Appendix of “Out-of-Candidate Rectification for Weakly Supervised Semantic Segmentation”

Zesen Cheng[1,2*] Pengcheng Qiao[1,2,3*] Kehan Li[1,2] Siheng Li[4] Pengxu Wei[5]
Xiangyang Ji[4] Li Yuan[1,3] Chang Liu[4 ✉] Jie Chen[1,2,3 ✉]

[1] School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2] AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School, China
[3] Peng Cheng Laboratory, Shenzhen, China [4] Tsinghua University [5] Sun Yat-Sen University

{cyanlaser,pcqiao,kehanli}@stu.pku.edu.cn weipx3@mail.sysu.edu.cn
{xyji,liuchang2022}@tsinghua.edu.cn {yuanli-ece,jiechen2019}@pku.edu.cn

https://github.com/sennnnn/Out-of-Candidate-Rectification

In this document, we first discuss the limitations of our method (§Sec. 1). Then we provide qualitative analysis of different pixel selection strategies (§Sec. 2), which is a supplement to quantitative analysis of pixel selection strategies in main paper. Thirdly, the detailed prior correlation information is illustrated and described (§Sec. 3). Finally, pseudo-codes of OCR (§Sec. 4) and extra qualitative results (§Sec. 5) are provided.

S1. Discussion

In this section, we discuss the limitations of our Out-of-Candidate Rectification (OCR). The limitations can be divided into two folds: Firstly, although the OCR is proposed for Out-of-Candidate (OC) phenomenon, the OCR can’t drastically solve the OC problem. OC phenomenon is essentially the intrinsic weakness of weakly supervised semantic segmentation and this problem may be intractable under current CAM-based technology. The second fold is that the proposed OCR requires a strong hypothesis, i.e., “*anchor class which has maximal prediction score is the ground truth class*”. However, the hypothesis does not always hold so it is not always possible to rectify OC pixels.

S2. Qualitative Analysis of Pixel Selection Strategy

In Fig. S1, we provide the spatial visualization of OCR spatial loss map by two types of pixel selection strategies (“ALL” and “Only OC”). By comparing the first column and second column of Fig. S1, we can easily identify those **Pink** pixels as OC pixels. These OC pixels whose ground truth label is “***cow***” are misled to “***horse***”. The third column and fourth column of Fig. S1 are the loss spatial map of OCR with “ALL” or “Only OC” pixel selection strategies, respectively. “ALL” strategy assign different intensity of supervision signals to all of the pixels, which means IC pixels are also forced to attend to the rectification of OCR. Rectification loss of OCR is only used to rectify OC pixels from OC group to IC group. However, IC pixels already belong to IC group so the rectification loss of OCR calculated on IC pixels is not useful for improving results. “Only OC” strategy only assigns supervision signals to OC pixels, which is reasonable and is in line with the motivation of proposed OCR (correcting OC pixels into IC group).

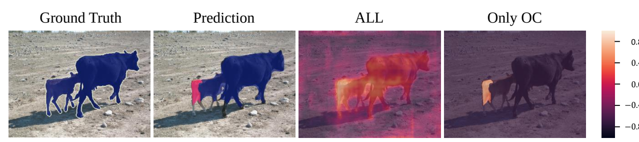
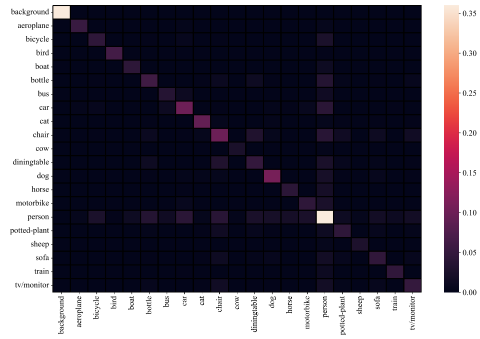


Figure S1. **Loss spatial map visualization of OCR with different pixel selection strategy.** “ALL” denotes we select all of the pixels to attend the calculation of OCR loss. “Only OC” means only the OC pixels are selected to attend the calculation of OCR loss. “ALL” pixel selection strategy provides unnecessary supervision signals to IC pixels, which causes suboptimal performance. “Only OC” strategy is highly selective to OC pixels, which can efficiently suppress OC error phenomenon for better performance. **Blue** denotes “***cow***” and **Pink** denotes “***horse***”.

