
	Det PL	14.1	43.1	49.1	53.6	53.4	61.5	40.9
	COOLer	**34.0**	**47.9**	**57.3**	**55.8**	**56.8**	**67.4**	**41.9**
	Oracle	45.4	50.1	60.9	62.5	58.7	70.2	44.5
M→L Stage 4 (+Bicycle)	Fine-tuning	-16.0	5.6	6.5	-0.8	4.2	0.8	4.6
	Distillation	-19.4	40.1	47.1	51.9	53.2	63.6	35.1
	Det PL	3.6	37.6	42.4	41.2	43.0	46.2	34.3
	COOLer	**28.6**	**44.4**	**53.9**	**53.2**	**55.7**	**66.1**	**36.5**
	Oracle	41.3	47.1	58.0	62.2	58.5	69.9	39.9
G→S Stage 0 (General)		45.6	50.3	61.1	62.4	59.0	70.2	44.6
G→S Stage 1 (+Specific)	Fine-tuning	-24.7	11.7	14.1	-0.5	5.7	1.5	8.4
	Distillation	-32.9	35.4	42.4	59.6	57.3	68.3	29.5
	Det PL	6.0	34.9	41.5	54.1	52.0	59.6	27.9
	COOLer	**28.6**	**38.5**	**48.2**	**60.5**	**58.2**	**69.3**	**29.9**
	Oracle	30.4	38.9	49.0	61.8	58.7	70.0	30.9
V→B→H Stage 0 (Vehicle)		33.1	39.2	46.8	65.1	61.0	72.3	35.0
V→B→H Stage 1 (+Bike)	Fine-tuning	-27.5	9.7	11.2	-0.6	4.9	1.2	7.3
	Distillation	-30.5	34.1	39.6	62.9	59.7	70.6	**29.1**
	Det PL	-3.3	31.3	37.0	53.7	51.4	57.9	26.7
	COOLer	**24.4**	**36.8**	**44.8**	**63.1**	**60.2**	**70.9**	**29.1**
	Oracle	27.6	38.5	47.8	64.3	60.4	71.6	30.8
V→B→H Stage 2 (+Human)	Fine-tuning	7.4	10.0	13.1	4.4	18.2	13.4	8.2
	Distillation	14.8	35.8	43.9	55.9	55.9	66.4	27.9
	Det PL	15.5	34.6	41.5	52.1	51.4	58.7	27.4
	COOLer	**27.1**	**37.5**	**46.8**	**59.2**	**57.8**	**68.8**	**28.8**
	Oracle	30.4	38.9	49.0	61.8	58.7	70.0	30.9

Distillation. We design a distillation baseline based on Faster-ILOD [22], a state-of-the-art class-incremental object detection method that uses distillation losses from a teacher model of the previous stage to alleviate forgetting. To

further address forgetting in data association, we add the following distillation loss on the similarity head of QDTrack to enforce the cosine similarity between teacher and student embeddings for the old classes:

$$\mathcal{L}_{\text{track}}^{\text{dist}} = \left(\frac{\mathbf{v}_{\text{teacher}} \cdot \mathbf{v}_{\text{student}}}{\|\mathbf{v}_{\text{teacher}}\|_2 \cdot \|\mathbf{v}_{\text{student}}\|_2} - 1\right)^2, \tag{10}$$

where $\mathbf{v}_{\text{teacher}}$ and $\mathbf{v}_{\text{student}}$ are the Re-ID embeddings of the teacher and the student model, computed from the same proposals sampled for Faster-ILOD's ROI head distillation. The final loss is then $\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{track}} + \mu_1 \mathcal{L}_{\text{det}}^{\text{dist}} + \mu_2 \mathcal{L}_{\text{track}}^{\text{dist}}$, where $\mathcal{L}_{\text{det}}^{\text{dist}}$ is the detection distillation loss in [22], and μ_1, μ_2 are set to 1.

Detection Pseudo-Labels (Det PL). We compare against a baseline that only trains on the joint set of ground-truth labels for the new classes and high-confident (> 0.7) detection pseudo-labels from the old detector for the old classes. Unlike our method, this baseline does not temporally refine the detection pseudo-labels with the tracker, and does not provide association pseudo-labels.

Oracle. We compare the result with an oracle tracker trained in a single stage on the ground truth annotations of all classes.

Table 3: **Class-incremental Learning on SHIFT.** We conduct experiments on **M→L**, **G→S** and **V→B→H** settings. We compare COOLer with the Fine-tuning and Det PL baselines and the oracle tracker.

