Continual Detection Transformer for Incremental Object Detection

Yaoyao Liu[1] Bernt Schiele[1] Andrea Vedaldi[2] Christian Rupprecht[2]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]Visual Geometry Group, Department of Engineering Science, University of Oxford
{yaoyao.liu, schiele}@mpi-inf.mpg.de {vedaldi, chrisr}@robots.ox.ac.uk

Abstract

Incremental object detection (IOD) aims to train an object detector in phases, each with annotations for new object categories. As other incremental settings, IOD is subject to catastrophic forgetting, which is often addressed by techniques such as knowledge distillation (KD) and exemplar replay (ER). However, KD and ER do not work well if applied directly to state-of-the-art transformer-based object detectors such as Deformable DETR [60] and UP-DETR [10]. In this paper, we solve these issues by proposing a ContinuaL DEtection TRansformer (CL-DETR), a new method for transformer-based IOD which enables effective usage of KD and ER in this context. First, we introduce a Detector Knowledge Distillation (DKD) loss, focusing on the most informative and reliable predictions from old versions of the model, ignoring redundant background predictions, and ensuring compatibility with the available ground-truth labels. We also improve ER by proposing a calibration strategy to preserve the label distribution of the training set, therefore better matching training and testing statistics. We conduct extensive experiments on COCO 2017 and demonstrate that CL-DETR achieves state-of-the-art results in the IOD setting.[1]

1. Introduction

Humans inherently learn in an incremental manner, acquiring new concepts over time without forgetting previous ones. In contrast, machine learning suffers from *catastrophic forgetting* [22, 36, 37], where learning from non-i.i.d. data can override knowledge acquired previously. Unsurprisingly, forgetting also affects object detection [2, 13, 21, 38, 45, 51, 55]. In this context, the problem was formalized by Shmelkov *et al*. [45], who defined an incremental object detection (IOD) protocol, where the training samples for different object categories are observed in phases, restricting the ability of the trainer to access past data.

[1]Code: https://lyy.mpi-inf.mpg.de/CL-DETR/

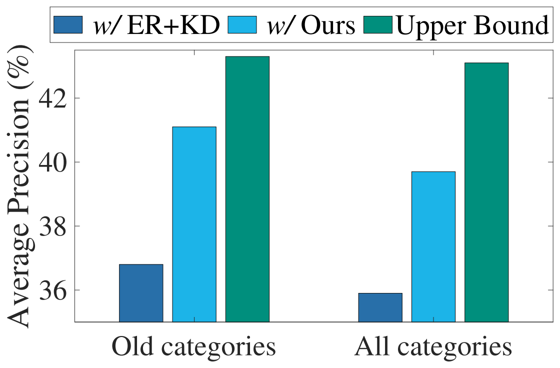


Figure 1. The final Average Precision (AP, %) of two-phase incremental object detection on COCO 2017. We observe 70 and 10 categories in the first and second phases, respectively. The baseline is Deformable DETR [60]. "Upper bound" shows the results of joint training with all previous data accessible in each phase.

Popular methods to address forgetting in tasks other than detection include Knowledge Distillation (KD) and Exemplar Replay (ER). KD [12, 17, 18, 27, 58] uses regularization in an attempt to preserve previous knowledge when training the model on new data. The key idea is to encourage the new model's logits or feature maps to be close to those of the old model. ER methods [6, 30, 33, 34, 42, 53] work instead by memorising some of the past training data (the *exemplars*), replaying them in the following phases to "remember" the old object categories.

Recent state-of-the-art results in object detection have been achieved by a family of transformer-based architectures that include DETR [5], Deformable DETR [60] and UP-DETR [10]. In this paper, we show that KD and ER do not work well if applied directly to these models. For instance, in Fig. 1 we show that applying KD and ER to Deformable DETR leads to much worse results compared to training with all data accessible in each phase (*i.e*., the standard non-incremental setting).

We identify two main issues that cause this drop in performance. First, transformer-based detectors work by testing a large number of object hypotheses in parallel. Because the number of hypotheses is much larger than the typical number of objects in an image, most of them are negative, resulting in an unbalanced KD loss. Furthermore, because both old and new object categories can co-exist in any given training image, the KD loss and regular training objective can provide contradictory evidence. Second, ER methods for image classification try to sample the same number of exemplars for each category. In IOD, this is not a good strategy because the true object category distribution is typically highly skewed. Balanced sampling causes a mismatch between the training and testing data statistics.

In this paper, we solve these issues by proposing *ContinuaL DEtection TRansformer* (CL-DETR), a new method for transformer-based IOD which enables effective usage of KD and ER in this context. CL-DETR introduces the concept of *Detector Knowledge Distillation* (DKD), selecting the most confident object predictions from the old model, merging them with the ground-truth labels for the new categories while resolving conflicts, and applying standard joint bipartite matching between the merged labels and the current model's predictions for training. This approach subsumes the KD loss, applying it only for foreground predictions correctly matched to the appropriate model's hypotheses. CL-DETR also improves ER by introducing a new calibration strategy to preserve the distribution of object categories observed in the training data. This is obtained by carefully engineering the set of exemplars remembered to match the desired distribution. Furthermore, each phase consists of a main training step followed by a smaller one focusing on better calibrating the model.

We also propose a more realistic variant of the IOD benchmark protocol. In previous works [13, 45], in each phase, the incremental detector is allowed to observe all images that contain a certain type of object. Because images often contain a mix of object classes, both old and new, this means that the same images can be observed in different training phases. This is incompatible with the standard definition of incremental learning [17, 34, 42] where, with the exception of the examples deliberately stored in the exemplar memory, the images observed in different phases do not repeat. We redefine the IOD protocol to avoid this issue.

We demonstrate CL-DETR by applying it to different transformer-based detectors including Deformable DETR [60] and UP-DETR [10]. As shown in Fig. 1, our results on COCO 2017 show that CL-DETR leads to significant improvements compared to the baseline, boosting AP by 4.2 percentage points compared to a direct application of KD and ER to the underlying detector model. We further study and justify our modelling choices via ablations.

To summarise, we make **four contributions**: (1) The DKD loss that improves KD for knowledge distillation by resolving conflicts between distilled knowledge and new evidence and by ignoring redundant background detections; (2) A calibration strategy for ER to match the stored exemplars to the training set distribution; (3) A revised IOD benchmark protocol that avoids observing the same images in different training phases; (4) Extensive experiments on COCO 2017, including state-of-the-art results, an in-depth ablation study, and further visualizations.

2. Related Work

Incremental learning. Incremental learning (also known as continual learning [2, 11, 35] and lifelong learning [1, 7, 9]) aims at learning models in phases that focus on different subsets of the label space. Recent incremental learning methods can be divided into two categories: (i) Knowledge Distillation (KD) tries to preserve the knowledge capture in a previous version of the model by matching logits [27,42], feature maps [12], or other information [19, 40, 46, 49, 52] in the new model. (ii) Exemplar Replay (ER) methods build a reservoir of samples or exemplars from old training rounds [3, 34, 41, 42, 44] and replay them in successive training phases as a way of recalling past knowledge. KD and ER are the starting point of our method.

Incremental object detection (IOD). IOD applies incremental learning to object detection specifically. This is more challenging than incremental image classification, as images can contain multiple objects, both of old and new types, with only the new types being annotated in any given training phase. Both KD and ER have been applied to detection before. [45] applies KD to the output of Faster R-CNN [14]. Inspired by this, recent IOD methods extended the KD framework to other detectors (*e.g.*, Faster-RCNN [43] and GFL [26]) by adding KD terms on the intermediate feature maps [13,56,59] and region proposal networks [8, 15, 38]. [20] proposes instead to store a set of exemplars and fine-tune the model on the exemplars after each incremental step. [31] proposes an adaptive sampling strategy to achieve more efficient exemplar selection for IOD.

However, existing IOD methods are designed based on conventional detectors such as Faster-RCNN [43] and GFL [26]. In this work, we show that a direct application of KD and ER to current state-of-the-art transformer-based detectors such as Deformable DETR [60] and UP-DETR [10] does not work well and we propose fixes to this issue.

