Supplementary Materials: Augmented Box Replay: Overcoming Foreground Shift for Incremental Object Detection

Yuyang Liu[1,2,3] Yang Cong[4] Dipam Goswami[5] Xialei Liu[6] Joost van de Weijer[5,7]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences
[2] Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences [4]South China University of Technology
[5]Computer Vision Center, Barcelona [6]VCIP, CS, Nankai University
[7]Department of Computer Science, Universitat Autònoma de Barcelona
liuyuyang@sia.cn, congyang81@gmail.com, {dgoswami, joost}@cvc.uab.es, xialei@nankai.edu.cn

A. Additional Methods

A.1. Prototype Box Selection

This method involves selecting the most representative boxes, as prototypes, from the current training data, which are then replayed along with the future training data. The memory buffer is commonly denoted as B^t, where t represents the current task and the size M of B^t is limited. Therefore, the selection is an important factor that affects the performance. We employ a frozen trained model to generate the Region of Interest (RoI)-Aligned feature maps $\{F_g^t \in \mathbb{R}^{C\times S\times S}\}_{g=1}^{G_n^t}$ for G_n^t groundtruth boxes in the current task t, where C is the number of feature planes and S is the spatial dimension. Then, a prototype feature map $\hat{F}_c^t$ for each class $c \in \mathcal{C}^t$ can be computed by:

$$\hat{F}_c^t = \frac{1}{|F_c^t|}\sum_{g=1}^{G_n^t} F_g^t, \quad \forall c_g = c, \tag{8}$$

The distance between each feature map F_g^t and the prototype feature map $\hat{F}_c^t$ for class c is computed using the Euclidean distance:

$$d(F_g^t, \hat{F}_c^t) = \sqrt{\sum (F_g^t - \hat{F}_c^t)^2}, \quad \forall c_g = c, \tag{9}$$

Then we sort $\{d(F_g^t, \hat{F}_c^t), \forall c_g = c\}_{g=1}^{G_n^t}$ in ascending order, and select the top $M_c = \frac{M}{|\mathcal{C}^{1:t}|}$ boxes for that class to form the box buffer B_c^t. The final B^t can focus on the most relevant information for each task and avoid redundant or irrelevant information, as shown in Algorithm 1.

Additionally, since boxes are typically smaller than whole images, the computational cost of training and rehearsal can be reduced, making the approach more scalable to large datasets and complex models. The entire flow of our proposed method is shown in Algorithm 2.

Algorithm 1 Prototype Box Selection (PBR)

Input: The frozen trained model in $f_{\theta_t}(\cdot)$, the stream data D^t at current task t, each image I_n^t has G_n^t groundtruth labels $\{y_g\}_{g=1}^{G_n^t}$, the box rehearsal memory B^{t-1} after task $t-1$, the box rehearsal memory size M, the seen classes $\mathcal{C}^{1:t}$ until task t.

Output: The updated B^t after task t.

Initialize: $B^t = \{\}$, m^t = ceil$(M/|\mathcal{C}^{1:t}|)$;
$F_g^t = f_{\theta_t}(I_n^t, y_g), \forall n \in N^t, \forall g \in G_n^t$;
$b_g = crop(I_n^t, y_g), \forall n \in N^t, \forall g \in G_n^t$;
for c in $\mathcal{C}^{1:t}$ **do**
  if $c \in \mathcal{C}^t$ **then**
    Compute $\hat{F}_c^t$ for each class c based on Eq. 8;
    $D_c = \{(b_g, y_g) \mid c_g = c\}$;
    Sort D_c following Eq. 9;
    $B^t += D_c[0 : m^t]$;
  else
    for $j = 1, 2, ..., m^t$ **do**
      $i = j * \left|B_c^{t-1}\right| / ceil(M/|\mathcal{C}^{1:t-1}|)$;
      $B^t += B_c^{t-1}[i]$;
    end for
  end if
end for

B. Additional Analysis

B.1. Analysis foreground shift problem

In Table 1 and Table 2, our algorithm demonstrates a remarkable improvement in mean Average Precision (mAP) ranging from 0.2∼20% across all categories. Additionally, it exhibits a substantial mAP boost of 4.5% to 25.2% in new categories (foreground categories), indicating the enhanced stability and plasticity achieved by our method.

