Learning Endogenous Attention for Incremental Object Detection

Supplementary Material

The supplementary material is organized as follows: Section 1 introduces more experimental setup, Section 2 provides more experimental results on VOC 2007, Section 3 presents additional ablation studies, Section 4 includes further visualization results, and Section 5 provides the PyTorch implementation of our code.

1. More experimental setup

Datasets. We conduct IOD experiments on MS COCO 2017 and Pascal VOC 2007 datasets, which are widely used in previous IOD works. The COCO 2017 consists of 80 diverse classes across 118,000 images for training and 5,000 images for testing. These classes are strategically divided based on our experimental protocols. The VOC 2007 contains 20 distinct classes and 9,963 images, with 50% of them allocated for training and the other half reserved for testing.

Distribution of image task IDs. In Figure S1(left), we illustrate the task ID distributions on the COCO dataset with different data splits, which shows that about 30–40% images contain multiple task IDs. Our ETM infers multiple task IDs by assigning each image to tasks with low energy. Figure S1(right) compares performances of our LEA with the state-of-the-art ABR method to further demonstrate the superiority of LEA when the task number increases (*i.e.*, $40 + 40 \rightarrow 40 + 2 \times 20 \rightarrow 40 + 4 \times 10$), wherein LEA steadily and significantly outperforms ABR by a large margin (at least 5% improvement), and the performance decrease of LEA is much slighter than the ones of ABR ($41.2 \rightarrow 39.9 \rightarrow 39.3$ v.s. $36.2 \rightarrow 32.7 \rightarrow 31.2$). Thus, our LEA 1) is a simple yet efficient method that is technically capable of handling multiple task IDs in a single image and 2) significantly outperforms the SOTA method when the task number increases.

Evaluation indicators of ETM. We employ precision (P), recall (R), and $F1$ scores to assess the accuracy of the task ID predictions generated by ETM. Specifically, P reflects the proportion of correctly predicted positives among all samples labeled as positive by the model. R, on the other hand, quantifies the model's ability to identify true positive examples from all actual positive examples. The $F1$ score, as the harmonic mean of precision and recall, offers a balanced perspective, comprehensively evaluating both aspects:

$$P = \frac{TP}{TP + FP}, \tag{1}$$

$$R = \frac{TP}{TP + FN}, \tag{2}$$

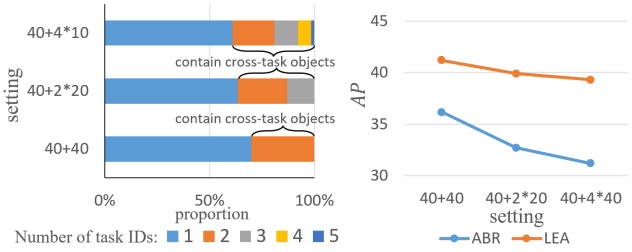


Figure S1. Supplementary to Section 4.1 (main paper). Task ID distributions on the COCO dataset with different data splits.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2 * P * R}{P + R}, \tag{3}$$

where True Positives (TP) signify samples where both the model's prediction and the actual value are positive, indicating a correct prediction of a positive sample. False Positives (FP) occur when the model predicts a positive outcome, but the actual value is negative, resulting in an incorrect prediction of a positive sample. Conversely, False Negatives (FN) arise when the model predicts a negative outcome for an actually positive example, misclassifying a positive sample as negative.

Energy-based Task Modulator Setup. 1) *Baseline method:* We use a frozen pre-trained ViT-B/16 as the shared backbone and train a prompt and a linear classifier for each task. The prompt and backbone outputs are concatenated and fed into the classifier. 2) *Loss function:* Each task is trained using Binary Cross-Entropy loss: $L(x, y) = -[y \cdot log(\sigma(x)) + (1 - y) \cdot log(1 - \sigma(x))]$, where x are output logits, y are labels, and σ is the sigmoid function. 3) *Classifier extension:* All existing prompts and classifiers are frozen to preserve old knowledge, while a new prompt and a new classifier are added for each new task.

