Weakly Supervised Co-training with Swapping Assignments for Semantic Segmentation

Xinyu Yang[1], Hossein Rahmani[1], Sue Black[2], and Bryan M. Williams[1]

[1] Lancaster University, Bailrigg Lancaster, LA1 4YW, UK
[2] University of Oxford, Wellington Square Oxford, OX1 2JD, UK
{xyang28,h.rahmani,b.williams6}@lancaster.ac.uk, sue.black@sjc.ox.ac.uk

Abstract. Class activation maps (CAMs) are commonly employed in weakly supervised semantic segmentation (WSSS) to produce pseudo-labels. Due to incomplete or excessive class activation, existing studies often resort to offline CAM refinement, introducing additional stages or proposing offline modules. This can cause optimization difficulties for single-stage methods and limit generalizability. In this study, we aim to reduce the observed CAM inconsistency and error to mitigate reliance on refinement processes. We propose an end-to-end WSSS model incorporating guided CAMs, wherein our segmentation model is trained while concurrently optimizing CAMs online. Our method, Co-training with Swapping Assignments (CoSA), leverages a dual-stream framework, where one sub-network learns from the swapped assignments generated by the other. We introduce three techniques in this framework: i) soft perplexity-based regularization to penalize uncertain regions; ii) a threshold-searching approach to dynamically revise the confidence threshold; and iii) contrastive separation to address the coexistence problem. CoSA demonstrates exceptional performance, achieving mIoU of 76.2% and 51.0% on VOC and COCO validation datasets, respectively, surpassing existing baselines by a substantial margin. Notably, CoSA is the first single-stage approach to outperform all existing multi-stage methods including those with additional supervision. Source code is publicly available at here.

Keywords: Weakly-supervised Learning · Semantic Segmentation · CAM

1 Introduction

The objective of weakly supervised semantic segmentation (WSSS) is to train a segmentation model without relying on pixel-level labels but on weak and cost-effective annotations, such as image-level classification labels [3, 26, 48], object points [4,43], and bounding boxes [16,28,34,55]. In particular, image-level classification labels have commonly been employed as weak labels due to the minimal or negligible annotation effort required [2,58]. With the absence of precise localization information, image-level WSSS often makes use of the coarse localization offered by class activation maps (CAMs) [72]. CAMs pertain to the intermediate outputs derived from a classification network. They visually illustrate the activation regions corresponding to each individual class. Thus, they are often used to generate pseudo masks for training. However, CAMs suffer from i) Inconsistent Activation: CAMs demonstrate variability and lack robustness in accommodating geometric transformations of input images [58], resulting in inconsistent activation regions for the same input. ii) Inaccurate Activation: activation region accuracy is often compromised, resulting in incomplete or excessive class activation, only covering discriminative object regions [1]. Despite enhanced localization mechanisms in the variants GradCAM [53] and GradCAM^{++} [7], they still struggle to generate satisfactory pseudo-labels for WSSS [58]. Thus, many WSSS works are dedicated to studying CAM refinement or post-processing [1, 15, 31].

In general, they [2, 18, 48, 62] comprise three stages: CAM generation, refinement, and segmentation training with pseudo-labels. Multi-stage frameworks are known to be time-consuming and complex as several models must be trained at different stages. In contrast, single-stage models [3, 50, 70], which include a unified network of all stages, are more efficient. They are trained to co-optimize the segmentation and classification tasks, but the generated CAMs are not explicitly trained. As a result, they need refinement to produce high-quality pseudo-labels, often leveraging hand-crafted modules, such as CRF in [70], PAMR in [3], PAR in [50, 51]. As the refinement modules are predefined and offline, they decouple the CAMs from the primary optimization. When the refined CAMs are employed as segmentation learning objectives, the optimization of the segmentation branch may deviate from that of the classification branch. Hence, it is difficult for single-stage models to optimize the segmentation task while yielding satisfactory CAM pseudo-labels. This optimization difficulty underlies the inferior performance in single-stage approaches compared to multi-stage [48, 62]. Further, hand-crafted refinement modules require heuristic tuning and empirical changes, thereby limiting their adaptability to novel datasets [3, 50]. Despite the potential benefits of post-refinement in addressing the aforementioned issues associated with CAMs, which have been extensively discussed in WSSS studies, there has been limited exploration of explicit online optimization for CAMs.

The absence of fully optimized CAMs is an important factor in the indispensability of this refinement. In this paper, we take a different approach by optimizing CAMs in an end-to-end fashion. We ask a core question: Can we train a model that delivers reliable, consistent and accurate CAMs, which can be applied directly for WSSS without the necessity for subsequent refinements? We show that the answer is *yes*, in two respects: 1) we note that even though CAM is differentiable, it is not robust to variation. As the intermediate output of classification, CAMs are not fully optimized for segmentation purposes since the primary objective is to minimize classification error. This implies that within an optimized network, numerous weight combinations exist that can yield accurate classification outcomes, while generating CAMs of varying qualities. To investigate this, we conduct *oracle experiments*, training a classification model while simultaneously guiding the CAMs with the segmentation ground truth. A noticeable enhancement in quality is observed in guided compared to vanilla CAMs, without compromising classification accuracy. 2) we demonstrate the feasibility of substituting the oracle with segmentation pseudo-labels (SPL) in the context of weak supervision. Consequently, we harness the potential of SPL for WSSS by co-training both CAMs and segmentation through mutual learning.

We explore an effective way to substitute the CAM refinement process, *i.e.* guiding CAMs in an end-to-end fashion. Our method optimizes the CAMs and segmentation prediction simultaneously thanks to the differentiability of CAMs. To achieve this, we adopt a dual-stream framework that includes an online network (ON) and an assignment network (AN), inspired by self-training frameworks [5,22,65]. The AN is responsible for producing CAM pseudo-labels (CPL) and segmentation pseudo-labels (SPL) to train the ON. Since CPL and SPL are swapped for supervising segmentation and CAMs, respectively, our method is named **Co**-training with **S**wapping **A**ssignments (CoSA).

The benefit of this end-to-end framework is that it enables us to quantify pseudo-label reliability online, as opposed to the offline hard pseudo-labels used in existing methodologies [2, 15, 48, 62]. We can then incorporate soft regularization to compensate for CPL uncertainty, where the segmentation loss for different regions is adaptively weighted according to our estimated perplexity map. In comparison to existing literature, this dynamic learning scheme can exploit the potential of CPL and enhance the final performance, as opposed to performance being constrained by predetermined CPL. The threshold is a key hyper-parameter for generating the CPL [48, 51, 58]. It not only requires tuning but necessitates dynamic adjustment to align with the model's learning state at various time-steps. CoSA integrates threshold searching to dynamically adapt its learning state, as opposed to the fixed thresholding [12, 18, 50]. This can enhance performance and help to eliminate the laborious manual parameter-tuning task. We further address a common issue with CAMs, known as the coexistence problem, whereby certain class activations display extensive false positive regions that inaccurately merge the objects with their surroundings (Fig. 4). In response, we introduce a technique to leverage low-level CAMs enriched with object-specific details to contrastively separate those coexistent classes.