Setting Stage (+New Classes)	Method	All Classes						
		mMOTA	mHOTA	mIDF1	MOTA	HOTA	IDF1	mAP
M→L Stage 0 (Pedestrian)		53.7	46.1	54.4	53.7	46.1	54.4	43.0
M→L Stage 1 (+Car)	Fine-tuning	25.8	28.5	30.9	24.2	40.5	37.7	25.3
	Det PL	44.6	46.8	49.7	44.2	47.2	49.1	45.6
	COOLer	**50.9**	**50.9**	**57.0**	**50.9**	**51.0**	**56.8**	**45.7**
	Oracle	53.7	51.8	58.7	53.7	51.4	58.4	46.2
M→L Stage 2 (+Truck)	Fine-tuning	11.7	18.0	19.6	2.7	16.9	8.1	15.5
	Det PL	34.6	44.7	45.9	33.5	43.0	40.4	**44.8**
	COOLer	**45.2**	**51.5**	**57.3**	**48.2**	**50.6**	**56.3**	**44.8**
	Oracle	52.6	53.5	60.7	53.5	51.8	58.9	46.0
G→S Stage 0 (General)		50.8	53.1	60.0	52.4	51.9	58.2	46.0
G→S Stage 1 (+Specific)	Fine-tuning	19.4	24.1	26.9	4.8	18.1	10.1	20.0
	Det PL	45.8	49.8	55.4	48.7	48.9	52.7	**43.2**
	COOLer	**46.0**	**50.8**	**57.0**	**50.8**	**51.4**	**57.5**	42.7
	Oracle	48.8	51.1	57.5	52.5	51.9	58.5	43.8
V→B→H Stage 0 (Vehicle)		47.2	52.1	57.4	51.9	56.4	61.4	45.2
V→B→H Stage 1 (+Bike)	Fine-tuning	16.1	20.4	22.4	5.9	20.5	12.2	16.6
	Det PL	39.4	47.1	51.0	41.4	47.4	48.6	42.0
	COOLer	**44.5**	**51.3**	**57.5**	**49.5**	**55.5**	**60.9**	**42.3**
	Oracle	47.8	52.1	58.0	51.5	55.5	60.8	44.2
V→B→H Stage 2 (+Human)	Fine-tuning	8.9	7.7	9.1	23.0	31.2	30.9	7.1
	Det PL	37.3	41.4	43.7	39.8	40.9	42.6	41.7
	COOLer	**47.0**	**50.6**	**57.5**	**50.7**	**51.3**	**57.7**	**42.2**
	Oracle	48.8	51.1	57.5	52.5	51.9	58.5	43.8

5.2 Implementation Details

COOLer's architecture is based on QDTrack with a ResNet-50 [11] backbone. The model is optimized with SGD with momentum of 0.9 and weight decay of 1e-4. We train the network with 8 NVIDIA 2080Ti GPUs with a total batch size of 16. For all experiments, we train for 6 epochs in each incremental stage. The initial learning rate is 0.02 and decayed to 0.002 after 4 epochs, and to 0.0002 after 5. For BDD100K experiments, the weight for the contrastive losses β_1, β_2 are 0.01; for SHIFT experiments, β_1, β_2 are 0.001. The hinge factor for the pushing loss Δ_{push} is set to 15.0. For the prior distribution $\mathcal{N}(\boldsymbol{\mu}_c, \text{diag}(\boldsymbol{\sigma}_p^2))$ of the pulling loss, we use $\boldsymbol{\sigma}_p = 0.05 \cdot \vec{\mathbf{1}}$, where $\vec{\mathbf{1}}$ is the unit vector. We select hyper-parameters from a grid search, and report sensitivity analysis in the supplement.

Table 4: **Ablation Study on Method Components.** We ablate on the choice of pseudo-labels (PL) and contrastive (CT) loss for COOLer on **M→L** setting on BDD100K. We compare training with our pseudo-labels generated by the tracker (Track) and the pseudo-labels generated by the detector (Det). We also compare our contrastive loss (Ours) with the contrastive loss proposed in OWOD [12].