2. More results on VOC 2007

More comparison methods. In Table S1, we compare more methods on the VOC 2007 dataset, including two model-expansion-based IOD methods—MultIOD [1] and DIODE [7], ABR [6], ABR combined with pseudo-labels (ABR+PL), and the performance upper bound (fully supervised training). Among them, MultIOD and DIODE achieve incremental learning by expanding the detection head of the model. From the table, it can be observed that our LEA method significantly outperforms these comparative methods. Additionally, integrating pseudo-labels (PL)

Method	Baseline	$15+5$	$10+10$	$5+3\times5$
Upper Bound	ViTDet	80.3	80.3	80.3
MultIOD [1]	ViTDet	67.8	66.4	59.0
DIODE [7]		63.5	62.6	56.7
ABR		67.7	66.8	59.8
ABR+PL		68.2	67.1	58.5
LEA (Ours)		**73.5**	**72.2**	**63.3**

Table S1. Supplementary to Table 3 (main paper). More comparison results on VOC 2007.

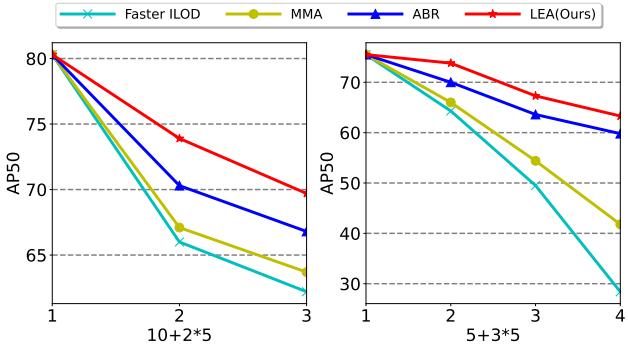


Figure S2. Supplementary to Table 3 (main paper). Experimental results (AP_{50}) in the multi-task settings of the VOC 2007 dataset.

predicted by the old model into ABR only yields a marginal improvement. This is because, although pseudo-labels alleviate the label conflict issue, they also introduce additional noisy annotations. Therefore, simply combining pseudo-labels with existing methods is not an effective solution to the IOD problem.

Detailed results of two-task settings. In Table S2, we present comprehensive experimental results, including the AP_{50} for each category, the average performance across all categories, and the additional detector memory (MB) required by our method in the $15+5$ and $10+10$ settings. To be specific, Old($\cdot$) denotes an old model trained on data from old classes, while Fine-tuning represents fully supervised training on new classes, leading to overfitting new classes and severe forgetting of old classes. In the $15+5$ setting, our proposed LEA outperforms all previous methods, with an average AP_{50} surpassing the second-best OSR by **4.0%**. Notably, our method only requires saving a minimal additional detector memory of 0.2 MB in this setting, outperforming previous methods that rely on storing representative samples. Similarly, in the $10+10$ setting, our method maintains its superiority, surpassing the second-best ABR by **1.3%** in average AP_{50}, while only consuming 0.9 MB additional detector memory.

Detailed results of multi-task settings. Figure S2 presents the detailed comparison results (AP_{50}) in multi-task settings of VOC 2007. The horizontal axis represents the average accuracy across previously learned categories. It is noticed that starting from the same accuracy in task 1, our method consistently outperforms the comparison method throughout the incremental process, demonstrating LEA's suitability for practical multi-task scenarios.

3. More ablation results

3.1. Classification accuracy for objects of different sizes.

We evaluate the classification $F1$ score of our ETM module on VOC and COCO datasets at different sizes (**small/medium/large**). Since classification models cannot predict objects' size, in Table S3, we select three subsets for COCO and VOC, respectively, each containing only small/medium/large objects, to evaluate the $F1$ score. It shows that our ETM achieves comparable accuracy on small and medium objects compared to large ones, indicating the capability of our method in handling small/medium objects.

3.2. Sensitivity of hyper-parameter γ

As described in Section 3.4 of the main paper, γ is a hyper-parameter related to the threshold τ to determine whether a test image belongs to a certain task. To determine the appropriate γ, in Table S4, we analyze the effect of ETM's hyper-parameter γ on the detection results in the $40+40$ setting of COCO 2017. From the table, it is evident that $\gamma=1$ yields the optimal $F1$ score. However, when γ is reduced to 0.1, P decreases, R increases, and AP also improves. This indicates that recall is more associated with model accuracy, whereas precision plays a crucial role in accurately selecting task IDs, thus reducing inference time. Further reducing γ to 0.01 results in unchanged AP with a slight decrease in P. Therefore, to minimize inference time while preserving accuracy, we set $\gamma=0.1$ in our experiments. Additionally, compared to Oracle (*i.e.* ground truth task IDs), our ETM only incurs a minimal loss of **1.6%** in AP and **2.4%** in AP_{50}, proving its efficacy.