Transformer-based object detection. DEtection TRansformer (DETR) [5] proposes an elegant architecture for object detection based on a visual transformer [50]. Compared to pre-transformer approaches, DETR eliminates the need for non-maximum suppression in post-processing because self-attention can learn to remove duplicated detection by itself. This is achieved by using the Hungarian loss, matching each object hypothesis to exactly one target or background using bipartite matching [48]. Deformable DETR [60] improves the performance of DETR, particularly for small objects, via sparse attention on multi-level feature maps. UP-DETR [10] leverages unsupervised learning to pre-train the parameters of the encoder and decoder in DETR to further boost the performance.

Our method does not fundamentally change these detectors and is in fact applicable to all similar ones. Instead, it proposes broadly-applicable changes that make transformer-based detectors work well in combination with KD and ER for the IOD problem.

3. Methodology

After defining the incremental detection problem (Sec. 3.1) and providing the necessary background (Sec. 3.2), we introduce ContinuaL DEtection TRansformer (CL-DETR), a new method for incremental object detection that extends DETR-like detectors with knowledge distillation (KD; Sec. 3.3) and exemplar replay (ER; Sec. 3.4).

3.1. Incremental object detection

In *incremental object detection* (IOD) the goal is to train a detector in phases, where in each phase the model is only given annotations for a subset of the object categories. Formally, let $\mathcal{D} = \{(x, y)\}$ be a dataset of images x with corresponding object annotations y, such as COCO 2017 [28], and let $\mathcal{C} = \{1, \dots, C\}$ be the set of object categories. We adapt such a dataset for benchmarking IOD as follows. First, we partition $\mathcal{D}$ and $\mathcal{C}$ into M subsets $\mathcal{D} = \mathcal{D}_1 \cup \dots \cup \mathcal{D}_M$ and $\mathcal{C} = \mathcal{C}_1 \cup \dots \cup \mathcal{C}_M$, one for each training phase. For each phase i, we modify the samples $(x, y) \in \mathcal{D}_i$ so that y only contains annotations for objects of class $\mathcal{C}_i$ and drop the others.[2]

In phase i of training, the model is only allowed to observe images $\mathcal{D}_i$ with annotations for objects of types $\mathcal{C}_i \subset \mathcal{C}$. Notably, images can and do contain objects of any possible type $\mathcal{C}$, but only types $\mathcal{C}_i$ are annotated in this phase. After phase i is complete, training switches to the next phase $i+1$, so the model observes different images $\mathcal{D}_{i+1}$ and annotations for objects of different types $\mathcal{C}_{i+1}$.

For exemplar replay, we relax this training protocol and allow the model to memorise a small number of *exemplars* $\mathcal{E}_i \subset \mathcal{D}_i$ from the previous phases. In this case, the model is trained on the union $\mathcal{D}_i \cup \mathcal{E}_{1:i-1}$ where $\mathcal{E}_{1:i-1} = \mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}$ forms the exemplar memory.

Note that this is a stricter and improved protocol compared to prior works in IOD [13, 45]. In these works, the model is still presented a subset of annotations restricted to classes $\mathcal{C}_i$ in each phase; however, $\mathcal{D}_i \subset \mathcal{D}$ is defined as the subset of all images that contain objects of type $\mathcal{C}_i$. Because images contain a mix of object categories that can span different subsets $\mathcal{C}_i$, this means that different subsets $\mathcal{D}_i$ can overlap, so that the same images can be observed multiple times in different phases. This violates the standard definition of incremental learning [17, 34, 42] which assumes that different samples are observed in different phases. Our setting retains this property.

[2]In this way, some images end up containing no annotated objects.

3.2. Transformer-based detectors

State-of-the-art methods like DETR [5, 10, 29, 48, 57, 60] build on powerful visual transformers to solve the object detection problem. In order to motivate and explain our method, we first review briefly how they work.

With reference to Fig. 2, the model Φ takes as input an image $x \in \mathbb{R}^{3\times H\times W}$ and outputs the object predictions $\hat{y} = \Phi(x)$ using a number of attention and self-attention layers. The output $\hat{y} = (\hat{y}_j)_{j\in\mathcal{N}}$ is a sequence $\mathcal{N} = \{1, \dots, N\}$ of object predictions $\hat{y}_j = (\hat{p}_j, \hat{b}_j)$, consisting of a class probability vector $\hat{p}_j : \mathcal{C} \cup \{\phi\} \to [0, 1]$ and a vector $b_j \in [0, 1]^4$ specifying the centre and size of the object bounding box relative to the image size. Note that the support of $\hat{p}_j$ includes element ϕ that denotes the background class, or 'no object' (hence, $\hat{p}_j$ has $C + 1$ dimensions).

The object predictions correspond to a fixed set of *object queries* internal to the model. Each query is thus mapped to an object instance or background. The order of the queries is conceptually immaterial, but queries are fixed and non-interchangeable after training. For instance, $\hat{y}_1$ is always the prediction that corresponds to the first query in the model. This is relevant for the application of KD.

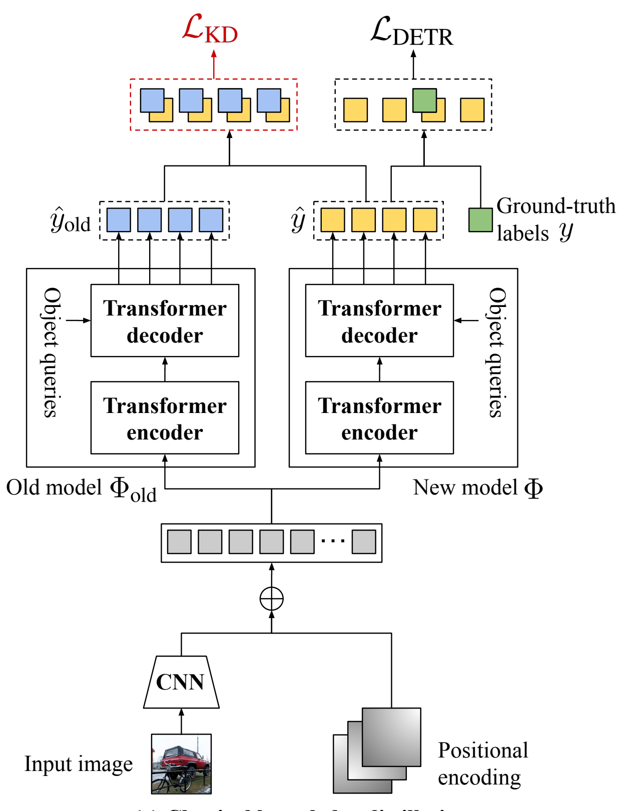
For supervised training, the model is given ground truth object annotations $y = ((p_j, b_j))_{j\in\mathcal{N}}$ where p_j is the indicator vector of the category of the object and $b_j \in [0, 1]^4$ is its bounding box. Images usually contain fewer objects than the number N of hypotheses, so y is padded with background detections for which $p_i(\phi) = 1$ and b_i is arbitrary. The model is trained end-to-end to optimise the loss,

$$\mathcal{L}_{\text{DETR}}(\hat{y}, y) = \sum_{i\in\mathcal{N}} \langle -\log \hat{p}_{\hat{\sigma}_i}, p_i\rangle + \mathbf{1}_{c(p_i)\neq\phi}\mathcal{L}_{\text{box}}(\hat{b}_{\hat{\sigma}_i}, b_i), \tag{1}$$

