Suppementary

This appendix reports the details of the experiment setting and implementation and provides additional ablation studies and qualitative results.

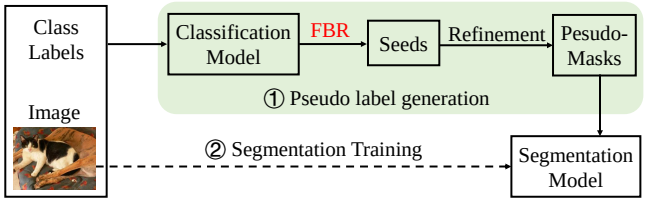

Fig. 1: Overall training pipeline. Our contribution (FBR method) is to improve the classification model to enhance CAMs and generate more precise seed predictions.

A. Experiment details

Overall training pipeline. As shown in Fig. 1, we first employ our FBR method on the classification model to enhance the seed predictions by addressing the co-occurring background problem and learning integral object regions. After refining [1] the seeds to obtain the pseudo masks, we utilize them to train the segmentation model and get the final results.

Seed generation. We follow the original configurations reported in PPC [10], SEAM [43], and AMN [25] for our baseline experiments. Based on the ablation results in Tab. XIII, we set the loss weights of the background pixel-wise cross-entropy loss (L_{pcl}^{bg}), foreground pixel-wise cross-entropy loss (L_{pcl}^{fg}), and segmentation loss (L_{seg}) to 0.1, 0.01, and 0.01, respectively, for PPC and SEAM. For AMN, we set these loss weights to 0.1, 0.05, and 0.01. When applying our FBR to PPC and SEAM, we employ a learning rate (lr) of 0.01 and a batch size of 9, optimized using the PolyOptimizer. For experiments on AMN, we adopt a lr of 5e-6, a batch size of 1, and utilize the Adam optimizer. Furthermore, when experimenting on the MS COCO and Pascal VOC datasets, we set the feature bank size to 200 K and 50 K, respectively.

Semantic segmentation. For the segmentation practice, we strictly follow the settings of the baselines. Training images are randomly scaled in the range of [0.5, 0.75, 1.0, 1.25, 1.5] and cropped to 321× 321 for Pascal Voc 2012, 481× 481 for MS COCO 2014. We adopted the SGD optimizer (lr=0.01) and set the batch size to 10 (16 for MS COCO 2014). The number of training steps is $30K$ ($100K$ for COCO).

Instance segmentation. We follow the training settings of BoxInst [38], except resizing the shorter side of images in [480,640]. In training, we set the batch size to 16 and use the SGD optimizer (with learning rate = 0.01).

B. Additional ablation analysis

Computational overhead & Complexity. Utilizing PPC [10] as the baseline model, we conduct a comprehensive analysis to evaluate the computational efficiency of our FBR approach.

Based on PPC, FBR slightly increases the training hours (4.8 h → 6.4 h) and model size (102.9 M → 103.9 M). The increased computation cost comes from the added background projection head φ_{bg} and the classifiers φ_{seg}. Notably, our method does not affect the inference speed of the baseline since all involved modules are removed at the test time.

TABLE I: We compare the computational cost of PPC [10] and our method regarding the parameter size (million, M), training time (hours, h), inference speed (second / per image), and GPU memory footprint (GB).

Method	Param.	Train. time	Infer. speed	Memory
PPC [10]	102.9 M	4.8 h	1.86 s	14.9 GB
GrayOurs	103.9 M	6.4 h	1.86 s	16.5 GB

Auxiliary BG segmentation. To obtain effective background negative samples, we define a learning objective for Z_{bg}, i.e., the background segmentation, requiring Z_{bg} to discriminate the background region and learn more representative features.

We consider pixels with a summed CAM score (on foreground classes) smaller than 0.05 as the background to conduct the auxiliary segmentation training. We add a loss weight α before the background segmentation loss L_{seg} in Eq. 10, and ablate α (test on Pascal Voc 2012 train set) in Tab. II. We observe that our method performs best when setting $\alpha = 0.01$.

TABLE II: Additional ablation experiments of the background segmentation. α is the loss weight of L_{seg} in Eq. 10.

α	0	0.01	0.025	0.05
mIoU	75.2	**75.5**	74.9	74.1

Effects of TAP. Proposed in [4], TAP only aggregates above-threshold pixels in the semantic feature f to compute the classification score. We express TAP as follows:

$$s_c^{tap} = \sum_{i=1}^{L} \theta_{c,i} \frac{\sum_{j\in\Omega} 1[f_{i,j} > \alpha] f_{i,j}}{\sum_{j\in\Omega} 1[f_{i,j} > \alpha]}, \quad (1)$$

where α is the threshold (set to 0.1) and Ω is the coordinate set of $\mathbb{R}^{H\times W}$. TAP has been proven to be better at filtering out over-activated BG regions [4] than GAP, thus generating more accurate seeds. In our work, TAP brings more accuracy improvements than GAP (74.7% $v.s$ 75.0%) when assembling our FBR method on PPC [10].

C. Additional Experimental Results

Category-level Evaluation Tab. III shows category-level mIoU evaluation in terms of the pseudo masks. We observe that FBR significantly enhances baseline models' ability to distinguish classes that have complicated shapes, like "bird", "dog", and "dining table (dt.)." Also, we report the segmentation results in Tab. IV after training the segmentation network

TABLE III: (**Pseudo Masks**) Category-level mIoU comparison on Pascal Voc 2012 train set. Highlights are classes in which we achieve up to 2.5% mIoU improvements.