3.3. Effect of parallel and series EAM

In Table S5, we compare the experimental results achieved by the parallel structure EAMs (illustrated in Figure S3) and the series structure used in the main paper, in the $10+10$ setting of VOC 2007. Notably, the table reveals that the series structure outperforms the parallel structure. This superiority can be attributed to the series structure's ability to capitalize on the abundant feature representations generated by multi-head attention and further refine them through subsequent layers, facilitating the model's enhanced focus on new category objects. Conversely, the parallel structure loses these rich feature representations produced by the multi-head attention, thus hindering the model's overall performance.

15 + 5 Setting	Baseline	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AP_{50}	Mem.
Old(1-15)	ViTDet	88.8	82.2	78.4	64.7	63.6	88.5	88.0	89.5	63.6	81.9	79.6	88.4	89.3	82.7	79.8	-	-	-	-	-	60.5	-
Fine-tuning	ViTDet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.4	40.0	60.1	61.8	70.4	13.5	-
Yang [8]	Faster R-CNN	66.8	74.9	69.7	51.8	52.4	64.4	82.1	83.2	46.7	73.7	62.2	80.8	83.2	76.7	75.9	39.0	64.3	61.9	67.0	64.0	67.0	-
OWOD [3]	Faster R-CNN	75.4	81.0	67.1	51.9	55.7	77.2	85.6	81.7	46.1	76.2	55.4	76.7	86.2	78.5	82.1	32.8	63.6	54.7	77.7	64.6	68.5	58
Joseph [4]	Faster R-CNN	78.4	79.7	66.9	54.8	56.2	77.7	84.6	79.1	47.7	75.0	61.8	74.7	81.6	77.5	80.2	37.8	58.0	54.6	73.0	56.1	67.8	13
OSR [9]	Faster R-CNN	75.8	76.4	72.2	54.1	52.7	73.8	84.0	79.9	52.4	75.5	68.7	81.5	83.4	76.5	76.5	41.7	68.7	53.7	75.3	66.5	69.5	0.4
MMA [2]	ViTDet	47.8	65.5	61.0	40.5	47.6	64.8	77.9	70.2	43.3	70.2	61.5	78.0	76.9	70.3	76.5	55.3	71.4	77.0	78.7	69.8	65.2	-
ABR [6]	ViTDet	50.1	70.4	62.2	42.8	51.3	66.2	78.4	80.1	51.9	74.2	62.3	78.8	79.8	69.1	77.9	52.7	70.6	76.8	77.5	69.1	67.7	21.0
LEA(Ours)	ViTDet	80.3	77.3	71.2	62.2	58.9	73.4	80.4	82.0	59.6	78.0	66.4	81.1	81.2	80.7	79.1	56.0	75.3	78.1	79.0	70.1	**73.5**	0.2
10 + 10 Setting	Baseline	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AP_{50}	Mem.
Old(1-10)	ViTDet	86.5	83.2	81.4	65.8	65.0	90.2	89.1	91.3	67.5	84.1	-	-	-	-	-	-	-	-	-	-	40.2	-
Fine-tuning	ViTDet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.8	84.3	85.9	78.1	77.2	51.6	72.4	71.0	87.7	76.3	37.8	-
Li [5]	RetinaNet	71.7	81.7	66.9	49.6	58.0	65.9	84.7	76.8	50.1	69.4	67.0	72.8	77.3	73.8	74.9	39.9	68.5	61.5	75.5	72.4	67.9	-
Zhang [10]	Faster R-CNN	68.6	71.2	73.1	48.1	56.0	64.4	81.9	77.8	49.4	67.8	61.5	67.7	67.5	52.2	74.0	37.8	63.0	55.5	65.3	72.4	63.8	-
Yang [8]	Faster R-CNN	73.3	67.5	68.4	52.1	52.6	75.2	81.4	77.3	41.7	73.8	57.6	79.4	76.7	74.7	72.2	31.7	68.0	61.3	73.0	65.4	66.2	-
OWOD [3]	Faster R-CNN	63.5	70.9	58.9	42.9	34.1	76.2	80.7	76.3	34.1	66.1	56.1	70.4	80.2	72.3	81.8	42.7	71.6	68.1	77.0	67.7	64.6	38
Joseph [4]	Faster R-CNN	76.0	74.6	67.5	55.9	57.6	75.1	85.4	77.0	43.7	70.8	60.1	66.4	76.0	72.6	74.6	39.7	64.0	60.2	68.5	60.5	66.3	7.9
OSR [9]	Faster R-CNN	74.8	70.0	71.5	57.4	61.3	76.5	84.2	74.8	45.0	71.7	64.0	79.6	83.0	76.4	76.4	42.5	71.7	68.8	74.1	71.5	69.8	0.3
MMA [2]	ViTDet	52.5	66.6	61.5	52.5	50.3	66.8	66.4	62.7	40.6	69.8	60.4	83.8	78.1	77.6	77.8	60.3	70.5	72.4	80.6	70.8	66.0	-
ABR [6]	ViTDet	57.0	72.3	67.3	53.8	53.1	72.2	77.2	81.7	52.1	77.9	58.0	83.6	84.2	81.2	80.7	64.1	72.4	73.8	82.6	73.2	70.9	18.9
LEA(Ours)	ViTDet	80.8	77.2	71.8	64.1	58.8	77.1	81.1	83.0	59.0	79.8	62.8	83.7	80.4	73.2	78.1	54.1	71.0	60.5	79.3	68.9	**72.2**	0.9