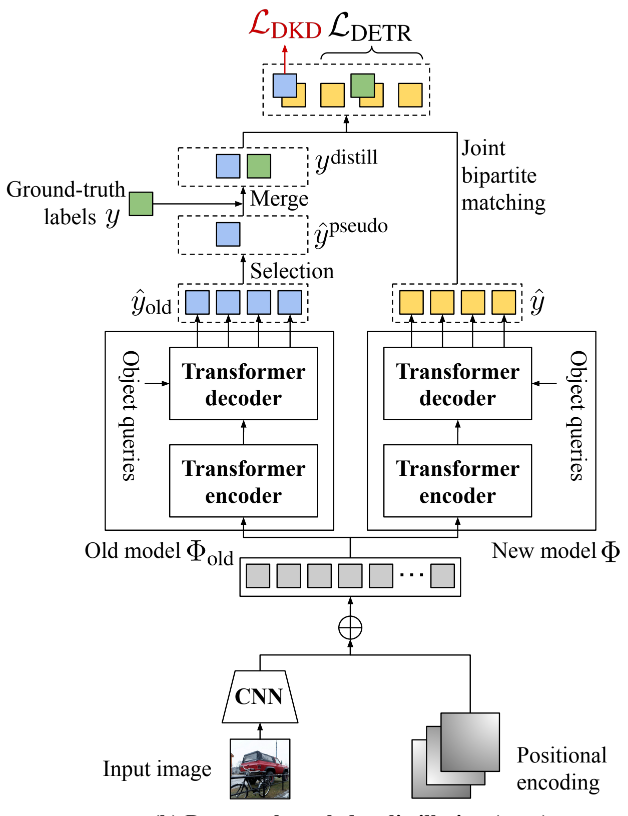
where $c(p_i) = \operatorname{argmax}_{c\in\mathcal{C}\cup\{\phi\}} p_i(c)$ is the class encoded by p_i, $\mathcal{L}_{\text{box}}(\hat{b}_{\hat{\sigma}_i}, b_i) = \gamma_1\mathcal{L}_{\text{IoU}}(\hat{b}_{\hat{\sigma}_i}, b_i) + \gamma_2\|\hat{b}_{\hat{\sigma}_i} - b_i\|_1$ is the bounding box prediction loss and $\hat{\sigma}$ is the best association of ground truth labels to object hypotheses, obtained by solving the matching problem,

$$\hat{\sigma} = \operatorname*{argmax}_{\sigma\in\mathcal{S}_N} \sum_{i\in\mathcal{N}} \mathbf{1}_{c(p_i)\neq\phi} \left\{ -\langle \hat{p}_{\sigma_i}, p_i\rangle + \mathcal{L}_{\text{box}}(\hat{b}_{\sigma_i}, b_i), \right\} \tag{2}$$

using the Hungarian algorithm [23, 47] (see [5] for details).



(a) Classical knowledge distillation



(b) Detector knowledge distillation (ours)

Figure 2. **(a) Classical knowledge distillation.** There are two issues when directly applying KD [16, 27] to the transformer-based detectors [5, 10, 60]. (i) Transformer-based detectors work by testing a large number of object hypotheses in parallel. Because the number of hypotheses is much larger than the typical number of objects in an image, most of them are negative, resulting in an unbalanced KD loss. (ii) Because both old and new object categories can co-exist in any given training image, the KD loss and regular training objective can provide contradictory evidence. **(b) Detector knowledge distillation (ours).** We select the most confident foreground predictions from the old model and use them as pseudo labels. We purposefully ignore background predictions because they are imbalanced and they can contradict the labels of the new classes available in the current phase. Then, we *merge* the pseudo labels for the old categories with the ground-truth labels for the new categories and use bipartite matching to train the model on the joint labels. This inherits the good properties of the original formulation such as ensuring one-to-one matching between labels and hypotheses and avoiding duplicate detections.

3.3. Detector knowledge distillation

In a multi-phase learning scenario, at the beginning of a new phase, the model is initialized as $\Phi \leftarrow \Phi^{\text{old}}$ where Φ^{old} is the model trained in the phase before. As the new data for the current phase is received, training the model Φ as normal by minimising eq. (1) leads to forgetting.

KD [16, 27] reduces forgetting by maintaining a copy of the old model and making sure that the outputs of the new and old models stay close. Applied to our transformer-based detectors, given a new training image-label pair (x, y), one computes the old model's output $\hat{y}^{\text{old}} = \Phi^{\text{old}}(x)$ and, minimizes the sum of the $\mathcal{L}_{\text{DETR}}(\hat{y}, y)$ loss with the *knowledge distillation loss*

$$\mathcal{L}_{\text{KD}}(\hat{y}, \hat{y}^{\text{old}}) = \sum_{j\in\mathcal{N}} \left[\sum_{c\in\mathcal{C}} -\hat{p}_j(c) \log \hat{p}_j^{\text{old}}(c) \right] + \mathcal{L}_{\text{box}}(\hat{b}_j, \hat{b}_j^{\text{old}}).$$

This loss compares the output tokens of the new and old models, which makes sense since they depend on the same object queries, at least initially, and are thus in correspondence. However, we find that this loss is dominated by background information because most of the tokens predict background. Furthermore, transformer-based detectors aim to find one-to-one matchings between predictions and ground-truth labels without duplicates, which is not accounted for by the classical KD loss.

The key issue is that summing losses $\mathcal{L}_{\text{DETR}} + \mathcal{L}_{\text{KD}}$ as in standard KD fails to properly account for the *structure* of the labels, which is crucial for detection problems, particularly in an incremental learning setting. Specifically, the old model knows about all categories seen so far during training *except* the new categories that are annotated in the current phase. However, the new training images contain multiple objects, including the old types, which are thus *not* annotated in the current phase. This means that $\mathcal{L}_{\text{DETR}}$ and $\mathcal{L}_{\text{KD}}$ provide potentially contradictory supervision.

We thus suggest that, in a detection context, new and old knowledge should be fused in a *structured manner*. As illustrated in Fig. 2, we do so by selecting the most confident foreground predictions from the old model and using them as pseudo labels. We purposefully ignore background predictions because they are imbalanced and they can contradict the labels of the new classes available in the current phase. Then, we *merge* the pseudo labels for the old categories with the ground-truth labels for the new categories and use bipartite matching to train the model on the joint labels. This inherits the good properties of the original formulation such as ensuring one-to-one matching between labels and hypotheses and avoiding duplicate detections.

Formally, given the predictions $\hat{y}^{\text{old}}$ from the old model, we first identify the subset $\mathcal{F} \subset \mathcal{N}$ of the ones that are predicted as foreground: $\mathcal{F} = \{j \in \mathcal{N} : \forall c \in \mathcal{C} : \hat{p}_j^{\text{old}}(c) > \hat{p}_j^{\text{old}}(\phi)\}$. Of these, we pick the subset $\mathcal{P} \subset \mathcal{F}$, $|\mathcal{P}| = K$ formed by the K most confident predictions, *i.e.*,

$$\forall i \in \mathcal{P},\ j \in \mathcal{F} - \mathcal{P}:\ \max_{c \in \mathcal{C}} \hat{p}_i^{\text{old}}(c) > \max_{c \in \mathcal{C}} \hat{p}_j^{\text{old}}(c).$$

Finally, we further restrict the predictions to the subset $\mathcal{Q} \subset \mathcal{P}$ that does not overlap too much with the ground-truth labels for the new categories:

$$\mathcal{Q} = \{j \in \mathcal{P} :\ \forall i \in \mathcal{N} :\ c(p_i) \neq \phi \Rightarrow \text{IoU}(\hat{b}_j^{\text{old}}, b_i) \leq \lambda\}.$$

In the experiments, we set $\lambda = 0.7$. With remain with a filtered set of pseudo-labels:

$$\hat{y}^{\text{pseudo}} = (\hat{y}_j^{\text{old}})_{j \in \mathcal{Q}}. \tag{3}$$

Next, we distill knowledge from the current labels y and the pseudo-labels obtained from the old model into a single, coherent set of labels

$$y^{\text{distill}} = (y_i)_{i:c(p_i) \neq \phi} \oplus \hat{y}^{\text{pseudo}} \oplus y^{\text{bg}}, \tag{4}$$

where we concatenate the object labels for the new categories, the pseudo-labels, and enough background labels y^{bg} to pad y^{distill} to contain N elements.