The proposed CoSA greatly surpasses existing WSSS methods. Our approach achieves the leading results on VOC and COCO benchmarks, highlighting the contribution of this work: **i)** We are the first to propose SPL as a substitute for guiding CAMs. We present compelling evidence of its potential to produce more reliable, consistent and accurate CAMs. **ii)** We introduce a dual-stream framework with swapped assignments, which co-optimizes the CAMs and segmentation predictions in an end-to-end fashion. **iii)** We address the learning dynamics, proposing two components within our framework: reliability-based adaptive weighting and dynamic thresholding. **iv)** We address the CAM coexistence issue, proposing a contrastive separation approach to regularize CAMs, significantly enhancing the results of affected classes.

2 Related Work

Multi-Stage WSSS. Most image-level WSSS work is multi-stage, typically comprising three stages: CAM generation, CAM refinement, and segmentation training. Some approaches employ heuristic strategies to address incomplete activation regions, such as adversarial erasing [30, 56, 67, 71], feature map optimization [12–14, 32], self-supervised learning [11, 54, 58], and contrastive learning [15,27,61,73]. Some methods focus on post-refining the CAMs by propagating object regions from the seeds to their semantically similar pixels. AffinityNet [2], for instance, learns pixel-level affinity to enhance CPL. This has motivated other work [1, 10, 20, 36] that utilize additional networks to generate more accurate CPL. Other work studies optimization given coarse pseudo-labels: [38] explores uncertainty of noisy labels, [41] adaptively corrects CPL during early learning, and [48] enhances boundary prediction through co-training. Since image-level labels alone do not yield satisfactory results, several methods incorporate additional modalities, such as saliency maps [18,35,36,73] or CLIP models [40,60,64]. Recently, vision transformers [17] have emerged as prominent models for various vision tasks. Several WSSS studies benefit from vision transformers: [21] enhances CAMs by incorporating the attention map from ViT; [62] introduces class-specific attention for discriminative object localization; [40] and [64] leverage multi-modal transformers to enhance performance.

Single-Stage WSSS. In contrast, single-stage methods are much more efficient. They contain a shared backbone with heads for classification and segmentation [3, 50, 51, 70]. The pipeline involves generating and refining the CAMs, leveraging an offline module, such as PAMR [3], PAR [50], or CRF [70]. Subsequently, the refined CPL are used for segmentation. Single-stage methods exhibit faster speed and a lower memory footprint but are challenging to optimize due to the obfuscation in offline refinement. As a result, they often yield inferior performance compared to multi-stage methods. More recently, with the success of ViT, single-stage WSSS has been greatly advanced. AFA [50] proposes learning reliable affinity from attention to refine the CAMs. Similarly, ToCo [51] mitigates the problem of over-smoothing in vision transformers by contrastively learning from patch tokens and class tokens. The existing works depend heavily on offline refinement of CAMs. In this study, we further explore the potential of single-stage approaches and showcase the redundancy of offline refinement. We propose an effective alternative for generating consistent, and accurate CAMs in WSSS.

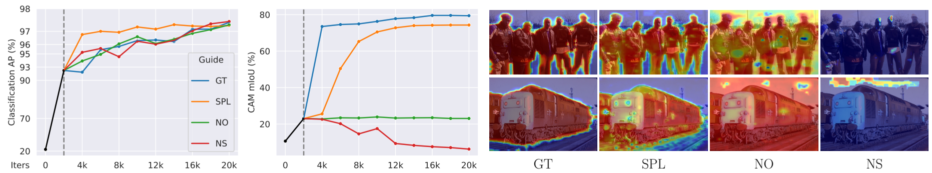
3 Method

3.1 Guiding Class Activation Maps

Class activation maps are determined by the feature map F and the weights W_{fc} for the last `FC` layer [72]. Let us consider a C classes classification problem:

$$\mathcal{L}_{\text{cls}}(Z,Y) = \frac{-1}{C}\sum_{c=1}^{C}\Big[Y^c \log \sigma_Z^c + (1-Y^c)\log(1-\sigma_Z^c)\Big], \tag{1}$$

where $\sigma_Z^c \triangleq \sigma(Z^c)$ represents `Sigmoid` activation, $Y \triangleq Y_{\text{gt}}$ denotes the one-hot multi-class label, and $Z \triangleq GW_{\text{fc}}^{\top} \in \mathbb{R}^C$ represents the prediction logits, derived



(a) (b) (c)

Fig. 1: Oracle Experiments on VOC. CAMs are guided by the ground truth (GT), proposed segmentation pseudo-labels (SPL), no guidance (NO) and random noise (NS). **(a)**: classification performance; **(b)**: CAM quality; **(c)** CAM visualization. All experiments employ 2k-iters warm-up before guidance is introduced.

from the final `FC` layer, where $G = \texttt{Pooling}(F) \in \mathbb{R}^D$ is a spatial pooled feature from $F \in \mathbb{R}^{HW \times D}$. During training, Eq. (1) is optimized with respect to the learnable parameters θ in the backbone. When gradients flow backwards from G to F, only a fraction of elements in F get optimized, implying that a perturbation in F does not guarantee corresponding response in G due to the spatial pooling, resulting in non-determinism in the feature map F. This indeterminate nature can lead to stochasticity of the generated CAMs.

To demonstrate, we conduct oracle experiments wherein we supervise the output CAMs from a classifier with the ground truth segmentation (GT), enabling optimization of all elements in F. For comparison, we conduct experiments where the CAMs are not guided (NO), and guided with random noise masks (NS). Results, shown in Fig. 1, demonstrate that different guidance for M does not affect classification even for the NS group, as all experiment groups achieved over 97% classification precision. However, drastic differences can be observed *w.r.t.* the quality of the CAMs. The GT group results in a notable quality improvement over the NO group, as shown in Fig. 1(b)(c). In contrast, the NS group sabotages the CAMs. This suggests the stochasticity of CAMs and explains their variability and lack robustness, something also observed in [2, 12, 58].