Table S2. Supplementary to Table 3 (main paper). Detailed experimental results in the two-task settings of VOC 2007. Except for Old(·), MMA, ABR, and our LEA, other experimental results are borrowed from OSR.

COCO 2017	70+10 71.9/80.5/89.3	40+40 74.0/83.4/93.4	40+4×10 55.7/59.1/69.5	40+2×10 67.6/76.8/88.8
VOC 2007	15+5 76.9/78.8/93.2	10+10 72.8/74.9/89.9	5+3×5 65.6/69.1/76.8	10+2×5 73.2/72.5/82.1

Table S3. $F1$ score (small/medium/large) for objects of different sizes.

Threshold	P	R	$F1$	AP	AP_{50}	AP_{75}
$\gamma = 1$	81.5	93.5	87.1	40.6	58.9	44.2
$\gamma = 0.5$	78.3	95.2	85.9	40.9	59.4	44.5
$\gamma = 0.1$	75.0	96.5	84.4	41.2	59.8	44.8
$\gamma = 0.05$	74.3	96.8	84.1	41.2	59.8	44.8
$\gamma = 0.01$	73.9	96.9	83.9	41.2	59.8	44.8
Oracle	100.0	100.0	100.0	42.8	62.2	46.6

Table S4. Supplementary to Section 4.1 (main paper). Ablation study of hyper-parameter γ in the $40 + 40$ setting of COCO 2017.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Parallel EAM	47.3	70.5	52.0	15.6	35.1	56.8
Series EAM	**48.8**	**72.2**	**53.7**	**15.9**	**35.7**	**58.4**

Table S5. Comparison results of parallel and series EAM in the $10 + 10$ setting of VOC 2007.

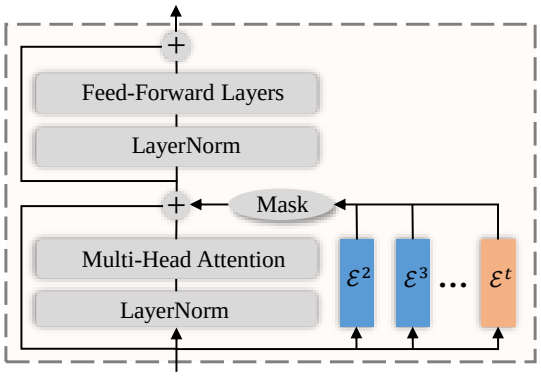


Figure S3. Structure of parallel EAMs.