In this manner, the distillation occurs at the level of the labels. The model is still trained by using eq. (1) as before, resulting in the *detector knowledge distillation* (DKD) loss:

$$\mathcal{L}_{\text{DKD}}(\hat{y}, y^{\text{distill}}) = \mathcal{L}_{\text{DETR}}(\hat{y}, y^{\text{distill}}). \tag{5}$$

Besides the usage of the distilled labels, the main difference between eqs. (1) and (5) is that, while the class distribution p_i for the new label is deterministic, it is *not* for the pseudo-labels. Plugged in eq. (1), this results in the standard distillation effect for categorical distributions trained using the cross entropy loss.

Algorithm 1: CL-DETR (the i-th phase)

```
Input: new category data $\mathcal{D}_i$; old category
       exemplars $\mathcal{E}_{1:i-1}$; old model $\Phi^{\text{old}}$.
Output: new model $\Phi$; exemplars $\mathcal{E}_{1:i}$.
1  Get $\mathcal{D}_i$ and load $\mathcal{E}_{1:i-1}$ from memory;
2  Let $\Phi \leftarrow \Phi^{\text{old}}$;
3  for epochs do
4      for mini-batches $(x, y) \in \mathcal{D}_i \cup \mathcal{E}_{1:i-1}$ do
5          Let $\hat{y}^{\text{old}} \leftarrow \Phi^{\text{old}}(x)$;
6          Get $\hat{y}^{\text{pseudo}}$ from $\hat{y}^{\text{old}}$ and $y$ using eq. (3);
7          Get $y^{\text{distill}}$ from $\hat{y}^{\text{pseudo}}$ and $y$ using eq. (4);
8          Let $\hat{y} \leftarrow \Phi(x)$;
9          Get $\hat{\sigma}$ by matching $y^{\text{distill}}$ to $\hat{y}$ using eq. (2);
10         Compute $\mathcal{L}_{\text{DKD}}(\hat{y}, y^{\text{distill}})$ using eq. (5);
11         Update $\Phi$ via a gradient step.
12 Build the exemplar set $\mathcal{E}_{1:i}$ using Algorithm 2;
13 for epochs do
14     for mini-batches $(x, y) \in \mathcal{E}_{1:i}$ do
15         Let $\hat{y} \leftarrow \Phi(x)$;
16         Compute $\mathcal{L}_{\text{DETR}}(\hat{y}, y)$ using eq. (1);
17         Update $\Phi$ via a gradient step;
18 Save $\mathcal{E}_{1:i}$ to the memory.
```

Algorithm 2: Exemplar selection (the i-th phase)

```
Input: new category data $\mathcal{D}_i$; old category
       exemplars $\mathcal{E}_{1:i-1}$; target number of
       exemplars $R_i$.
Output: exemplars $\mathcal{E}_{1:i}$.
1  Let $\mathcal{E}_i \leftarrow \{\}$;
2  repeat
3      Select $e \in \mathcal{D}_i$ according to eq. (6);
4      Let $\mathcal{E}_i \leftarrow \mathcal{E}_i \cup \{x\}$;
5  until $R_i$ times;
6  Let $\mathcal{E}_{1:i} \leftarrow \mathcal{E}_i \cup \mathcal{E}_{1:i-1}$.
```

3.4. Distribution-preserving calibration

ER methods, which store a small number of exemplars and replay them in future phases, are shown to be effective in preserving the old category knowledge in IOD [20, 31], but can suffer from the severe imbalance between old and new category annotations. Incremental learning methods for classification [17,32,54] usually use re-balancing strategies to address the imbalance problem. They create a category-balanced subset of the data and finetune some model components (*e.g.*, the classifier) on it. However, such strategies do not apply directly to the IOD setting. First, the class distribution in detection is far from balanced, and a better strategy is to match the natural data distribution instead of the uniform one. Second, because there are multi-

Setting	Method	Detection baseline	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
70+10	ERD [13]	UP-DETR	36.2$_{\pm 0.3}$	54.8$_{\pm 0.4}$	39.3$_{\pm 0.4}$	20.8$_{\pm 0.3}$	39.3$_{\pm 0.5}$	47.9$_{\pm 0.3}$
	CL-DETR (ours)	UP-DETR	37.6$_{\pm 0.2}$	56.5$_{\pm 0.4}$	39.4$_{\pm 0.3}$	20.5$_{\pm 0.3}$	39.1$_{\pm 0.4}$	49.9$_{\pm 0.3}$
	LwF [27]	Deformable DETR	24.5$_{\pm 0.3}$	36.6$_{\pm 0.2}$	26.7$_{\pm 0.4}$	12.4$_{\pm 0.2}$	28.2$_{\pm 0.4}$	35.2$_{\pm 0.4}$
	iCaRL [42]	Deformable DETR	35.9$_{\pm 0.4}$	52.5$_{\pm 0.3}$	39.2$_{\pm 0.3}$	19.1 $_{\pm 0.3}$	39.4$_{\pm 0.5}$	48.6$_{\pm 0.3}$
	ERD [13]	Deformable DETR	36.9$_{\pm 0.4}$	55.7$_{\pm 0.4}$	40.1$_{\pm 0.4}$	21.4$_{\pm 0.3}$	39.6$_{\pm 0.3}$	48.7$_{\pm 0.3}$
	CL-DETR (ours)	Deformable DETR	**40.1**$_{\pm 0.3}$	**57.8**$_{\pm 0.4}$	**43.7**$_{\pm 0.3}$	**23.2**$_{\pm 0.3}$	**43.2**$_{\pm 0.2}$	**52.1**$_{\pm 0.3}$
40+40	ERD [13]	UP-DETR	35.4$_{\pm 0.4}$	55.1$_{\pm 0.3}$	38.3$_{\pm 0.3}$	17.9$_{\pm 0.4}$	39.0$_{\pm 0.3}$	49.8$_{\pm 0.3}$
	CL-DETR (ours)	UP-DETR	37.0$_{\pm 0.2}$	**56.2**$_{\pm 0.2}$	39.1$_{\pm 0.4}$	**20.9**$_{\pm 0.2}$	38.9$_{\pm 0.3}$	49.2$_{\pm 0.3}$
	LwF [27]	Deformable DETR	23.9$_{\pm 0.2}$	41.5$_{\pm 0.3}$	25.0$_{\pm 0.3}$	12.0$_{\pm 0.4}$	26.4$_{\pm 0.3}$	33.0$_{\pm 0.5}$
	iCaRL [42]	Deformable DETR	33.4$_{\pm 0.4}$	52.0$_{\pm 0.3}$	36.0$_{\pm 0.2}$	18.0$_{\pm 0.3}$	36.4$_{\pm 0.3}$	45.5$_{\pm 0.4}$
	ERD [13]	Deformable DETR	36.0$_{\pm 0.2}$	55.2$_{\pm 0.2}$	38.7$_{\pm 0.3}$	19.5$_{\pm 0.2}$	38.7$_{\pm 0.3}$	49.0$_{\pm 0.4}$
	CL-DETR (ours)	Deformable DETR	**37.5**$_{\pm 0.3}$	55.1$_{\pm 0.4}$	**40.3**$_{\pm 0.2}$	**20.9**$_{\pm 0.2}$	**40.8**$_{\pm 0.4}$	**50.7**$_{\pm 0.2}$

Table 1. IOD results (%) on COCO 2017. In the $A+B$ setup, in the first phase, we observe a fraction $\frac{A}{A+B}$ of the training samples with A categories annotated. Then, in the second phase, we observe the remaining $\frac{B}{A+B}$ of the training samples, where B new categories are annotated. We test settings $A+B = 40+40$ and $70+10$. Exemplar replay is applied for all methods except for LwF [27]. We run experiments for three different categories and data orders and report the average AP with 95% confidence interval.

ple objects in each image, it is non-trivial to create a subset of exemplar images with a set number of objects for each category. We address these issues next.