Setting Stage (+New Classes)	Components		All Classes						
	PL	CT Loss	mMOTA	mHOTA	mIDF1	MOTA	HOTA	IDF1	mAP
M→L Stage 1 (+Pedestrian)	Det	✗	46.7	49.5	59.2	56.3	53.9	61.8	46.8
	Track	✗	54.1	**52.6**	**64.5**	62.5	59.3	70.3	47.2
	Track	OWOD [12]	53.7	52.2	63.9	62.1	58.9	69.8	47.2
	Track	Ours	**54.2**	**52.6**	64.3	**62.7**	**59.5**	**70.5**	**47.4**
M→L Stage 2 (+Truck)	Det	✗	34.9	47.1	55.6	57.3	55.5	65.1	42.5
	Track	✗	**43.4**	49.1	59.5	58.2	57.5	68.3	**42.8**
	Track	OWOD [12]	41.9	48.7	58.9	57.6	57.3	68.0	**42.8**
	Track	Ours	42.8	**49.2**	**59.6**	**58.6**	**57.9**	**68.7**	42.6
M→L Stage 3 (+Bus)	Det	✗	14.1	43.1	49.1	53.6	53.4	61.5	40.9
	Track	✗	32.8	47.6	56.9	54.5	56.3	66.8	**41.9**
	Track	OWOD [12]	33.1	47.5	56.9	53.9	56.1	66.6	**41.9**
	Track	Ours	**34.0**	**47.9**	**57.3**	**55.8**	**56.8**	**67.4**	**41.9**
M→L Stage 4 (+Bicycle)	Det	✗	3.6	37.6	42.4	41.2	43.0	46.2	34.3
	Track	✗	25.7	43.9	52.9	50.8	55.1	65.4	36.1
	Track	OWOD [12]	24.7	43.7	52.7	49.8	54.8	65.0	36.1
	Track	Ours	**28.6**	**44.4**	**53.9**	**53.2**	**55.7**	**66.1**	**36.5**

5.3 Experimental Results

We compare our method to the above-mentioned baselines on the BDD100K and SHIFT datasets. We evaluate the mAP for object detection, and representative tracking metrics for MOT. mMOTA, mHOTA, mIDF1 are averaged across category-specific metrics, while MOTA, HOTA, IDF1 are the overall metrics.

BDD100K. Table 2 shows the results on the BDD100K dataset. In the **M→L** setting, we show results up to stage 4 (+Bicycle) due to space constraints, and report full results in the supplement. COOLer achieves the best tracking performance among all methods and in all settings. In (**M→L**, Stage 4) COOLer

Fig. 4: Qualitative Results of the Det PL baseline and COOLer on a validation video sequence of BDD100K in the fourth step of the **M**→**L** setting (+Bicycle). Different bounding box colors represent different classes, and the number above the bound box denotes the instance ID. Best viewed in color with zoom.

has a noteworthy 48.0%, and 25.0% mMOTA improvement compared to the Distillation and Det PL baselines, showing its effectiveness in class-incremental tracking. Besides boosting the tracking performance, COOLer also improves continual object detection. Compared to the Distillation baseline based on the class-incremental object detector Faster-ILOD [22], COOLer achieves higher mAP thanks to our temporally-refined detection pseudo-labels. COOLer obtains +1.4% and +2.2% mAP wrt. Distillation and Det PL in (**M**→**L**, Stage 4), and +0.9% and +1.4% mAP wrt. Distillation and Det PL in (**V**→**B**→**H**, Stage 2).

SHIFT. We conduct experiments on the SHIFT dataset, and report the results in Tab. 3. The results confirm the findings and trends observed ob BDD100K. COOLer consistently outperforms all other baselines across all tracking metrics on all classes, further showing the superiority and generality of our approach.

5.4 Ablation Study

We here ablate on method components and analyze qualitative results. In the supplement, we provide an additional analysis of the performance on old classes (model's rigidity) and new classes (model's plasticity) under incremental stages.

Ablation on Method Components. We show the effectiveness of each proposed component of COOLer in Tab. 4. We compare our detection pseudo-labels refined by the tracker (Track) vs. unrefined detection pseudo-labels from the object detector only (Det). Moreover, we analyze the effect of additional class-incremental contrastive losses, comparing ours (Ours) vs. OWOD's [12] (OWOD). Our components consistently improve over the baselines, and the improvement is more significant as more incremental stages are performed, suggesting that more stages pose a greater challenge in CIL for MOT. Notably, using the tracking pseudo-labels improves over all metrics, with 22.2% mMOTA, 6.9% mHOTA, 10.5% mIDF1, and 1.8% mAP at stage 4. Enabling the class-incremental contrastive loss further boosts 2.9% mMOTA, 0.5% mHOTA, and 1.0% mIDF1, and 0.4% mAP, highlighting the superiority of our contrastive loss. The results confirm that COOLer can (i) utilize the tracker's temporal refinement to produce higher-quality labels for detection, (ii) better preserve the association performance thanks to the association pseudo-labels, and (iii) that our contrastive loss design outperforms OWOD's.