Since relying on GT segmentation is not feasible in WSSS, we propose an alternative for guiding CAMs, employing predicted masks as segmentation pseudo-labels (SPL). As shown in Fig. 1, a SPL-guided classifier yields CAMs that significantly outperform vanilla CAMs (NO), performing close to the oracle (GT). Motivated by this, we introduce a co-training mechanism in which CAMs and predicted masks are optimized mutually without additional CAM refinement.

3.2 Co-training with Swapping Assignments

Overall Framework. As shown in Fig. 2, CoSA contains two networks: an *online* network (ON) and an *assignment* network (AN). ON, parameterized by Θ, comprises three parts: a backbone encoder, `FC` layers, and a segmentation head. AN has the same architecture as ON but uses different weights, denoted Θ'. ON is trained with the pseudo assignments generated by AN, while AN is updated by the exponential moving average of ON: $\Theta' \leftarrow m\Theta' + (1-m)\Theta$, where $m \in [0,1]$ denotes a momentum coefficient. Consequently, the weights of

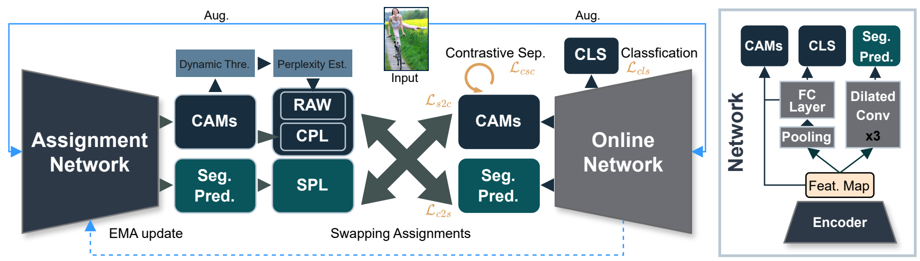


Fig. 2: Co-training with Swapping Assignments (CoSA). We propose an end-to-end dual-stream weakly-supervised segmentation framework, capable of co-optimizing the segmentation prediction and CAMs by leveraging the swapped assignments, namely CAM pseudo-labels (CPL) and segmentation pseudo-labels (SPL). Our framework comprises two networks: an assignment network (AN) and an online network (ON), where the AN is responsible for generating pseudo-labels for training the ON. While the AN has identical architecture to the ON, it is updated through exponential moving average (EMA) of the ON. The diagram on the right provides an illustration of the architecture. Given weak-augmented images as input, the AN produces CPL to supervise segmentation in the ON ($\mathcal{L}_{\text{c2s}}$). During training, the CPL is softened by reliability-based adaptive weighting (RAW), formed based on CAM perplexity estimation and dynamic thresholding. The AN also generates SPL which is utilized to supervise the CAMs ($\mathcal{L}_{\text{s2c}}$). Further, the CAMs are regularized to contrastively separate the foreground from the background regions ($\mathcal{L}_{\text{csc}}$). Note that the ON is also trained for classification using the image-level class labels ($\mathcal{L}_{\text{cls}}$).

AN represent a delayed and more stable version of the weights of ON, which helps to yield a consistent and stabilized learning target [22].

An image and class label pair (x, Y_{gt}) is randomly sampled from a WSSS dataset $\mathcal{D}$. CoSA utilizes two augmented views $\mathcal{T}_s(x)$ and $\mathcal{T}_w(x)$ as input for ON and AN, respectively, representing strong and weak image transformations. During training, AN produces CAMs $\mathcal{M}'$ and segmentation predictions $\mathcal{S}'$. The CAM pseudo-labels (CPL) and segmentation pseudo-labels (SPL) are generated by $\mathcal{M}'$ and $\mathcal{S}'$ after filtering with respect to Y_{gt}. CPL and SPL are subsequently used as learning targets for supervising the segmentation predictions $\mathcal{S}$ and CAMs $\mathcal{M}$ from ON, respectively.

Swapping Assignments. Our objective is to co-optimize $\mathcal{S}$ and $\mathcal{M}$. A naive approach could enforce the learning objectives $\mathcal{S} \triangleq \mathcal{S}'$ and $\mathcal{M} \triangleq \mathcal{M}'$ as a knowledge distillation process [25], where AN and ON play the roles of teacher and student. However, this assumes availability of a pretrained teacher which is not possible in WSSS settings. Instead, we setup a swapped self-distillation objective:

$$\mathcal{L}_{\text{swap}} = \mathcal{L}_{\text{c2s}}(\mathcal{S}, \mathcal{M}') + \mathcal{L}_{\text{s2c}}(\mathcal{M}, \mathcal{S}') \ , \tag{2}$$

where $\mathcal{L}_{\text{c2s}}$ optimizes the segmentation performance given the CPL, and $\mathcal{L}_{\text{s2c}}$ assesses the CAM quality with respect to SPL. Building on self-distillation [6, 46], we present this swapped self-distillation framework tailored specifically to facilitate information exchange between the CAMs and segmentation.

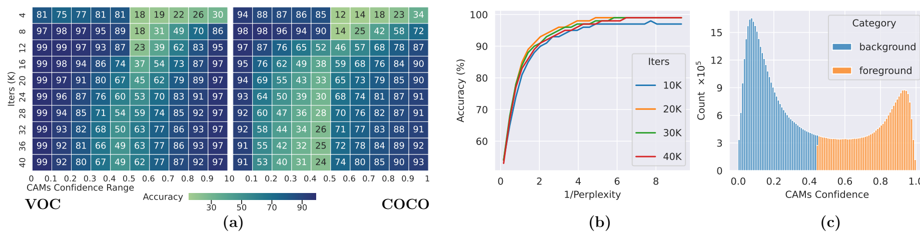


Fig. 3: CPL Analysis (a): heatmap of CPL accuracy *vs.* confident ranges (x-axis) for different time-steps (y-axis) for VOC and COCO. **(b)**: correlation between perplexity and accuracy of CPL for different time-steps. **(c)**: distribution of CAMs' confidence categorized by the proposed dynamic threshold on VOC. See *Supp.* for COCO analysis.

3.3 Segmentation Optimization

CAM2Seg Learning. Previous studies [2, 3, 29, 50] refine the CAMs to obtain pseudo-label, then perform pseudo-label to segmentation learning (PL2Seg). As our guided-CAMs do not require extra refinement process, they can be directly employed as learning targets (CAM2Seg). Nonetheless, CAMs primarily concentrate on the activated regions of the foreground while disregarding the background. As per the established convention [15, 51, 58], a threshold value ξ is employed for splitting the foreground and the background. Formally, our CAM pseudo-label (CPL) is given by:

$$\hat{\mathcal{Y}}^{\mathrm{CPL}}_{x,y} = \begin{cases} \texttt{argmax}(\mathcal{M}'_{x,y}) + 1, & \text{if } \nu \geq \xi, \\ 0, & \text{if } \nu < \xi, \end{cases} \tag{3}$$

where $\nu \triangleq \texttt{max}(\mathcal{M}'_{x,y})$ denotes the the maximum activation, 0 denotes the background index. Then, the CAM2seg learning objective $\mathcal{L}_{\mathrm{c2s}}$ is cross entropy between $\mathcal{Y}^{\mathrm{CPL}}$ and $\mathcal{S}$, as with the general supervised segmentation loss [8].