Selecting exemplars to match the training distribution. Called during phase i, Algorithm 2 produces a new exemplar subset $\mathcal{E}_i$ whose distribution matches as well as possible the distribution of categories in the subset $\mathcal{D}_i$ of the data. This is achieved by adding to $\mathcal{E}_i$ a set number R_i of one exemplar $e^* \in \mathcal{D}_i$, one at a time, chosen by minimizing the Kullback-Leibler divergence [24] between the category marginals of $\mathcal{E}_i$ and $\mathcal{D}_i$:

$$e^* \leftarrow \sum_{c \in \mathcal{C}_i} p_{\mathcal{D}_i}(c) \log p_{\mathcal{E}_i \cup \{e\}}(c), \tag{6}$$

where $p_{\mathcal{D}}(c)$ denotes the probability of category c in dataset $\mathcal{D}$. Then, the overall exemplar set $\mathcal{E}_{1:i} = \mathcal{E}_i \cup \mathcal{E}_{1:i-1}$ is obtained as the union of the new subset just found and the previous exemplar et $\mathcal{E}_{1:i-1}$. Because classes in different subsets $\mathcal{D}_i$ are disjoint, this also means that, by the end of the training, the distribution of classes in $\mathcal{E}_{1:M}$ approximates the one of the overall training set $\mathcal{D}$.

Learning using balanced data. In order to use the available data as well as possible while balancing the detector Φ, in each phase we update it in two steps. In the first step, the model is trained using the DKD loss on all the available data $\mathcal{D}_i \cup \mathcal{E}_{1:i-1}$ given by the union of the current data subset $\mathcal{D}_i$ and the exemplar memory $\mathcal{E}_{1:i-1}$ carried over the previous training phases. In the second step, the model is fine-tuned using the new exemplar set $\mathcal{E}_{1:i}$, ignoring $\mathcal{D}_i$ and using only the DETR loss, using fewer data but achieving better calibration. The overall algorithm is given in Algorithm 1.

4. Experiments

We evaluate CL-DETR on COCO 2017 using two transformer-based detectors, Deformable DETR and UP-DETR [10, 60] as the baselines and achieve consistent improvements compared to the baselines and a direct application of KD and ER. Below we describe the dataset and implementation details (Sec. 4.1) followed by results and analyses (Sec. 4.2).

4.1. Dataset and implementation details

Dataset and evaluation metrics. We conduct IOD experiments on COCO 2017 [28], which is widely used in related works [10, 13, 39, 60]. Following [13], the standard COCO metrics are used for evaluation, *i.e.*, AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. In the ablation study, we introduce a new metric, *forgetting percentage points* (FPP), measuring the difference between the AP of the first and last phase models on the categories observed in the first phase.

Experiment setup. We conduct IOD experiments in the following setting. ***Two-phase setting:*** In the $A+B$ setup, in the first phase, we observe a fraction $\frac{A}{A+B}$ of the training samples with A categories annotated. Then, in the second phase, we observe the remaining $\frac{B}{A+B}$ of the training samples, where B new categories are annotated. We test settings $A+B = 40+40$ and $70+10$. ***Multiple-phase setting:*** In the $40+X\times Y$ setup, in the first phase, we observe half of the training samples with 40 categories annotated. In each following phase, we observe $\frac{1}{2Y}$ of the training samples we have never seen before with annotations for X new categories. We run experiments for $40+20\times 2$ and $40+10\times 4$. We repeat each experiment three times, randomizing the order of categories and data in the different phases, and report the average APs. The total memory budget for the exem-

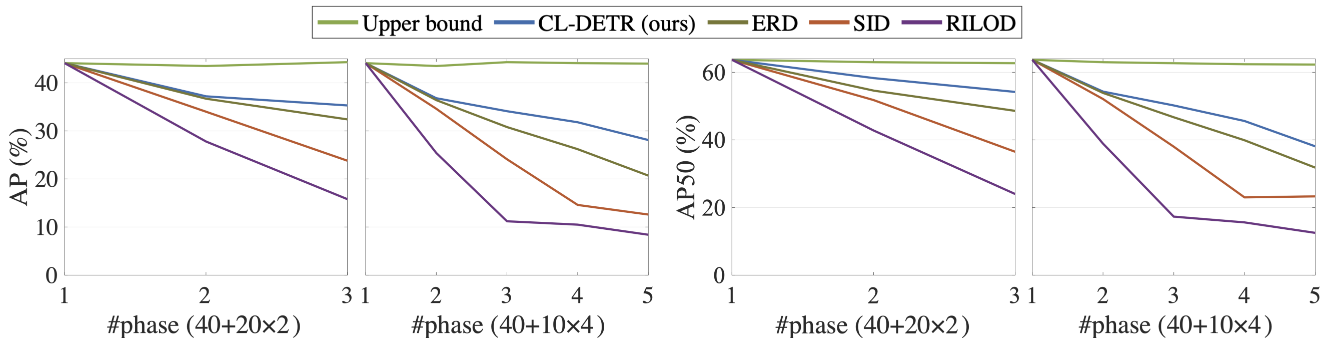


Figure 3. IOD results (AP/AP$_{50}$, %) on COCO 2017 in the $40+20\times2$ and $40+10\times4$ settings. Our method is based on Deformable DETR. Comparing methods: Upper Bound (the results of joint training with all previous data accessible in each phase), ERD [13], SID [39], and RILOD [25]. The results of the related works are from [13]. We use the same data split as [13] for a fair comparison.

Row	Knowledge distillation (KD)	Joint bipartite matching	Pseudo label selection	Exemplar replay (ER)	Distribution preserving calibration	All categories ↑				Old categories ↑				FPP ↓			
						AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
1						4.2	1.6	4.7	5.8	0.7	0.2	0.8	0.8	42.6	25.6	45.1	56.7
2	✓					24.5	12.4	28.2	35.2	24.0	12.3	27.7	34.4	19.3	13.5	18.2	23.1
3	✓	✓				30.3	19.5	33.0	39.0	33.4	21.8	36.4	43.2	9.9	4.0	9.5	14.3
4	✓	✓	✓			33.9	16.3	37.1	49.2	33.9	16.6	36.8	50.0	9.4	9.2	9.1	7.5
5	✓	✓	✓	✓		37.9	20.8	40.9	50.4	39.0	21.6	41.7	52.3	4.3	4.2	4.2	5.2
6	✓	✓	✓		✓	**40.1**	**23.2**	**43.2**	**52.1**	**41.8**	**24.5**	**44.7**	**54.6**	**1.5**	**1.3**	**1.2**	**2.9**

Table 2. Ablation results (%) for KD and ER, using Deformable DETR [60] on COCO 2017 in the $70+10$ setting. "All categories" (higher is better) denote the results of the last phase model on 80 categories. "Old categories" (higher is better) denote the results of the last phase model on 70 categories observed in the first phase. "Forgetting percentage points (FPP)" (lower is better) show the difference between the AP of the first-phase model and the last-phase model on 70 categories observed in the first phase. The baseline (row 1) is finetuning the model without IOD techniques. Our method (CL-DETR) is shown in row 6.

plars is set as 10% of the total dataset size.

Implementation details. We follow [10, 60] and use an ImageNet pre-trained ResNet-50 backbone. For the experiments on Deformable DETR [60], we use the standard configurations without their iterative bounding box refinement mechanism and the two-stage Deformable DETR. We train the model for 50 (Deformable DETR) and 150 epochs (UP-DETR), following the original implementations [10, 60]. In order to apply our distribution-preserving calibration (Sec. 3.4), we train the coarse Deformable DETR (UP-DETR) model for 40 (120) epochs and perform calibration for 10 (30) epochs to preserve the total number of epochs.

4.2. Results and analyses

Two-phase setting. Table 1 shows that, in the two-phase settings $70+10$ and $40+40$, applying CL-DETR to Deformable DETR [60] and UP-DETR [10] consistently performs better than the state-of-the-art [13] and other IOD methods [27, 42]. In particular, Deformable DETR [60] *w/* ours achieves the highest AP, *e.g.*, 40.1% and 37.5% in the $70+10$ and $40+40$ settings, respectively. The performance gap is larger with more categories in the 1-st phase. *E.g.*, the AP differences between our method and [13] are 3.2 and 1.5 percentage points when we observe 70 and 40 categories in the first phase, respectively, likely due to CL-DETR benefiting more from a well-pre-trained model.