Qualitative Comparison. Fig. 4 shows that the Det PL baseline would suffer from ID switches of the car (red) in the middle, due to the misclassification of it as a bus (purple) in the second frame. It can also not associate the pedestrians beside the pole across two frames (ID 322 switches to ID 325). Nevertheless, COOLer can both correctly classify the car and associate the pedestrians. This demonstrates that COOLer can better retain the knowledge of associating objects of the old classes while reducing misclassifications.

6 Conclusion

Our work is the first to address continual learning for MOT, a practical problem as MOT datasets are expensive to collect. We introduce COOLer, the first comprehensive approach to class-incremental learning for multiple object tracking. COOLer adopts a continual pseudo-label generation strategy for tracking that leverages the previous tracker to generate association pseudo-labels and temporally-refine detection pseudo-labels, while introducing class-incremental instance representation learning to further improve the tracking performance. Experimental results demonstrate that COOLer overcomes the drawbacks of detection-oriented methods, improving both detection and association performance. Although highly effective in the proposed setting, COOLer assumes that instances from the previous classes are present in the new training data. We believe experience replay to be a possible solution to this limitation, and we leave its exploration to future work. We hope our work can stimulate future research in this challenging yet practical direction.

7 Acknowledgments

This work was funded by the Max Planck ETH Center for Learning Systems.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
3. Cha, H., Lee, J., Shin, J.: Co2l: Contrastive continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9516–9525 (2021)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
5. Chen, Z., Liu, B.: Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning **12**(3), 1–207 (2018)

6. Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D.: Tao: A large-scale benchmark for tracking any object. In: European conference on computer vision. pp. 436–454. Springer (2020)
7. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
8. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia (2023)
9. Fan, H., Zheng, L., Yan, C., Yang, Y.: Unsupervised person re-identification: Clustering and fine-tuning. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **14**(4), 1–18 (2018)
10. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5830–5840 (2021)
13. Karthik, S., Prabhu, A., Gandhi, V.: Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609 (2020)
14. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
15. Li, M., Zhu, X., Gong, S.: Unsupervised tracklet person re-identification. IEEE transactions on pattern analysis and machine intelligence **42**(7), 1770–1782 (2019)
16. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
17. Liu, Y., Schiele, B., Vedaldi, A., Rupprecht, C.: Continual detection transformer for incremental object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23799–23808 (2023)
18. Mai, Z., Li, R., Kim, H., Sanner, S.: Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3589–3599 (2021)
19. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
20. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
21. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 164–173 (2021)
22. Peng, C., Zhao, K., Lovell, B.C.: Faster ilod: Incremental learning for object detectors based on faster rcnn. Pattern Recognition Letters **140**, 109–115 (2020)
23. Ramanan, D., Forsyth, D.A.: Finding and tracking people from the bottom up. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. vol. 2, pp. II–II. IEEE (2003)

24. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
26. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
27. Segu, M., Schiele, B., Yu, F.: Darth: Holistic test-time adaptation for multiple object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
28. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of the IEEE international conference on computer vision. pp. 3400–3409 (2017)
29. Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., Yu, F.: SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21371–21382 (June 2022)
30. Wang, Y.H.: Smiletrack: Similarity learning for multiple object tracking. arXiv preprint arXiv:2211.08824 (2022)
31. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3645–3649. IEEE (2017). https://doi.org/10.1109/ICIP.2017.8296962
32. Wu, G., Zhu, X., Gong, S.: Tracklet self-supervised learning for unsupervised person re-identification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12362–12369 (2020)
33. Xie, J., Yan, S., He, X.: General incremental learning with domain-aware categorical representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14351–14360 (2022)
34. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2636–2645 (2020)
35. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: European Conference on Computer Vision. pp. 1–21. Springer (2022)
36. Zheng, K., Chen, C.: Contrast r-cnn for continual learning in object detection. arXiv preprint arXiv:2108.04224 (2021)
37. Zhou, W., Chang, S., Sosa, N., Hamann, H., Cox, D.: Lifelong object detection. arXiv preprint arXiv:2009.01129 (2020)
38. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)