Reliability based Adaptive Weighting (RAW). Segmentation performance depends heavily on the reliability of the pseudo-labels. Thus, it is important to assess their reliability. Existing methods use post-refinement to enhance pseudo-label credibility [3, 70]. As CoSA can generate online CPL, we propose to leverage confidence information to compensate the CAM2Seg loss during training. Specifically, we propose to assess the perplexity scores for each pixel in $\hat{\mathcal{Y}}^{\mathrm{CPL}}$ and leverage these scores to re-weight $\mathcal{L}_{\mathrm{c2s}}$ for penalizing unreliable regions. However, estimating per-pixel perplexity is non-trivial. Through empirical analysis, we observe a noteworthy association between the confidence values of CAMs and their accuracy at each time-step. This correlation suggests that regions with extremely low or high confidence exhibit higher accuracy throughout training, as shown in Fig. 3(a). To quantitatively model perplexity, we make two assumptions: i) the reliability of pseudo-labels is positively correlated with their accuracy, and ii) the perplexity score is negatively correlated with the reliability. Then, per-pixel perplexity of $\hat{\mathcal{Y}}^{\mathrm{CPL}}_{x,y}$ is defined as:

$$\mathcal{P}_{x,y} = \begin{cases} \left[-\log\left(\lambda_\alpha(\nu - \xi)/(1 - \xi)\right)\right]^{\lambda_\beta} & \text{if } \nu \geq \xi, \\ \left[-\log\left(\lambda_\alpha(\xi - \nu)/\xi\right)\right]^{\lambda_\beta} & \text{if } \nu < \xi, \end{cases} \tag{4}$$

where the term within the logarithm denotes the normalized distance to ξ in $[0, 1]$. The logarithm ensures $\mathcal{P}_{x,y} \to +\infty$ as distance $\to 0$, and $\mathcal{P}_{x,y} \to 0$ as distance $\to 1$. $\lambda_\alpha \in \mathbb{R}^+$ controls the perplexity score's minimum value and $\lambda_\beta \in \mathbb{R}^+$ determines the sharpness or smoothness of the distribution. Higher $\mathcal{P}_{x,y}$ indicates confidence of $\hat{\mathcal{Y}}^{\text{CPL}}_{x,y}$ closer to threshold ξ. This observation is substantiated by Fig. 3(a), where confidence values near $\xi = 0.5$ exhibit lower reliability. Furthermore, the correlation between perplexity and accuracy remains significant across various training time-steps and datasets, as depicted in Fig. 3(b).

Since we hypothesize negative reliability-perplexity correlation, the reliability score can be defined as the reciprocal of perplexity. To accommodate reliability variation for different input, we use the normalized reliability as the per-pixel weights for $\mathcal{L}_{\text{c2s}}$. This arrives our RAW-based CAM2Seg objective:

$$\mathcal{L}_{\text{c2s}}(x,y) = -\frac{|\mathcal{R}|}{\sum_{i,j\in\mathcal{R}} (\mathcal{P}_{i,j}\mathcal{P}_{x,y})^{-1}} \sum_{c=0}^{C} \left[\mathbb{1}[\hat{\mathcal{Y}}^{\text{CPL}}_{x,y} = c] \log\left(\frac{\exp \mathcal{S}^c_{x,y}}{\sum_{k=0}^{C} \exp \mathcal{S}^k_{x,y}} \right) \right], \tag{5}$$

where $|\mathcal{R}|$ represents total number of pixels in a batch.

Dynamic Threshold. Existing WSSS work [50, 51] prescribes a fixed threshold to separate foreground and background, which neglects inherent variability due to prediction confidence fluctuation during training. Obviously, applying a fixed threshold in Fig. 3(a) is sub-optimal.

To alleviate this, we introduce dynamic thresholding. As shown in Fig. 3(c), the confidence distribution reveals discernible clusters. We assume the foreground and background pixels satisfy a bimodal Gaussian Mixture distribution. Then, the optimal dynamic threshold $\xi^\star$ is determined by maximizing the Gaussian Mixture likelihood:

$$\xi^\star = \underset{\xi}{\texttt{argmax}} \prod_{x\in\{\mathcal{M}' \geq \xi\}} \tilde{\pi}_{fg} \mathcal{N}\left(x | \tilde{\mu}_{fg}, \tilde{\Sigma}_{fg}\right) + \prod_{x\in\{\mathcal{M}' < \xi\}} \tilde{\pi}_{bg} \mathcal{N}\left(x | \tilde{\mu}_{bg}, \tilde{\Sigma}_{bg}\right) , \tag{6}$$

where $\mathcal{N}(x|\mu, \Sigma)$ denotes the Gaussian function and π, μ, Σ are the weight, mean and covariance. To avoid mini-batch bias, we maintain a queue to fit GMM, with the current $\mathcal{M}'$ batch enqueued and the oldest dequeued. This facilitates establishment of a gradually evolving threshold, contributing to learning stabilization.

3.4 CAM Optimization

Seg2CAM Learning. To generate SPL, segmentation predictions $\mathcal{S}'$ are filtered by the weak labels Y_{gt} and transformed into probabilities:

$$\mathcal{S}'^{\,c}_{x,y} = \begin{cases} -\infty, & \text{if } Y^c_{\text{gt}} = 0, \\ \mathcal{S}'^{\,c}_{x,y}, & \text{if } Y^c_{\text{gt}} \neq 0, \end{cases} \qquad \hat{\mathcal{Y}}^{\text{SPL}}_{x,y} = \texttt{Softmax}(\frac{\mathcal{S}'_{x,y}}{\tau}) , \tag{7}$$

where τ denotes the temperature to sharpen $\hat{\mathcal{Y}}^{\text{SPL}}_{x,y}$. Let $\mathcal{R}$ be all the positions in SPL, then the Seg2CAM learning objective is defined as:

$$\mathcal{L}_{\text{s2c}} = -\frac{1}{C|\mathcal{R}|} \sum_{c=1}^{C} \sum_{x,y\in\mathcal{R}} \left[\hat{\mathcal{Y}}^{\text{SPL}}_{x,y}[c] \log(\sigma(\mathcal{M}^c_{x,y})) + (1 - \hat{\mathcal{Y}}^{\text{SPL}}_{x,y}[c]) \log(1 - \sigma(\mathcal{M}^c_{x,y})) \right] . \tag{8}$$

Fig. 4: Illustration of Coexistence Problem in CAMs. The first row shows the input images. The second row shows the coexistence problem e.g. 'bird' with 'branches', 'train' with 'railways' and 'boat' with 'the sea'.