Multiple-phase setting. Figure 3 evaluates CL-DETR in the multiple-phase setting with large gains compared to other IOD methods in both the $40+20\times2$ and $40+10\times4$ experimental variants. The relative advantage of CL-DETR increases with the number of phases. For instance, our method improves the AP of [13] by 2.9 percentage points in the $40+20\times2$ setting and by 7.4 percentage points in the $40+10\times4$ setting. This suggests that the advantage of CL-DETR shows more in challenging settings, where the forgetting problem is stronger due to the larger number of training phases.

Ablation study for DKD. In Tab. 2 (Rows 1–4) we ablate our DKD approach. By comparing row 2 to row 1, we observe that classical KD significantly improves the IOD performance compared to the baseline (*i.e.*, finetuning the model without IOD techniques), but still results in large

Row	Setting	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	K=5	39.7	57.4	43.1	22.7	42.6	**52.7**
2	K=10	**40.1**	**57.8**	**43.7**	23.2	**43.2**	52.1
3	K=20	39.9	57.8	43.2	**23.5**	42.9	51.7
4	p≥0.1	39.3	57.1	42.9	22.6	42.3	52.5
5	p≥0.3	39.6	57.5	43.0	23.2	42.4	52.2
6	p≥0.5	39.2	56.8	42.4	22.3	41.9	51.8

Table 3. Ablation result (%) for different pseudo label selection strategies on COCO 2017 using the 70 + 10 setting. Rows 1–3 show the results for using different K when selecting top-K most-confident non-background predictions. Rows 4–6 show the results for using different thresholds p of the prediction scores to select the non-background predictions.

overall forgetting: 19.3 FPP. Comparing row 3 to row 2, we can see that joint bipartite matching works well and boosts the AP of all categories by 5.8 percentage points compared to conventional KD. The reason is that joint bipartite matching helps ensure a one-to-one matching between objects and hypotheses and discourages duplicate detections. Comparing row 4 to row 3, our pseudo label selection further improves the AP and reduces forgetting, helping the model to ignore the redundant background information and reducing conflicts between old and new labels.

Ablation study for ER. In Tab. 2 (Rows 5–6), we ablate our ER method. Comparing row 6 to row 5, we can see that the calibration strategy of Sec. 3.4 boosts both the all-category and old-category performance, by 1.8 and 2.1 percentage points respectively, compared to using conventional ER [31, 42]. It also helps to overcome the catastrophic forgetting problem in IOD, reducing the AP forgetting by 2.1 percentage points. This is because the conventional ER balances the sample distributions, changing the category distribution of the training set, whereas our method preserves it, thus improving performance.

Ablation study for pseudo label selection strategies. In Tab. 3, we show the results for two pseudo-label selection strategies: (1) selecting top-K most-confident non-background predictions (Rows 1–3); and (2) selecting the predictions using a threshold for the prediction scores (Rows 4–6). We observe the first strategy works better, with peak AP when K=10. The maximum performance difference is only 0.4 percentage points when using different values for K. This indicates our method is robust to its hyper-parameter settings.

Visualizations. Figure 4 visualizes the old category pseudo (blue) and ground-truth (green) bounding boxes in some training samples in COCO 2017. In Fig. 4 (a, b), CL-DETR generates accurate pseudo bounding boxes that exactly match the ground-truth ones. This shows the effectiveness of our pseudo-label selection strategy. In Fig. 4 (c, d), CL-DETR fails to generate pseudo bounding boxes for all objects in the images when there are too many. This is explained by our strategy of selecting the top-K most-confident non-background bounding boxes as the pseudo-labels followed by removing the ones that overlap with the new category ground-truth labels excessively. In this manner, the number of pseudo bounding boxes is always smaller than K. The trade-off, justified by our improvements in the experiments, is to prefer correct although possibly incomplete annotations to contradictory or noisy ones.

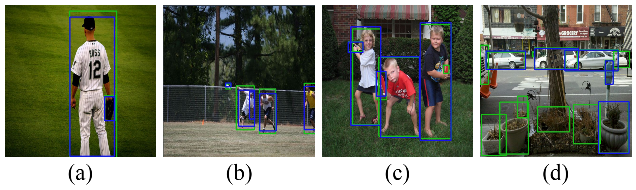


Figure 4. Visualizations of the old category pseudo (blue) and ground-truth (green) bounding boxes on COCO 2017 using the 70 + 10 setting. **(a, b)**: Our method generates accurate pseudo bounding boxes that exactly match the ground-truth ones. **(c, d)**: When there are too many annotations in the images, generated pseudo bounding boxes cannot cover all ground-truth ones. However, the pseudo bounding boxes are still focused on the foreground objects.

5. Conclusions

This paper introduced CL-DETR, a novel IOD method that can effectively use KD and ER in transformer-based detectors. CL-DETR improves the standard KD loss by introducing DKD which selects the most informative predictions from the old model, rejecting redundant background predictions, and ensuring that the distilled information is consistent with the new ground-truth evidence. CL-DETR also improves ER by selecting exemplars to match the distribution of the training set. CL-DETR is fairly generic and can be easily applied to different transformer-based detectors, including Deformable DETR [60] and UP-DETR [10], achieving large improvements. We have also defined a more realistic IOD benchmark protocol that avoids using duplicated images in different training phases. In the future, we plan to extend our method to more challenging settings such as online learning.

Ethics. We use the COCO dataset in a manner compatible with their terms; this data contains personal information (faces). For further details on ethics, data protection, and copyright please see https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html.

Acknowledgments. C. R. and A. V. are supported by ERC-UNION-CoG-101001212. C. R. is also supported by VisualAI EP/T028572/1.

References

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 3366–3375, 2017. 2

[2] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *CVPR*, pages 11254–11263, 2019. 1, 2

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pages 8218–8227, 2021. 2

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009. 11

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

[6] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 241–257, 2018. 1

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *ICLR*, 2019. 2

[8] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *IJCNN*, pages 1–7, 2019. 2

[9] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018. 2

[10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021. 1, 2, 3, 4, 6, 7, 8

[11] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *TPAMI*, 44(7):3366–3385, 2021. 2

[12] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020. 1, 2

[13] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *CVPR*, pages 9427–9436, 2022. 1, 2, 3, 6, 7, 11, 12

[14] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 2

[15] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qi Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *ICME*, pages 1–6, 2019. 2

[16] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. In *NIPS Workshops*, 2014. 4

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 1, 2, 3, 5, 11

[18] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *CVPR*, pages 3957–3966, 2021. 1

[19] K. J. Joseph, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Vineeth N Balasubramanian. Energy-based latent aligner for incremental learning. In *CVPR*, pages 7452–7461, 2022. 2

[20] K. J. Joseph, Salman H. Khan, Fahad Shahbaz Khan, and Vineeth N. Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021. 2, 5

[21] K. J. Joseph, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *TPAMI*, 2021. 1

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, pages 3521–3526, 2017. 1

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

[24] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 6

[25] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry P. Heck. RILOD: near real-time incremental learning for object detection at the edge. In Songqing Chen, Ryokichi Onishi, Ganesh Ananthanarayanan, and Qun Li, editors, *SEC*, pages 113–126, 2019. 7, 11, 12

[26] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020. 2

[27] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2018. 1, 2, 4, 6, 7, 11, 12

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 3, 6, 11, 12

[29] Fanfan Liu, Haoran Wei, Wenzhe Zhao, Guozhen Li, Jingquan Peng, and Zihao Li. WB-DETR: Transformer-based detector without backbone. In *ICCV*, pages 2979–2987, 2021. 3

[30] Xialei Liu, Chenshen Wu, Mikel Menta, Luis Herranz, Bogdan Raducanu, Andrew D Bagdanov, Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. In *CVPR Workshops*, pages 226–227, 2020. 1

[31] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learn-

ing for object detection. *arXiv preprint arXiv:2002.05347*, 2020. 2, 5, 8, 13

[32] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021. 5

[33] Yaoyao Liu, Bernt Schiele, and Qianru Sun. RMM: reinforced memory management for class-incremental learning. In *NeurIPS*, pages 3478–3490, 2021. 1