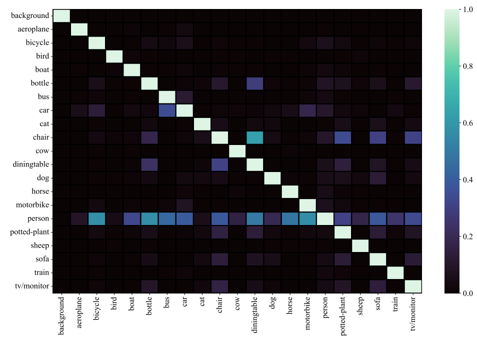
S3. Prior Correlation Information

As mentioned in the description of adaptive group split strategy in Out-Of-Candidate Rectification (OCR) Section, we adopt prior and posterior correlation information from co-occurrence of tag labels and network predictions for filtering useless classes in IC group. In Fig. S2a, we illustrate the prior correlation matrix of Pascal VOC *train* split. For

* Equal contribution ✉ Corresponding Author



(a) Prior Correlation



(b) Normalized Prior Correlation

Figure S2. **Visualization of prior correlation matrix.** The original prior correlation matrix calculated by co-occurrence between different classes is shown in (a). For clearly showing the correlation between different classes, we conduct min-max normalization on prior correlation matrix along the horizontal dimension, which is shown in (b).

clearly showing the correlation between different classes, we extra provide normalized prior correlation matrix in Fig. S2b. There are some interesting correlation relationship between different classes, e.g., *bus* + *car*, *chair* + *diningtable* and *bottle* + *dingingtable*. The most two special classes are *person* and *background*. The former nearly is highly correlated to nearly all of other classes (except *background*). For latter, *background* is contained in any images of dataset so it has no correlation to other classes.

S4. Pseudo Code

In this section, we provide the pseudo-codes of OCR for better grasping our method. We first provide general workflow of OCR in Algo. 1. OCR comprises three components: Then we describe the details of adaptive IC/OC group split strategy in Algo. 2.

S5. Extra Qualitative Results

In main paper, we already provide comparison qualitative results of baseline and our methods for showing the correct effect of OC phenomenon. In this section, some regular cases of segmentation results are shown in Fig. S3.

Algorithm 1 PyTorch-style Pseudo-code of Out-of-Candidate Rectification (OCR)

```
# im_gt: image-level tag labels [C]
# logits: network output logits [(C + 1),H,W]
# delta: the margin value

# Out-of-Candidate Rectification Loss
def ocr_loss(logits, im_gt):
   # OC pixels selection mask
   clean_logits = logits.detach().clone()
   ## broadcast operation
   clean_logits[1:] = clean_logits[1:] * im_gt
   pred = torch.argmax(logits, dim=0)
   clean_pred = torch.argmax(clean_logits, dim=0)
   ## bool index [H,W]
   oc_mask = (clean_pred != pred)

   # Adaptive Group Split
   ic_gp = ic_gp_build(logits, im_gt)
   oc_gp = oc_gp_build(logits, im_gt)

   # Rectification Loss
   se_ic = sum(exp(-ic_gp), dim=0)
   se_oc = sum(exp(oc_gp + delta), dim=0)
   l_rec = (oc_mask * log(1 + se_ic * se_oc)).mean()

   return l_rec
```

Notes: `sum`(·), `exp`(·) and `log`(·) are the functions of pytorch. `[1:]` denotes slice operation which removes the zero element. `ic_gp_build`(·) and `oc_gp_build`(·) are introduced in algo. 2.

Algorithm 2 PyTorch-like Pseudo-code of adaptive IC/OC group split.

```
# im_gt: image-level tag labels [C]
# logit: single pixel network output [(C + 1),H,W]
# M: prior correlation matrix [(C + 1),(C + 1)]
# tau: IC group useless classes filter threshold

def get_ic_ids(im_gt):
   # acquire initial IC class ids (tag labels and
       background)
   return cat([0, nonzero(im_gt) + 1])

def get_oc_ids(im_gt):
   # acquire OC class ids
   return nonzero(1 - im_gt) + 1

# single pixel behavior
def ic_gp_build(logit, im_gt):
   ic_cls_ids = get_ic_ids(im_gt)
   # network output probability
   prob = softmax(logit)
   # 1. Extract initial IC classes by tag labels.
   ic_gp_prob = prob[ic_class_ids]
   ic_filter_mask = zeros(len(ic_class_ids))
   # 2. Anchor Predictions and Classes
   anchor_prob, anchor_id = max(ic_gp_prob)
   ic_filter_mask[anchor_id] = 1
   # 3. utilize correlation for filtering
   cond = (anchor_prob - prob * M[anchor_id]) < tau
   ic_filter_mask[cond] = 1

   return (ic_filter_mask * logit)[ic_cls_ids]

# single pixel behavior
def oc_group_build(logit, im_gt):
   oc_cls_ids = get_oc_ids(im_gt)
   return logit[oc_cls_ids]
```

Notes: `cat`(·) is the concatenate operation. `max`(·) returns the max value and related indices of tensor. `nonzero`(·) is used to acquire the indices of non-zero elements. `zeros`(·) is used to create a special shape of tensor filled with 0.