Coexistence Problem in CAMs. Certain class activations often exhibit large false positive regions, where objects are incorrectly merged with surroundings, as shown in Fig. 4. For instance, the classes 'bird' and 'tree branches', 'train' and 'railways', *etc.* frequently appear together in VOC dataset. We refer to this issue as the coexistence problem. We hypothesize that the coexistence problem is attributed to three factors: i) Objects that coexist in images, such as 'tree branches', are not annotated *w.r.t.* weak labels, which makes it challenging for a model to semantically distinguish coexistence. ii) Training datasets lack sufficient samples for such classes. iii) High-level feature maps, though rich in abstract representations and semantic information, lack essential low-level features such as edges, textures, and colors [24]. Thus, CAMs generated from the last layer are poor in low-level information for segmenting objects. Conversely, segmenting objects with high-level semantics is hindered due to factors i) and ii).

Contrastive Separation in CAMs. We posit that the effective usage of low-level information can alleviate the coexistence problem. Since shallower-layer feature is rich in low-level info [69], we propose to extract CAMs $\mathcal{M}^{\dagger}$ from an earlier layer, and present its comparison with $\mathcal{M}$ in Fig. 5, showing that directly substituting $\mathcal{M}$ with $\mathcal{M}^{\dagger}$ is not feasible due to the lower mIoU upperbound of $\mathcal{M}^{\dagger}$. However, if we consider the confident regions in $\mathcal{M}$ and $\mathcal{M}^{\dagger}$, *i.e.* filter by a low-pass perplexity, then $\{\mathcal{M}^{\dagger}_{x,y} \mid \mathcal{P}_{x,y} \leq \epsilon\}$ result in higher mIoU than $\{\mathcal{M}_{x,y} \mid \mathcal{P}_{x,y} \leq \epsilon\}$, as shown in Fig. 5(b), where ϵ denotes a low-pass coefficient. Further, we observe in some examples the presence of coexistence issues in $\mathcal{M}$ but absence in $\mathcal{M}^{\dagger}$ as shown in Fig. 5(c). This suggests that $\mathcal{M}^{\dagger}$ performs worse than $\mathcal{M}$ in general, but better for those regions with low perplexity. Driven by these findings, we propose to regularize $\mathcal{M}$ by $\mathcal{M}^{\dagger\prime}$ (from AN). Specifically, $\mathcal{M}^{\dagger\prime}$ after a low-pass filter are used to determine the positive $\mathcal{R}^{+}_{i,j}$ and negative $\mathcal{R}^{-}_{i,j}$ regions:

$$
\begin{aligned}
\mathcal{R}^{+}_{i,j} &= \Big\{(x,y) \mid \mathcal{P}_{x,y} \leq \epsilon,\ \hat{y}^{\text{CPL}}_{x,y} = \hat{y}^{\text{CPL}}_{i,j}, (x,y) \neq (i,j)\Big\} \\
\mathcal{R}^{-}_{i,j} &= \Big\{(x,y) \mid \mathcal{P}_{x,y} \leq \epsilon,\ \hat{y}^{\text{CPL}}_{x,y} \neq \hat{y}^{\text{CPL}}_{i,j}\Big\} ,
\end{aligned}
\tag{9}
$$

where $(i,j) \in \Omega$, $\Omega = \{(x,y) \mid \mathcal{P}_{x,y} \leq \epsilon\}$ is low-perplexity region in $\mathcal{M}^{\dagger\prime}$, and $\hat{y}^{\text{CPL}}$ represents the CPL of $\mathcal{M}^{\dagger\prime}$. Then, we have contrastive separation loss for $\mathcal{M}$:

$$
\mathcal{L}_{\text{csc}} = -\frac{1}{|\Omega|} \sum_{i,j \in \Omega} \frac{1}{|\mathcal{R}^{+}_{i,j}|} \sum_{x,y \in \mathcal{R}^{+}_{i,j}} \log \frac{L^{i,j}_{x,y}}{L^{i,j}_{x,y} + K^{i,j}_{n,m}} , \tag{10}
$$

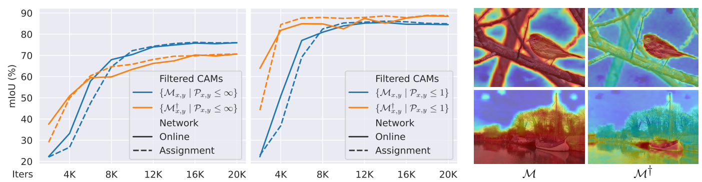


Fig. 5: $\mathcal{M}$ **and** $\mathcal{M}^{\dagger}$ **Comparisons. (a)**: mIoU *vs.* time-steps for $\mathcal{M}$ and $\mathcal{M}^{\dagger}$ on VOC `val`. **(b)**: same as **(a)** but filtered by perplexity. **(c)**: cases of coexistence in $\mathcal{M}$ but not in $\mathcal{M}^{\dagger}$.

where $L_{x,y}^{i,j} = \exp(l_d(\mathcal{M}_{i,j}, \mathcal{M}_{x,y})/\tau)$, $l_d(a,b)$ measures the similarity between (a,b), τ denotes the InfoNCE loss [45] temperature, and $K_{n,m}^{i,j} = \sum_{n,m \in \mathcal{R}_{i,j}^{-}} L_{n,m}^{i,j}$.

Overall Objectives. The objectives encompass the aforementioned losses and a further $\mathcal{L}_{\text{c2s}}^{\mathcal{M}^{\dagger}}$ to stabilize training and accelerate convergence, resulting in the CoSA objective:

$$\mathcal{L}_{\text{CoSA}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{cls}}^{\mathcal{M}^{\dagger}} + \lambda_{\text{c2s}} \left(\mathcal{L}_{\text{c2s}} + \mathcal{L}_{\text{c2s}}^{\mathcal{M}^{\dagger}} \right) + \lambda_{\text{s2c}} \mathcal{L}_{\text{s2c}} + \lambda_{\text{csc}} \mathcal{L}_{\text{csc}}. \tag{11}$$

4 Experiments

4.1 Experiment Details and Results

Datasets. We evaluate on two benchmarks: VOC [19] and COCO [39]. VOC encompasses 20 categories with `train`, `val`, and `test` splits of 1464, 1449, and 1456 images. Following WSSS practice [2,3,62], SBD [23] is used to augment the `train` split to 10,582. COCO contains 80 categories with `train` and `val` splits of approx. 82K and 40K images. Our model is trained and evaluated using *only* the image-level classification labels[3], and employing mIoU as evaluation metrics.