Method	mIoU	bgr	aero	bicy	bird	boat	bottle	bus	car	cat	chair	cow	dt.	dog	horse	motor	pers.	pott	sheep	sofa	train	tv.
AMN	72.2	90.2	75.3	40.1	77.4	67.9	73.4	85.6	78.9	80.7	36.5	86.1	65.8	78.7	83.4	81.0	74.4	62.4	89.4	62.8	65.3	63.1
+ours	**73.1**	90.8	74.4	**45.1**	**84.9**	69.9	71.7	84.4	79.3	**87.1**	37.2	85.6	61.6	**84.5**	81.0	79.2	73.8	63.9	89.8	63.5	66.3	60.2
PPC	73.3	91.2	86.6	44.6	82.8	80.9	73.1	84.0	81.4	88.9	31.2	83.7	52.7	85.6	86.9	81.7	80.5	54.2	85.9	52.5	77.6	53.8
+ours	**75.9**	92.4	86.4	**47.7**	**85.0**	83.1	75.1	85.0	**85.5**	**91.6**	39.7	**88.3**	50.6	**91.6**	**90.7**	83.6	81.1	**63.2**	**90.0**	48.5	**83.9**	51.7

TABLE IV: (**Semantic Segmentation**) Category-level mIoU comparison on Pascal Voc 2012 test set. Highlights are classes in which we achieve up to 3.0% mIoU improvements. The reported AMN performance is from the ImageNet pre-trained model.

Method	mIoU	bgr	aero	bicy	bird	boat	bottle	bus	car	cat	chair	cow	dt.	dog	horse	motor	pers.	pott	sheep	sofa	train	tv.
AMN	69.6	90.7	82.8	32.4	84.8	59.4	70.0	86.7	83.0	86.9	30.1	79.2	56.6	83.0	81.9	78.3	72.7	52.9	81.4	59.8	53.1	56.4
+ours	**73.2**	91.3	82.9	**35.3**	**90.5**	59.1	70.1	**90.1**	84.0	**91.2**	**36.5**	**85.9**	**66.3**	**88.8**	**87.3**	79.1	**77.2**	**63.6**	**86.1**	59.7	53.4	58.4
PPC	73.6	92.1	92.3	40.6	89.8	65.4	69.9	91.5	83.6	90.9	31.4	86.2	48.2	85.1	89.8	81.9	80.2	59.6	87.7	52.9	80.3	46.4
+ours	**74.9**	92.2	92.6	40.7	86.4	63.6	70.1	92.1	84.2	91.3	**36.1**	**88.2**	**51.6**	**89.3**	89.9	83.2	78.1	**72.7**	90.1	55.1	79.3	45.3

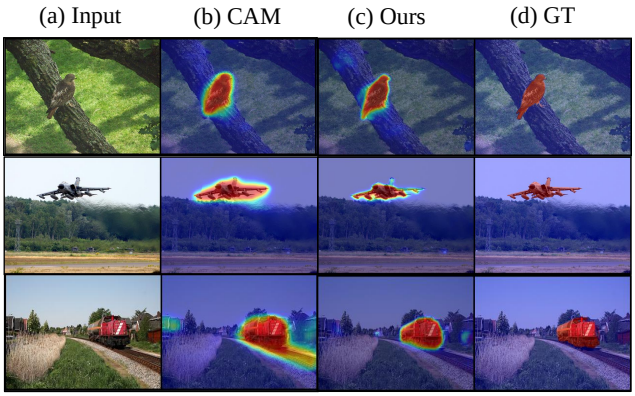

Fig. 2: Example CAM results (solve co-occurring BG). Left to Right: (a) input image, (b) CAM results generated by AMN, (c) CAM results from AMN W/ ours, and (d) the ground truth.

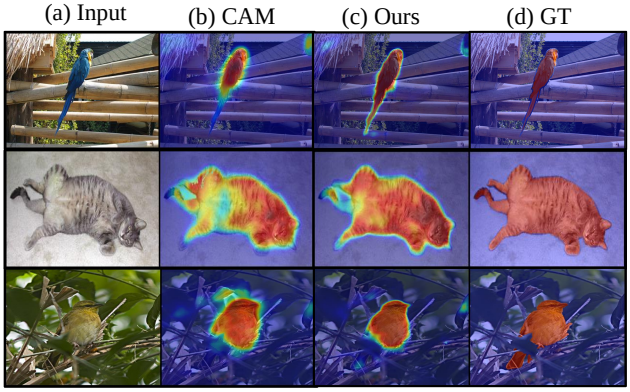

Fig. 3: Example results (activating more object regions).

with the pseudo masks and observe considerable performance improvements on the baselines.

Quantitative results Fig. 2 and Fig. 4 provide more CAM results to show FBR's effectiveness. In Fig. 2, we see that FBR effectively helps the baselines distinguish the target FG object from the confusing BG semantics. In Fig. 4, we notice that the baseline (with FBR) learns more class features, thus activating more integral object regions.

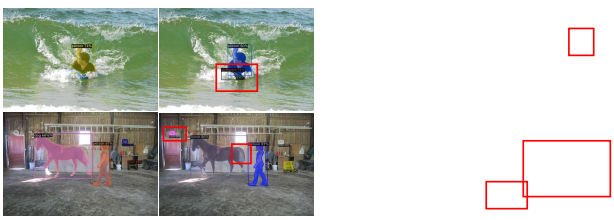

Fig. 4: Example results of instance segmentation.