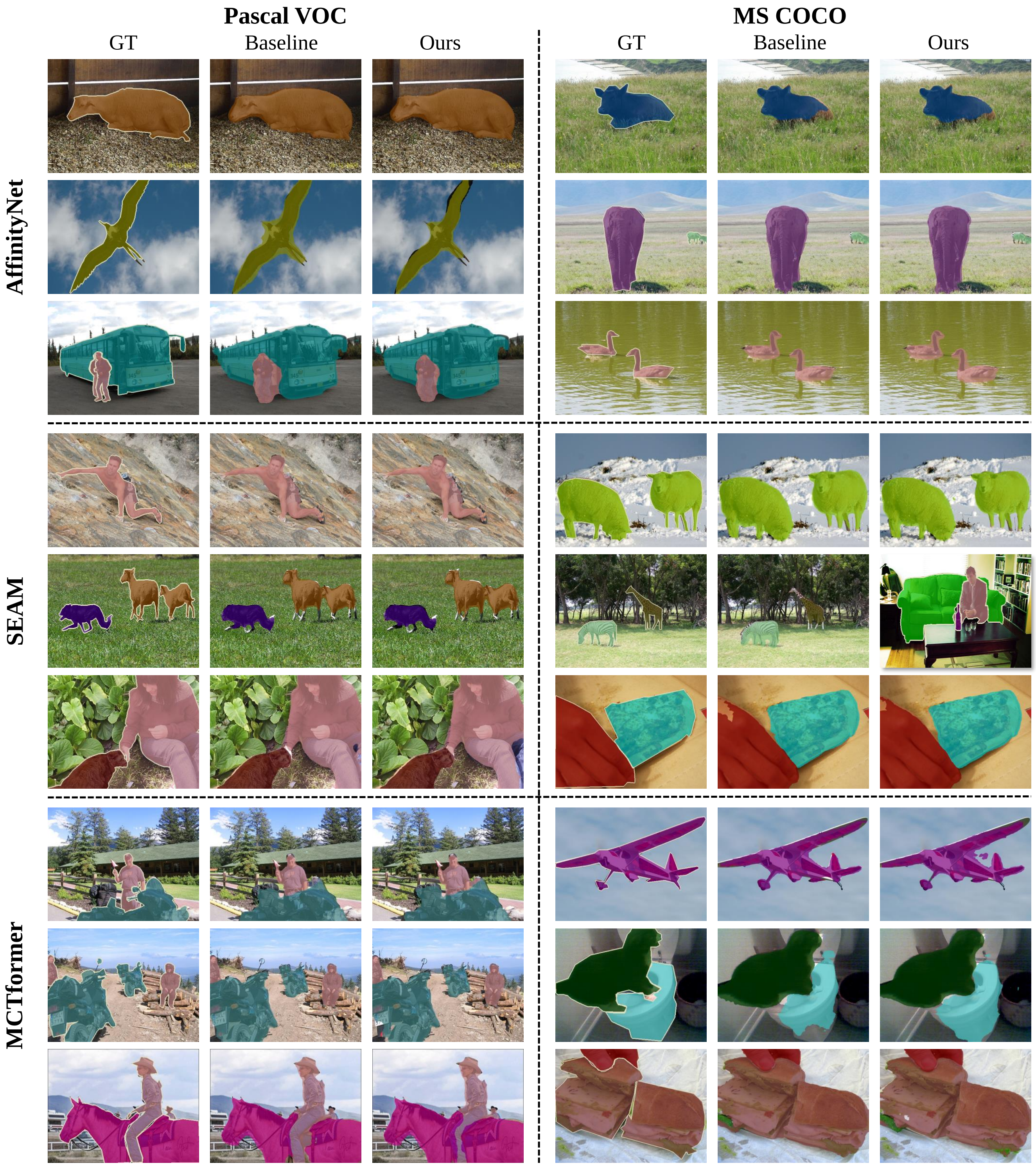


Figure S3. **Extra Qualitative Results.** We extra provide some normal cases of segmentation results for showing regular segmentation performance.