Implementation Details. Following [51], we use ImageNet pretrained ViT-base (ViT-B) [17] as the encoder. For classification, we use global max pooling (GMP) [49] and the CAM approach [72]. For the segmentation decoder, we use LargeFOV [8], as with [51]. ON is trained with AdamW [44]. The learning rate is set to 6E-5 in tandem with polynomial decay. AN is updated with a momentum of 0.9994. For preprocessing, the images are cropped to 448^2, then weak/strong augmentations are applied (see *Supp.*). The perplexity constants $(\lambda_\alpha, \lambda_\beta)$ are set to $(0.8, 1)$, GMM-fitting queue length is 100, and softmax temperature τ is 0.01. The low perplexity threshold ϵ is set to 1 and the loss weight factors $(\lambda_{\text{c2s}}, \lambda_{\text{s2c}}, \lambda_{\text{csc}})$ to $(0.1, 0.05, 0.1)$.

Semantic Segmentation Comparison. We compare our method with existing SOTA WSSS methods on VOC and COCO for semantic segmentation in Tab. 1. CoSA achieves 76.2% and 75.1% on VOC12 `val` and `test`, respectively, surpassing the highest-performing single-stage model (ToCo) by 5.1% and 2.9%, as well as all multi-stage methods, including those with additional supervision. In the COCO evaluation, CoSA consistently outperforms other approaches,

[3] Not available for VOC `test` split and so not used in evaluation.

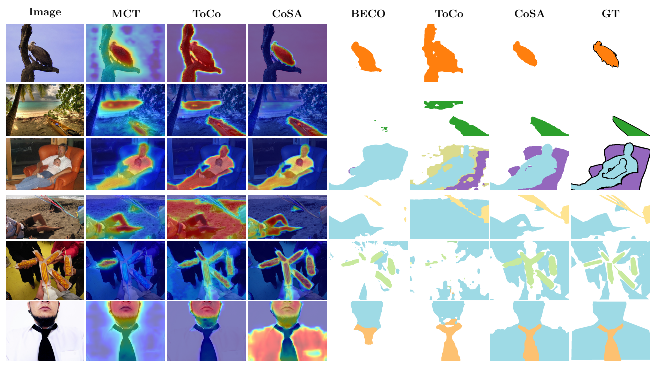
demonstrating a significant increase of 8.7% over the top-performing single-stage methods. Further, there is a also 2.7% improvement over the leading multi-stage method [13]. While our primary goal is to provide an end-to-end WSSS solution, we also offer a multi-stage version of CoSA, denoted as **CoSA-MS** in Tab. 1, where various standalone segmentation networks are trained using our CPL. Our CoSA-MS models can also attains SOTA performance in multi-stage scenarios.

Methods	***Sup.***	***Net.***	**VOC**		**COCO**
			`val`	`test`	`val`
***Supervised Upperbounds*.**					
Deeplab [8] TPAMI'2017	$\mathcal{F}$	R101	77.6	79.7	–
WideRes38 [59] PR'2019	$\mathcal{F}$	WR38	80.8	82.5	–
ViT-Base [17] ICLR'2021	$\mathcal{F}$	ViT-B	80.5	81.0	–
UperNet-Swin [42] ICCV'2021	$\mathcal{F}$	SWIN	83.4	83.7	–
***Multi-stage Methods*.**					
L2G [26] CVPR'2022	$\mathcal{I}+\mathcal{S}$	R101	72.1	71.7	44.2
Du *et al.* [18] CVPR'2022	$\mathcal{I}+\mathcal{S}$	R101	72.6	73.6	–
CLIP-ES [40] CVPR'2023	$\mathcal{I}+\mathcal{L}$	R101	73.8	73.9	45.4
ESOL [37] NeurIPS'2022	$\mathcal{I}$	R101	69.9	69.3	42.6
BECO [48] CVPR'2023	$\mathcal{I}$	R101	72.1	71.8	45.1
Mat-Label [57] ICCV'2023	$\mathcal{I}$	R101	73.0	72.7	45.6
CoSA-MS	$\mathcal{I}$	R101	**76.5**	**75.3**[1]	**50.9**
Xu *et al.* [63] CVPR'2023	$\mathcal{I}+\mathcal{L}$	WR38	72.2	72.2	45.9
W-OoD [33] CVPR'2022	$\mathcal{I}$	WR38	70.7	70.1	–
MCT [62] CVPR'2022	$\mathcal{I}$	WR38	71.9	71.6	42.0
ex-ViT [68] PR'2023	$\mathcal{I}$	WR38	71.2	71.1	42.9
ACR-ViT [31] CVPR'2023	$\mathcal{I}$	WR38	72.4	72.4	–
MCT+OCR [15] CVPR'2023	$\mathcal{I}$	WR38	72.7	72.0	42.0
CoSA-MS	$\mathcal{I}$	WR38	**76.6**	**74.9**[2]	**50.1**
ReCAM [14] CVPR'2022	$\mathcal{I}$	SWIN	70.4	71.7	47.9
LPCAM [13] CVPR'2023	$\mathcal{I}$	SWIN	73.1	73.4	48.3
CoSA-MS	$\mathcal{I}$	SWIN	**81.4**	**78.4**[3]	**53.7**
***Single-stage (End-to-end) Methods*.**					
RRM [70] AAAI'2020	$\mathcal{I}$	WR38	62.6	62.9	–
AFA [50] CVPR'2022	$\mathcal{I}$	MiT-B1	66.0	66.3	38.9
RRM [70]† AAAI'2020	$\mathcal{I}$	ViT-B	63.1	62.4	–
ViT-PCM [49] ECCV'2022	$\mathcal{I}$	ViT-B	69.3	–	45.0
ToCo [51] CVPR'2023	$\mathcal{I}$	ViT-B	71.1	72.2	42.3
SeCo [66] CVPR'2024	$\mathcal{I}$	ViT-B	74.0	73.8	46.7
CoSA	$\mathcal{I}$	ViT-B	76.2	75.1[4]	51.0
CoSA*	$\mathcal{I}$	ViT-B	**76.4**	**75.2**[5]	**51.1**

Table 1: Weakly Supervised Semantic Segmentation Results. *Sup.*: supervision type. *Net.*: segmentation backbone. $\mathcal{F}$: Fully supervised, $\mathcal{I}$: Image-level labels, $\mathcal{S}$: Saliency maps, $\mathcal{L}$: language models. $*$ represents CRF [8] postprocessing results.