Method	AP_S	AP_M	AP_L	time (ms)
MMA	19.0	39.8	43.9	149.0
ABR	19.6	40.7	46.2	149.0
LEA (Ours)	**28.5**	**48.8**	**59.2**	149.4

Table S6. Inference time comparison in the 70+10 setting of COCO 2017.

3.4. Inference time

In Table S6, we analyze the inference time of our approach. It is observed that our method is computationally efficient, improving performance by at least **8%** over ABR with a latency of only **0.4** ms. This latency comes from the introduction of an additional ETM for inferring task IDs.

4. More visualization results

4.1. Visualization of detection results

The inference results in the $10 + 10$ setting of VOC 2007 are shown in Figure S4. The first column displays the test images, followed by the ground-truth annotations in the second column, the detection results of the ABR method in the third column, and our method's results in the fourth column. Our model notably preserves the detection precision for old categories like boat (first row), bus (second row), chair (third and sixth rows), and car (fifth row), as evident from the figure. Conversely, the prevalent ABR method misses the detection of several objects in the above four categories. Moreover, ABR also yields false detections for new categories, such as diningtable (second row), person (fourth row), and tvmonitor (seventh row). These findings underscore the superiority of our LEA method in mitigating the forgetting of old classes and adapting to new ones in IOD.

4.2. More heatmap visualizations

In addition to Figure 3 of the main paper, we provide further visualizations of heatmaps in the COCO $40 + 40$ setting in Figure S5. Specifically, for $N_D = 6$, Figure S5 demonstrates similar results to those observed in the main paper: 1) The heatmaps generated from block 3 in model θ^1 reveal task-agnostic features that comprehensively encompass all objects depicted in the image; 2) The heatmaps derived from block 12 in model θ^1 primarily focuses on and highlights the classes trained in task 1, encompassing cow, cat, car, bus, bird, and traffic light; 3) After additionally adding and training EAM, the heatmaps of block 12 in model $\{\theta^1, \mathcal{E}^2\}$ shifts its attention to the classes relevant to the task 2, including tv, laptop, chair, and clock. In essence, our method can effectively achieve endogenous attention to focus on task-specific objects.

5. PyTorch implementation of our code

To run our project, please follow the Build Project, Dataset Preparation, and Run Experiments instructions.

Build Project

- `conda create -n lea python=3.9`
- `conda activate lea`
- `conda install pytorch==1.12.1 torchvision==0.13.1 torchaudio==0.12.1 cudatoolkit=11.6 -c pytorch -c conda-forge`
- `python -m pip install -e LEA_project`