Moreover, we conducted a comprehensive analysis of False Positives (FP) [**?**] under the VOC 10-10 setting. Fig. 6 visually represents the number of background errors, specifically detections confused with the background or unlabeled

Algorithm 2 Augmented Box Replay Method

Input: $f_{\theta_{t-1}}(\cdot)$, D^t=$\{I_n^t, G_n^t\}_{n=1}^{N_t}$, B^{t-1} and Rat=1:1:2.
Output: The updated B^t and $f_{\theta_t}(\cdot)$ after task t.

Initialize: $\theta_t = \theta_{t-1}$;
for n in N_t **do**
  MIX,MOS,NEW=GenerateReplayType(Rat);
  if MIX **then**
    Compute $\hat{I}_n^t, \hat{G}_n^t$ by MixupBoxReply(I_n^t, G_n^t);
  else if MOS **then**
    Compute $\hat{I}_n^t, \hat{G}_n^t$ by MosaicBoxReply(I_n^t, G_n^t);
  else if NEW **then**
    $\{\hat{I}_n^t, \hat{G}_n^t\} = \{I_n^t, G_n^t\}$;
  end if
  $\mathcal{L}_{Dis}$ = DistiallationLosses($f_{\theta_{t-1}}(\cdot), f_{\theta_t}(\cdot), \hat{I}_n^t$);
  $\mathcal{L}_{Det}$ = DetectionLosses($f_{\theta_t}(\cdot), \{\hat{I}_n^t, \hat{G}_n^t\}$);
  Update θ_t by $\mathcal{L}_{Dis} + \mathcal{L}_{Det}$;
end for
Update B_t by PBS($f_{\theta_t}(\cdot), D^t, B^{t-1}$);

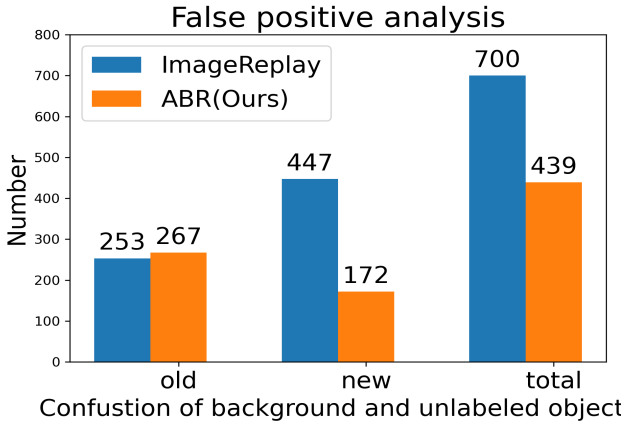


Figure 6: False-Positive Analysis

objects. Notably, our approach (ABR) demonstrates a clear advantage, exhibiting a substantial reduction of 275 errors in new (foreground) classes compared to the ImageReplay method. This compelling result strongly suggests the successful mitigation of the foreground shift problem by our proposed approach.

B.2. Analysis Attentive RoI Distillation (ARD)

While existing methods have utilized attention distillation primarily on feature maps, we advance this approach by integrating location information of Region of Interest (RoI) proposals. By doing so, our model gains the capability to distill both feature and localization information from the replayed and new objects, leading to an overall performance enhancement.

Fig. 7 showcases some additional attention maps, highlighting how our Attention-based RoI Distillation (ARD) loss effectively retains attention on the old class (e.g., bicycle). This observation confirms ARD's competence in alleviating catastrophic forgetting, a phenomenon that impacts model performance when learning new tasks.

(a) Image (b) w/o ARD (c) w ARD

Figure 7: Attention maps during training (person and bicycle are new and old classes respectively).

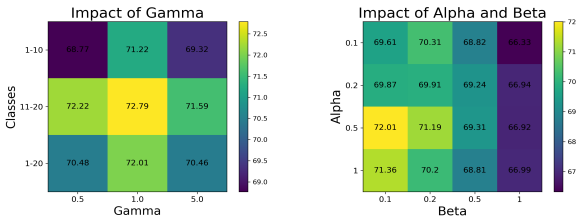


Figure 8: Impact of the hyperparameters γ, α and β.

Through the inclusion of location-awareness in attention distillation, our proposed ARD method exemplifies its potential to mitigate catastrophic forgetting and reinforce the preservation of crucial knowledge from previous tasks, resulting in improved overall model performance.