CAM Quality Comparison. Tab. 2 shows CoSA's CPL results compared with existing WSSS methods. Our method yields 78.5% and 76.4% mIoU on `train` and `val`. Notably, an ensemble of $\mathcal{M}'$ and $\mathcal{M}^{\dagger\prime}$ improves performance to 78.9% and 77.2%, suggesting the activation of $\mathcal{M}'$ is orthogonal to that of $\mathcal{M}^{\dagger\prime}$.

Qualitative Comparison. Fig. 6 presents CAMs and segmentation visualizations, comparing with recent methods: MCT, BECO, and ToCo. As shown, our method can generate improved CAMs and produce well-aligned segmentation, exhibiting superior results in challenging segmentation problems with intra-class



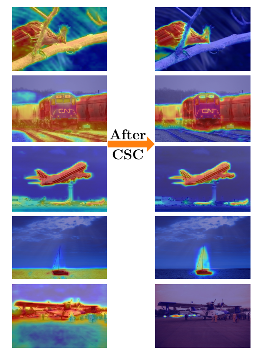


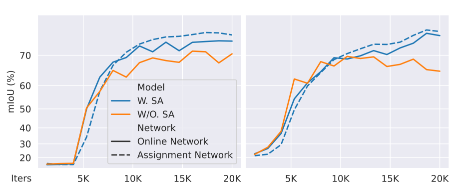
Fig. 6: Qualitative Comparison. The results are reported on the `val` splits of VOC (in *R1* - *R3*) and COCO (in *R4* - *R6*). The official codebases and provided weights for MCT [62], BECO [48], and ToCo [51] are used for this comparison. (best viewed under zoom; see *Supp.* for more).

Fig. 7: Effect of CSC. The class activation for bird, train, plane, boat and car are presented from top to bottom.

variation and occlusions. In addition, CoSA performs well *w.r.t.* the coexistence cases (Fig. 6 *R1*, *R2*), while existing methods struggle. Moreover, CoSA reveals limitations in the GT segmentation (Fig. 6 *R4*).

4.2 Ablation Studies

CoSA Module Analysis. We begin by employing CAMs directly as the supervision signal for segmentation, akin to [70], albeit without refinement, and gradually apply CoSA modules to this baseline. As shown in Tab. 3(a), the mIoUs progressively improve with addition of our components. Further, we examine the efficacy of each CoSA component. As shown in Tab. 3(b), the elimination of each component results in deteriorated performance, most notably for CSC.



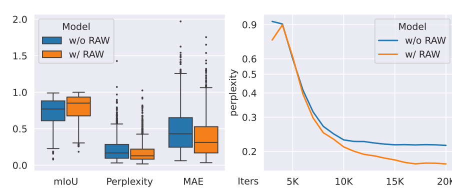


Fig. 8: Ablative Study of SA. The performance of SPL (*left*) and CPL (*right*) *w.r.t.* iterations on VOC `val` set are shown for CoSA with or without SA.

Fig. 9: Ablation Study of RAW. (*left*) boxplot of mIoU, perplexity and MAE to (1,0) for individual CPLs on VOC `val`. (*right*) perplexity reduction over times.

Method	ViT-PCM [49]	ACR-ViT [31]	CLIP-ES [40]	SeCo [66]	ToCo [51]	CoSA	CoSA$^\bullet$
`train`	71.4	70.9	75.0	76.5	73.6	78.5	**78.9**
`val`	69.3	–	–	–	72.3	76.4	**77.2**

Table 2: Comparisons of CPL. All methods use ViT as the backbone for generating the CAMs on VOC dataset. $\bullet$ represents the ensemble of $\mathcal{M}'$ and $\mathcal{M}^{\dagger\prime}$ in CoSA.

(a)

Base.	GC	SA	RAW	CSC	DT	mIoU (inc.) VOC	COCO
✓						55.96	37.32
✓	✓					63.09 (+7.13)	42.55 (+5.23)
✓	✓	✓				64.41 (+8.45)	43.92 (+6.60)
✓	✓	✓	✓			68.22 (+12.26)	45.39 (+8.07)
✓	✓	✓		✓		71.66 (+15.70)	47.10 (+9.78)
✓	✓	✓	✓	✓		75.54 (+19.58)	49.67 (+12.35)
✓	✓	✓	✓	✓	✓	**76.19** (+20.23)	**51.00** (+13.68)

(b)

CoSA	GC	SA	RAW	CSC	DT	mIoU (dec.) VOC	COCO
✓						**76.19**	**51.00**
✓					✗	75.54 (-0.65)	49.67 (-1.33)
✓				✗		69.89 (-6.30)	45.95 (-5.05)
✓			✗			72.45 (-3.74)	47.83 (-3.17)
✓		✗				72.10 (-4.09)	49.04 (-1.96)
✓	✗					74.12 (-2.07)	49.67 (-1.33)

Table 3: Ablation Study on Contribution of Each Component. ***(a)***: gradually add proposed components to baseline. ***(b)***: systematically exclude components from CoSA. **GC**: Guided CAMs, **SA**: Swapping Assignments, **RAW**: Reliability based Adaptive Weighting, **CSC**: Contrastive Separation in CAMs, and **DT**: Dynamic Threshold. **mIoU** is reported on PASCAL VOC12 and COCO `val` splits.

(a)

Source	Detach	`train`	`val`
GT	None	83.99	80.16
NO	–	72.28	71.38
SPL	F	74.19	73.36
SPL	W_{fc}	78.05	76.15
SPL	None	**78.54**	**76.37**

(b)

Method	C-mIoU	mIoU
FPR [9]	53.09	53.34
ToCo [51]	63.62	72.33
SeCo [66]	73.18	73.63
w/o CSC	62.61	67.82
w/ CSC	**82.34**	**76.37**

(c)

Use CRF?	mIoU(%) ✓	mIoU(%) ✗	Speed ✓	Speed ✗
BECO-R101	72.1	70.9(-1.2)	1.95	4.94
COSA-R101	76.5	76.4**(-0.1)**	2.36	**9.60**
ToCo-ViT	71.1	69.2(-1.9)	1.82	3.99
COSA-ViT	76.4	76.2**(-0.2)**	1.83	**4.11**

Table 4: Ablation Study of GC, CSC and CRF. (a): CPL performance comparison on VOC. **Source**: guidance sources. **Detach**: stop gradient in GC for feature map F or W_{fc}. **(b)**: CPL performance comparison. FPR, ToCo and SeCo results are based on their code and weights. **C-mIoU**: mIoU for classes with coexistence. **(c)**: CRF Impact. Best speed-accuracy tradeoff is achieved without using CRF. Inference speeds (FPS) are tested on RTX 3090.