Dataset Preparation

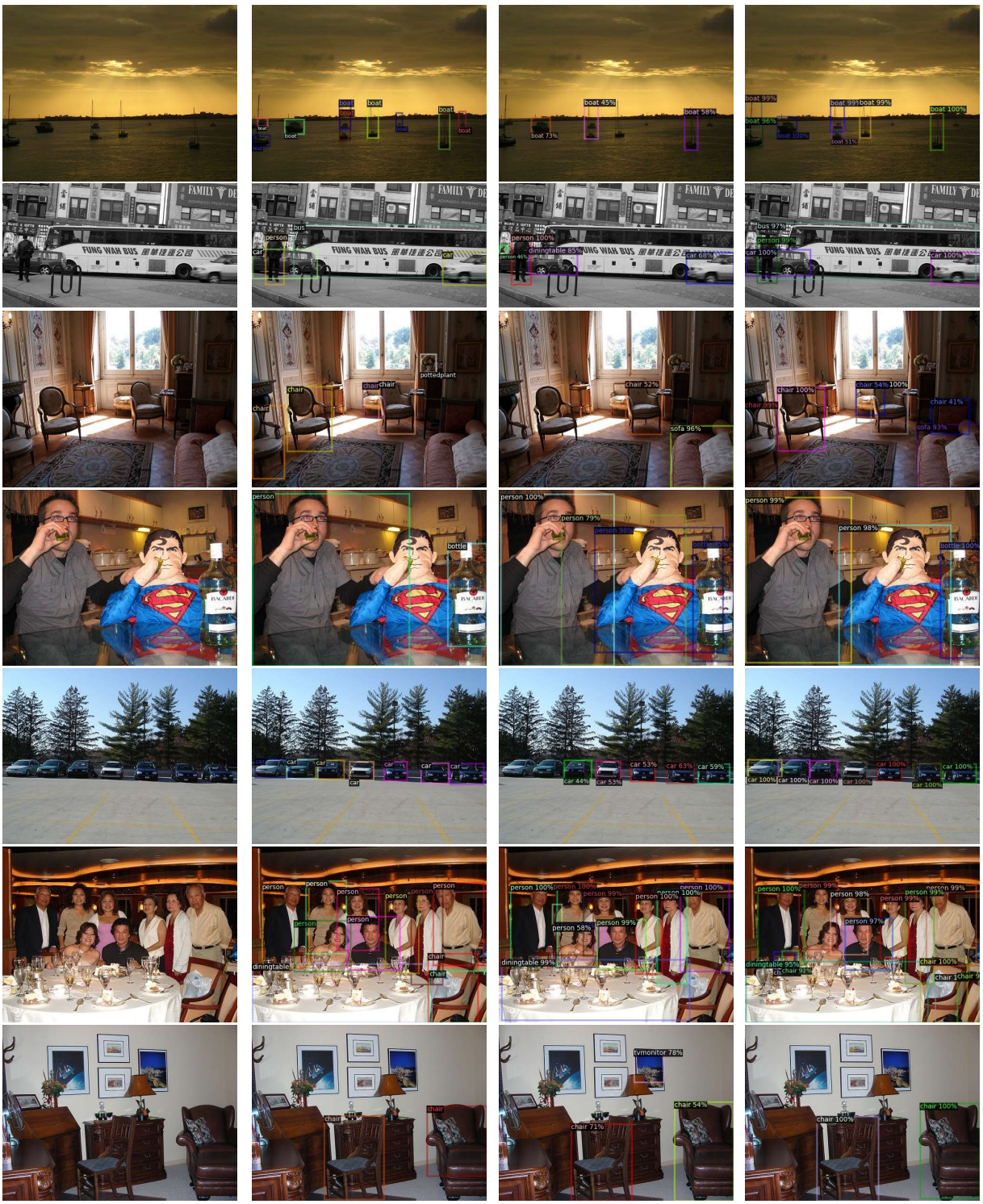
 # Download COCO 2017 and split the data.
- `python datasets/coco_deal.py`

 # Download VOC 2007, convert to COCO format, and split the data.
- `python datasets/voc_deal.py`

Run Experiments

 # LEA model training & test, settings can be changed in 'run_coco.sh' or 'run_voc.sh'.
- `python run_coco.sh`
- `python run_voc.sh`