[34] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, pages 12245–12254, 2020. 1, 2, 3, 11

[35] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, pages 6467–6476, 2017. 2

[36] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[37] K. McRae and P. Hetherington. Catastrophic interference is eliminated in pre-trained networks. In *CogSci*, 1993. 1

[38] Can Peng, Kun Zhao, and Brian C. Lovell. Faster ILOD: incremental learning for object detectors based on faster RCNN. *Pattern Recognition Letter*, 140:109–115, 2020. 1, 2

[39] Can Peng, Kun Zhao, Sam Maksoud, Meng Li, and Brian C. Lovell. SID: incremental learning for anchor-free object detection via selective and inter-related distillation. *CVIU*, 210:103229, 2021. 6, 7, 11, 12

[40] Mozhgan PourKeshavarzi, Guoying Zhao, and Mohammad Sabokrou. Looking back on learned experiences for class/-task incremental learning. In *ICLR*, 2022. 2

[41] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020. 2

[42] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017. 1, 2, 3, 6, 7, 8, 11, 13

[43] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2017. 2

[44] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, pages 2990–2999, 2017. 2

[45] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, pages 3420–3429, 2017. 1, 2, 3

[46] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *CVPR*, pages 1591–1600, 2021. 2

[47] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *CVPR*, pages 2325–2333, 2016. 3

[48] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *ICCV*, pages 3611–3620, 2021. 3

[49] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *ECCV*, pages 254–270, 2020. 2

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

[51] Eli Verwimp, Kuo Yang, Sarah Parisot, Hong Lanqing, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Re-examining distillation for continual object detection. In *BMVC*, 2022. 1

[52] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *ECCV*, 2022. 2

[53] Liyuan Wang, Xingxing Zhang, Kuo Yang, Longhui Yu, Chongxuan Li, Lanqing Hong, Shifeng Zhang, Zhenguo Li, Yi Zhong, and Jun Zhu. Memory replay with data compression for continual learning. In *ICLR*, 2022. 1

[54] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 5

[55] Binbin Yang, Xinchi Deng, Han Shi, Changlin Li, Gengwei Zhang, Hang Xu, Shen Zhao, Liang Lin, and Xiaodan Liang. Continual object detection via prototypical task correlation guided gating mechanism. In *CVPR*, pages 9255–9264, 2022. 1

[56] Dongbao Yang, Yu Zhou, Aoting Zhang, Xurui Sun, Dayan Wu, Weiping Wang, and Qixiang Ye. Multi-view correlation distillation for incremental object detection. *Pattern Recognition*, 131:108863, 2022. 2, 11

[57] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[58] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13208–13217, 2020. 1

[59] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020. 2

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 2, 3, 4, 6, 7, 8, 12, 13

Supplementary materials

We present the following supplementary content: results with the traditional IOD benchmark protocol (§A), more ablation results (§B), more visualization results (§C), and instructions for our code (§D).

A. Traditional IOD protocol and results

This is supplementary to Section 4.2. In previous work [13], in each phase, the incremental detector is allowed to observe all images that contain a certain type of objects. Because images often contain a mix of object classes, both old and new, this means that the same images can be observed in different training phases. This is incompatible with the standard definition of incremental learning [17, 34, 42] where, with the exception of the examples deliberately stored in the exemplar memory, the images observed in different phases do not repeat. Thus, we provide results with our new IOD benchmark protocol in the main paper.

For completeness and comparison, here we also evaluate performance using the traditional IOD benchmark protocol used in [13], and provide comparison results between our method and other top-performing IOD methods with this protocol.

Traditional IOD protocol. Formally, let $\mathcal{D} = \{(x, y)\}$ be a dataset of images x with corresponding object annotations y, such as COCO 2017 [28], and let $\mathcal{C} = \{1, \ldots, C\}$ be the set of object categories. We adopt such a dataset for benchmarking IOD as follows. First, we partition $\mathcal{C}$ into M subsets $\mathcal{C} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_M$, one for each training phase. For each phase i, we modify the samples $(x, y) \in \mathcal{D}$, where y only contains annotations for objects of class $\mathcal{C}_i$ and drop the others. In phase i of training, the model is only allowed to observe images that contain at least one annotation for objects of types $\mathcal{C}_i \subset \mathcal{C}$.

Experiment results. Table S1 shows that also with the traditional IOD protocol our CL-DETR consistently performs better than the state-of-the-art [13] and other IOD methods [25, 27, 39]. Interestingly, our method achieves better performance than other methods [13, 25, 27, 39] even without using exemplars. For example, the AP of our CL-DETR *w/o* ER is 2.3 percentage points higher than the AP of ERD [13] in the $40 + 40$ setting.

B. More ablation results

This is supplementary to Section 4.2.

Ablation results for λ. In Tab. S2, we show the ablation results for λ on COCO 2017 in the 70+10 setting. We can observe the peak AP is at λ=0.7, with a maximum performance difference of only 1.0 percentage points using different values. This demonstrates the robustness of our method to different λ values. Further results and analysis will be included in the final paper.

Separate validation sets. In Tab. S3, we provide ablation results for different pseudo label selection strategies on a separate validation set (COCO 2017, 70+10 setting). Results show that the "top-K selection" strategy performs best, consistent with the findings in the main paper.

Iteratively improves detection. We apply curriculum learning [4] for the hyperparameter, p, i.e., decreasing p from 0.5 to 0.1 during the training. This way, the loss for objects with low confidence will be ignored in the beginning and only included later when the model becomes more stable. Table S4 shows the results on COCO 2017 in the 70+10 setting. Curriculum learning for p slightly improves (+0.3 AP) the final performance.

More fine-grained ablation results. Table S5 presents partial fine-grained results using Deformable DETR on COCO 2017 in the 70+10 setting. Results show that our method, CL-DETR, outperforms related methods such as LwF and iCaRL in terms of AP, old category AP, and FPP. These results highlight the effectiveness of our CL-DETR in addressing the forgetting problem.

Different exemplar replay methods. In Tab. S6, we provide ablation results for different exemplar replay methods. Our "distribution-persevering" exemplar replay strategy achieves better performance (higher AP and lower FPP) compared to the existing strategies in the related works [42, 56]. This shows that creating an exemplar set that follows the natural data distribution of COCO 2017 improves the results, compared to existing strategies that try to select a category-balanced subset of the data as the exemplar set and thus change the original data distribution.

C. More visualization results

This is supplementary to Section 4.2. Figure S1 visualizes the one-to-one matching between the merged bounding boxes and new model predictions (yellow) in some training samples in COCO 2017. The merged bounding boxes include the old category pseudo (blue) and new category ground-truth (green) bounding boxes. We can observe that the old category pseudo and new category ground-truth bounding boxes are complementary and indicate the old and new category objects, respectively. It shows our method successfully resolves conflicts between pseudo and ground-truth bounding boxes and ensures the model ignores background predictions.

D. Source Code in PyTorch

In the following, we introduce how to install the environment and run the code.