B.3. Effect of Hyperparameters

We conducted additional experiments under the VOC 10-10 setting to analyze the impact of all hyperparameters in our study, as depicted in Fig. 8. For γ in Eq. 5 of the overall ARD loss function, we vary it in range [0.5, 1.0, 5.0]. From the results shown in the first figure of Fig. 8, we find that the default $\gamma = 1$ provides good results.

In consequence, we optimize the total objective function to realize incremental object detecion learning:

$$\mathcal{L}_{total} = \mathcal{L}_{faster\_rcnn} + \alpha\mathcal{L}_{ID} + \beta\mathcal{L}_{ARD} \quad (10)$$

where α and β weight for the Inclusive Distillation Loss and Attentive RoI Distillation, respectively. We vary it in range [0.1, 0.2, 0.5, 1]. The performance varies as a function of α, β outperforming the state-of-the-art (66.8) for most combinations.

C. Additional Results

C.1. Detailed Results for the Long Sequences

In Table 7, we present the results of our experiments with long sequences on the PASCAL-VOC 2007 dataset. To simulate this scenario, we trained our detector on images from

Table 7: Per-Class AP@50 and Overall mAP@50 values in different task on PASCAL-VOC 2007 5-5 setting.

Class Split	Method	aero	cycle	bird	boat	bottle	bus	car	cat	chair	cow	mAP-task1	table	dog	horse	bike	person	mAP-task2	plant	sheep	sofa	train	tv	mAP-task3	mAP-total
1-20	**JT**	72.7	81.0	76.0	58.9	62.0	76.4	87.4	85.7	72.6	82.4	75.5	57.7	83.2	85.7	80.5	84.2	78.3	45.8	77.1	65.9	75.7	74.5	67.8	74.3
(1-5)+6-10	**MMA**	73.8	80.8	71.2	52.5	63.3	55.2	74.9	65.2	39.1	73.3	64.9													64.9
	ABR	71.7	82.6	69.5	53.6	63.8	63.0	79.0	68.5	47.0	78.4	**67.7**													**67.7**
(1-10)+11-15	**MMA**	67.4	78.1	64.5	49.7	63.5	23.1	34.5	26.3	8.7	35.0	45.1	47.5	52.8	67.5	65.9	76.0	61.9							50.7
	ABR	68.5	79.6	67.3	51.9	56.7	60.2	75.2	62.8	38.6	62.0	**62.3**	54.0	66.3	76.9	74.5	77.3	**69.8**							**64.8**
(1-15)+16-20	**MMA**	72.3	75.5	57.0	46.9	59.9	4.8	32.4	38.5	3.3	1.4	39.2	0.7	28.8	42.2	44.1	18.2	26.8	36.0	46.5	52.0	52.0	66.6	50.6	38.9
	ABR	69.3	80.0	65.6	53.9	54.6	52.2	75.5	69.4	34.3	69.6	**62.4**	22.9	41.8	48.7	53.7	60.8	**45.6**	39.6	71.3	59.2	76.1	70.4	**63.3**	**58.4**

