Continual Detection Transformer for Incremental Object Detection
Supplementary Materials

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

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). The source files of our code are available at https://lyy.mpi-inf.mpg.de/CL-DETR/

A. Traditional IOD protocol and results

This is supplementary to Section 4.2. In previous work [2], 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 [3, 8, 10] 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 [2], 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 [6], 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 [2] and other IOD methods [4, 5, 9]. Interestingly, our method achieves better performance than other methods [2, 4, 5, 9] 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 [2] 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 [1] 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.

Setting	Method	Detection baseline	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
40+40	LwF [5]	GFLv1	17.2	25.4	18.6	7.9	18.4	24.3
	RILOD [4]	GFLv1	29.9	45.0	32.0	15.8	33.0	40.5
	SID [9]	GFLv1	34.0	51.4	36.3	18.4	38.4	44.9
	ERD [2]	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$0.4
70+10	LwF [5]	GFLv1	7.1	12.4	7.0	4.8	9.5	10.0
	RILOD [4]	GFLv1	24.5	37.9	25.7	14.2	27.4	33.5
	SID [9]	GFLv1	32.8	49.0	35.0	17.1	36.9	44.5
	ERD [2]	GFLv1	34.9	51.9	37.4	18.7	38.8	45.5
	CL-DETR *w/o* ER	Deformable DETR	35.8$\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 [2]. "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 [2, 4, 5, 9] are from [2]. 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	$p\geq0.1$	$p\geq0.3$	$p\geq0.5$
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	$p\geq0.1$	$p\geq0.3$	$p\geq0.5$	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 [12] on COCO 2017 in the $70+10$ setting.

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 [10, 11]. 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

We provide our PyTorch code at https://lyy.mpi-inf.mpg.de/CL-DETR/

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

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

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

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 [10]	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 [7]	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 [12] on COCO 2017 in the $70 + 10$ setting. "Herding" and "adaptive sampling" are from [10] and [7], respectively.

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 [6], and set up the dataset as in Deformable DETR [12].

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".

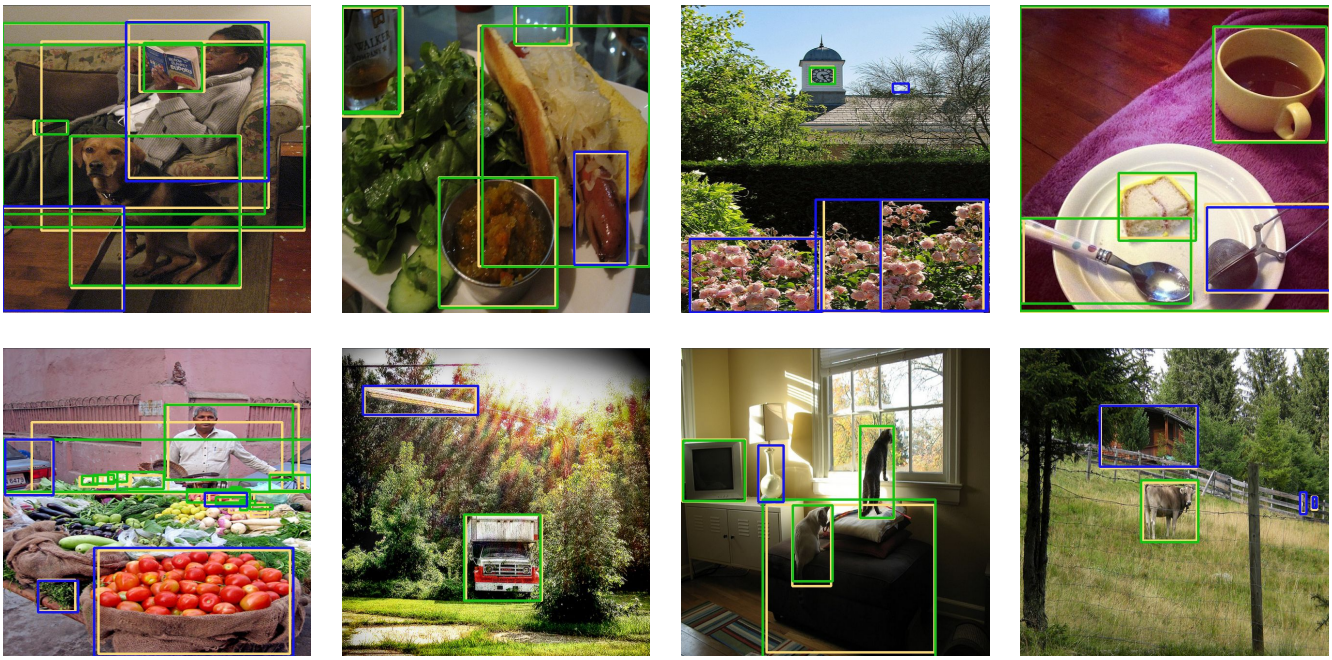

Figure S1. Supplementary to Section 4.2 (main paper). Visualizations of the one-to-one matching between the merged bounding boxes and new model predictions (yellow) on COCO 2017 using the $70 + 10$ setting. The merged bounding boxes include the old category pseudo (blue) and new category ground-truth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

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009. 1

[2] 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] 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

[4] 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. 1, 2

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

[6] 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. 1, 3

[7] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learning for object detection. *arXiv preprint arXiv:2002.05347*, 2020. 3

[8] 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

[9] 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. 1, 2

[10] 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

[11] 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

[12] 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. 2, 3