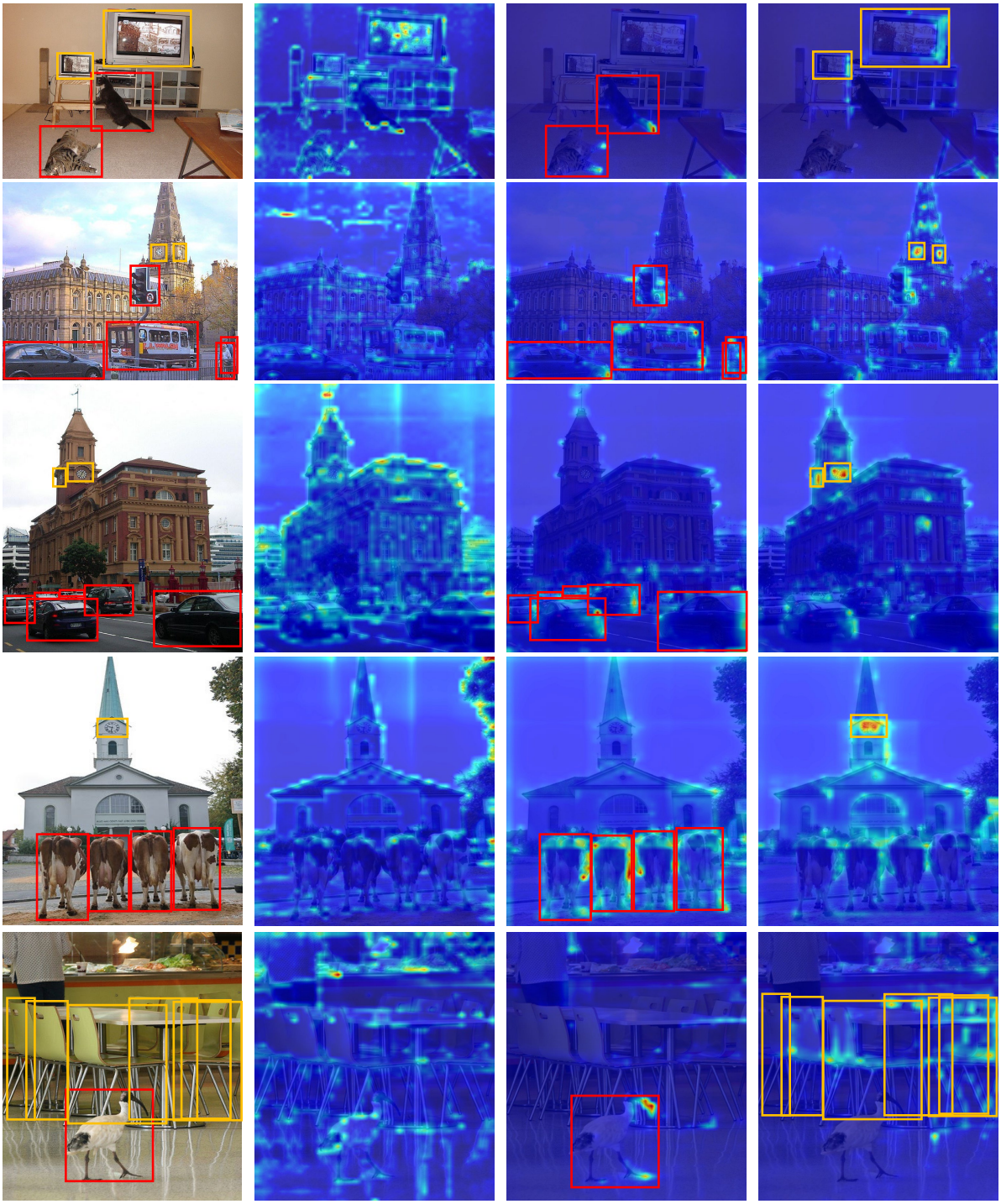
CVPR
#13478



(a) Test images (b) Ground-truth (c) ABR (d) LEA(Ours)

Figure S4. Supplementary to Section 4.3 (main paper). Visualization results in the $10 + 10$ setting of VOC 2007. The old categories involved in these figures include boat, bus, bottle, car, and chair. The new categories include diningtable, person, sofa, and tvmonitor.

CVPR
#13478



(a) Test images (b) Block 3 heatmap, model θ^1 (c) Block 12 heatmap, model θ^1 (d) Block 12 heatmap, model $\{\theta^1, \mathcal{E}^2\}$

Figure S5. Supplementary to Figure 3 (main paper). Visualization of heatmaps in the $40 + 40$ setting of COCO 2017. The red-boxed classes, encompassing cow, cat, car, bus, bird, and traffic light are learned in task 1. The orange-boxed classes, including tv, laptop, chair, and clock, are learned in task 2.

References

[1] Eden Belouadah, Arnaud Dapogny, and Kevin Bailly. Multiod: Rehearsal-free multihead incremental object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4107–4117, 2024. 1, 2

[2] Fabio Cermelli, Antonino Geraci, Dario Fontanel, and Barbara Caputo. Modeling missing annotations for incremental learning in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3700–3710, 2022. 3

[3] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5830–5840, 2021. 3

[4] KJ Joseph, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12): 9209–9216, 2021. 3

[5] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. Rilod: Near real-time incremental learning for object detection at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126, 2019. 3

[6] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost van de Weijer. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11367–11377, 2023. 1, 3

[7] Can Peng, Kun Zhao, Sam Maksoud, Tianren Wang, and Brian C Lovell. Diode: dilatable incremental object detection. *Pattern Recognition*, 136:109244, 2023. 1, 2

[8] 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. 3

[9] Dongbao Yang, Yu Zhou, Xiaopeng Hong, Aoting Zhang, and Weiping Wang. One-shot replay: Boosting incremental object detection via retrospecting one object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3127–3135, 2023. 3

[10] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1131–1140, 2020. 3