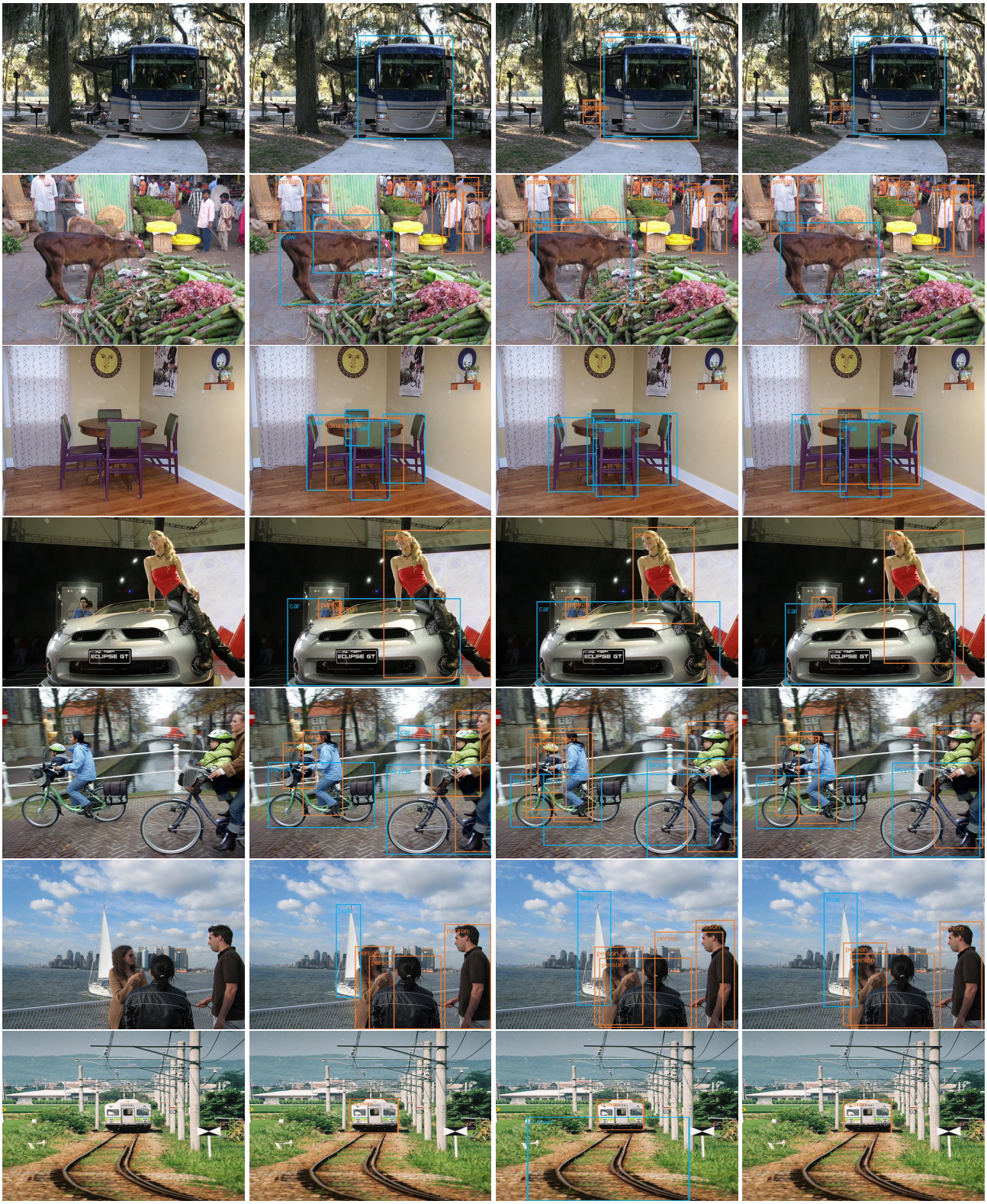
the first 5 classes and gradually added classes 6 to 20 in groups of five.

The table shows the class-wise average precision (AP)@0.5 and the corresponding mean average precision (mAP). The first row (JT) represents the upper-bound where the detector is trained on data from all 20 classes. The subsequent three pairs of rows demonstrate the results obtained when adding five new classes at a time. The notation (1-5)+6..10 is used to represent this setting. Our proposed ABR method outperforms the previous state-of-the-art method MMA [**?**] on all sequential tasks, as can be seen from the results in Table 7. Therefore, the ABR method can be more useful in real-world scenarios where new object classes are frequently introduced. Additionally, the ABR method is a novel approach that may have implications for future research in object detection.

C.2. Visualization

The inference results are presented in Fig. 9, which demonstrate the effectiveness of our proposed ABR method in avoiding the forgetting of previous classes and improving adaptation to new classes. In the first two rows, our method is capable of accurately distinguishing new classes from similar classes in the previous classes, as seen in the detection of a *bus* in the first row of images and a *cow* in the second row of images. However, the popular MMA method misclassifies the *bus* as a *train* or *bus* and the *cow* as a *dog* or *cow*. In the third row, our algorithm successfully detects the new class, a *dining table*, while also accurately locating a previous class, a *chair*. In comparison to the MMA method, our method achieves more precise position detection, as demonstrated in the last two rows where *person* and *boat* are detected.

Overall, these results suggest that the proposed ABR method can more effectively handle the problem of incremental learning in object detection tasks, particularly in scenarios where new classes are similar to previous ones. The ability to avoid forgetting and adapt to new classes is crucial for practical applications, and the improved performance of our method is promising for future research in this area.



(a) Image (b) GT (c) MMA (d) Ours

Figure 9: Visualization of the inference results in MMA and Ours for 8 test images on PASCAL-VOC 2007 10-10 scenario.