Setting	Method	Detection baseline	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
40+40	LwF [27]	GFLv1	17.2	25.4	18.6	7.9	18.4	24.3
	RILOD [25]	GFLv1	29.9	45.0	32.0	15.8	33.0	40.5
	SID [39]	GFLv1	34.0	51.4	36.3	18.4	38.4	44.9
	ERD [13]	GFLv1	36.9	54.5	39.6	21.3	40.4	47.5
	CL-DETR *w/o* ER	Deformable DETR	39.2$\pm$0.2	56.1$\pm$0.3	42.6$\pm$0.4	21.0$\pm$0.3	42.8$\pm$0.4	52.6$\pm$0.3
	CL-DETR	Deformable DETR	**42.0**$\pm$0.3	**60.1**$\pm$0.2	**45.9**$\pm$0.3	**24.0**$\pm$0.3	**45.3**$\pm$0.2	**55.6**$\pm$0.4
70+10	LwF [27]	GFLv1	7.1	12.4	7.0	4.8	9.5	10.0
	RILOD [25]	GFLv1	24.5	37.9	25.7	14.2	27.4	33.5
	SID [39]	GFLv1	32.8	49.0	35.0	17.1	36.9	44.5
	ERD [13]	GFLv1	34.9	51.9	37.4	18.7	38.8	45.5
	CL-DETR *w/o* ER	Deformable DETR	35.8$\pm$0.3	53.5$\pm$0.2	39.5$\pm$0.3	19.4$\pm$0.3	41.5$\pm$0.3	46.1$\pm$0.4
	CL-DETR	Deformable DETR	**40.4**$\pm$0.2	**58.0**$\pm$0.3	**43.9**$\pm$0.2	**23.8**$\pm$0.4	**43.6**$\pm$0.3	**53.5**$\pm$0.3

Table S1. Supplementary to Table 1 (main paper). IOD results (%) on COCO 2017 with the traditional IOD protocol [13]. "CL-DETR" and "CL-DETR *w/o* ER" are our methods. For "CL-DETR *w/o* ER", we don't save any exemplars. For "CL-DETR", the total memory budget for the exemplars is set as 10% of the total dataset size. The results for the related methods [13, 25, 27, 39] are from [13]. In the $A+B$ setup, in the first phase, we observe a fraction $\frac{A}{A+B}$ of the training samples with A categories annotated. Then, in the second phase, we observe the remaining $\frac{B}{A+B}$ of the training samples, where B new categories are annotated. We test settings $A+B = 40+40$ and $70+10$. We run experiments for three different categories and data orders and report the average AP with 95% confidence interval.

Setting	KD	Our KD	KD-oracle	ER	Our ER	ER-oracle
AP	24.5	33.9	36.1	33.3	36.1	36.5

Table S2. Ablation results (%) for λ on COCO 2017 in the $70+10$ setting.

Setting	K=5	K=10	K=20	p≥0.1	p≥0.3	p≥0.5
AP	39.1	39.9	39.5	38.6	38.9	38.2

Table S3. Ablation results (%) for different pseudo label selection strategies on a separate validation set (COCO 2017, $70+10$ setting).

Setting	p≥0.1	p≥0.3	p≥0.5	Curriculum for p
AP	38.6	38.9	38.2	39.2

Table S4. Ablation results (%) for curriculum learning on COCO 2017 in the $70+10$ setting.

Method	All categories ↑				Old categories ↑				FPP ↓			
	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
LwF	24.5	12.4	28.2	35.2	24.0	12.3	27.7	34.4	19.3	13.5	18.2	23.1
iCaRL	35.9	19.1	39.4	48.6	36.8	20.3	39.9	50.0	6.5	5.5	6.0	7.5
Ours	40.1	23.2	43.2	52.1	41.8	24.5	44.7	54.6	1.5	1.3	1.2	2.9

Table S5. Supplementary to Table 2 (main paper). More fine-grained ablation results (%) for KD and ER, using Deformable DETR [60] on COCO 2017 in the $70+10$ setting.

Installation. To run this project, please install Python 3.7 with Anaconda.

```
conda create -n cl_detr python=3.7
```

Activate the environment as follows,

```
conda activate cl_detr
```

Install PyTorch and torchvision. For example, for CUDA version is 9.2, install PyTorch and torchvision as follows,

```
conda install pytorch=1.5.1 torchvision=0.6.1
    cudatoolkit=9.2 -c pytorch
```

We install other requirements as follows,

```
pip install -r requirements.txt
```

Finally, we compile the CUDA operators as follows,

```
cd ./models/ops
sh ./make.sh
# unit test (should see all checking is True)
python test.py
```

Running experiments. First, please download COCO 2017 [28], and set up the dataset as in Deformable DETR [60].

The following command runs the experiments:

```
GPUS_PER_NODE=4 ./tools/run_dist_launch.sh 4 ./
    configs/r50_deformable_detr.sh
```

Settings can be changed in "main.py".

Row	Exemplar replay strategies	All categories ↑				Old categories ↑				FPP ↓			
		AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
1	Random	37.9	20.8	40.9	50.4	39.0	21.6	41.7	52.3	4.3	4.2	4.2	5.2
2	Herding [42]	38.1	22.5	41.0	49.3	39.0	23.2	41.6	50.4	4.3	2.6	4.3	7.1
3	Adaptive sampling [31]	38.5	22.7	41.4	49.9	39.4	23.5	42.1	51.2	3.9	2.3	3.8	6.3
4	Distribution-preserving calibration (ours)	**40.1**	**23.2**	**43.2**	**52.1**	**41.8**	**24.5**	**44.7**	**54.6**	**1.5**	**1.3**	**1.2**	**2.9**

Table S6. Supplementary to Table 2 (main paper). Ablation results (%) for different exemplar replay strategies, using Deformable DETR [60] on COCO 2017 in the $70 + 10$ setting. "Herding" and "adaptive sampling" are from [42] and [31], respectively.

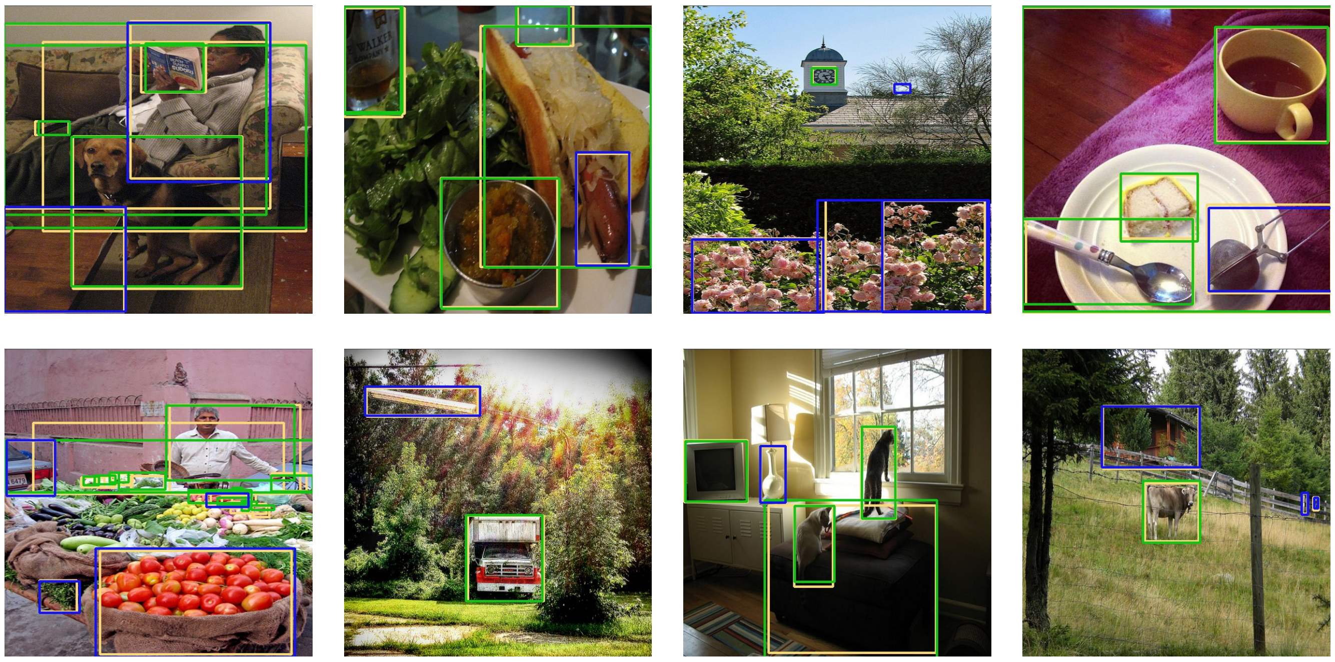


Figure S1. Supplementary to Section 4.2 (main paper). Visualizations of the one-to-one matching between the merged bounding boxes and new model predictions (yellow) on COCO 2017 using the $70 + 10$ setting. The merged bounding boxes include the old category pseudo (blue) and new category ground-truth (green) bounding boxes. Our method ensures the old category pseudo and new category ground-truth bounding boxes are merged successfully.