Impact of Guided CAMs. Our model is compared with a baseline [70] that directly uses CAMs as CPL. As shown in Tab. 4(a), our guided CAMs notably enhance CPL quality by 6.26% and 4.99% for `train` and `val` splits. Further, we conduct experiments to ascertain the extent to which the two CAM components, feature F and weights W_{fc}, exert greater impact on guiding CAMs. As shown, the deteachment of F results in 74.19% and 73.36%, but W_{fc} can decrease the results slightly to 78.05% and 76.37%. This suggests that guiding CAMs primarily optimizes feature maps, verifying our hypothesis of the non-deterministic feature map contributing to the stochasticity of CAMs in Sec. 3.1.

Impact of Swapping Assignments (SA). Tab. 3(b) suggests that eliminating SA results in significant mIoU decreases, highlighting the importance of this training strategy. Further examination of the ON and AN *w.r.t.* SPL and CPL indicates that, in later training stages, AN consistently outperforms ON for both SPL and CPL, as shown in Fig. 8, due to AN performing a form of model ensembling similar to Polyak-Ruppert averaging [47,52]. We observe a noticeable disparity of mIoUs between two ONs (solid orange line *vs.* solid blue line in Fig. 8), which may be attributed to the superior quality of CPL and SPL from the AN facilitating a more robust ON for CoSA. The momentum framework, originally introduced to mitigate noise and fluctuations of the online learning target [6,22], is used for info exchange across CAMs and segmentation in CoSA.

Impact of RAW. Tab. 3(b) shows notable mIoU reduction without RAW. We conduct further studies to investigate its effect on perplexity reduction. The boxplot in Fig. 9 suggests that RAW leads to higher mIoU but lower perplexity. Fig. 9(*right*) illustrates a faster decrease in perplexity when RAW is used, affirming its impact on perplexity reduction.

Impact of CSC. Our CSC is introduced to address the coexistence issues. We establish C-mIoU to measure the CAM quality for those coexistence-affected classes. As shown in Tab. 4(b), applying CSC sees a boost in C-mIoU and mIoU, which surpass the existing methods. Some visual comparisons are given in Fig. 7.

Impact of Dynamic Threshold. We evaluate CoSA using some predetermined thresholds, comparing them with one employing dynamic threshold on VOC `val` split (see *Supp.* for results). The performance is sensitive to the threshold, but dynamic thresholding achieves 0.65% increased performance over the best manual finetuning while saving 80% of hyper-parameter searching time.

4.3 Further Analysis

Training and Inference Efficiency Analysis. Unlike multi-stage approaches, CoSA can be trained end-to-end efficiently. Compared to BECO [48], our method is 240% faster in training, uses 50% fewer parameters, and yields a 4.3% higher mIoU on VOC `test`. Please refer to *Supp.* for more discussion. At inference time, we find that CRF post-processing, which is commonly adopted for refining masks [15,48] or the CAMs [49,62,70], can greatly slow down the inference speed. Through our experiments, we show that CoSA does not heavily depend on CRF: incorporating CRF results in marginal improvement of 0.2%, 0.1%, and 0.1% for VOC `val`, VOC `test`, and COCO `val`, respectively (Tab. 1). Conversely, eliminating CRF in CoSA can greatly speed up inference (a noteworthy 307% and 165% $\uparrow$) and achieve better speed-accuracy tradeoff as suggested in Tab. 4(c).

5 Conclusion

This paper presents an end-to-end WSSS method: Co-training with Swapping Assignments (CoSA), which eliminates the need for CAM refinement and enables concurrent CAM and segmentation optimization. Our empirical study reveals the non-deterministic behaviors of CAMs and that proper guidance can mitigate such stochasticity, leading to substantial quality enhancement. We propose explicit CAM optimization leveraging segmentation pseudo-labels in our approach, where a dual-stream model comprising an online network for predicting CAMs and segmentation masks, and an ancillary assignment network providing swapped assignments (SPL and CPL) for training, is introduced. We further propose three techniques within this framework: RAW, designed to mitigate the issue of unreliable pseudo-labels; contrastive separation, aimed at resolving coexistence problems; and a dynamic threshold search algorithm. Incorporating these techniques, CoSA outperforms all SOTA methods on both VOC and COCO WSSS benchmarks while achieving exceptional speed-accuracy trade-off.

Acknowledgments

The work is supported by European Research Council under the European Union's Horizon 2020 research and innovation programme (GA No 787768).

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2209–2218 (2019)
2. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4981–4990 (2018)
3. Araslanov, N., Roth, S.: Single-stage semantic segmentation from image labels. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4253–4262 (2020)
4. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: European Conference on Computer Vision (ECCV). pp. 549–565. Springer (2016)
5. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Neural Information Processing Systems (NeurIPS)**33**, 9912–9924 (2020)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 9650–9660 (2021)
7. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 839–847. IEEE (2018)
8. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)**40**(4), 834–848 (2017)
9. Chen, L., Lei, C., Li, R., Li, S., Zhang, Z., Zhang, L.: Fpr: False positive rectification for weakly supervised semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV). pp. 1108–1118 (2023)
10. Chen, L., Wu, W., Fu, C., Han, X., Zhang, Y.: Weakly supervised semantic segmentation with boundary exploration. In: European Conference on Computer Vision (ECCV). pp. 347–362. Springer (2020)
11. Chen, Q., Yang, L., Lai, J.H., Xie, X.: Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4288–4298 (2022)
12. Chen, Z., Tian, Z., Zhu, J., Li, C., Du, S.: C-cam: Causal cam for weakly supervised semantic segmentation on medical image. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11676–11685 (2022)
13. Chen, Z., Sun, Q.: Extracting class activation maps from non-discriminative features as well. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3135–3144 (2023)
14. Chen, Z., Wang, T., Wu, X., Hua, X.S., Zhang, H., Sun, Q.: Class re-activation maps for weakly-supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 969–978 (2022)

15. Cheng, Z., Qiao, P., Li, K., Li, S., Wei, P., Ji, X., Yuan, L., Liu, C., Chen, J.: Out-of-candidate rectification for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 23673–23684 (2023)
16. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV). pp. 1635–1643 (2015)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR)(2021)
18. Du, Y., Fu, Z., Liu, Q., Wang, Y.: Weakly supervised semantic segmentation by pixel-to-prototype contrast. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4320–4329 (2022)
19. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision (IJCV) **88**, 303–338 (2010)
20. Fan, J., Zhang, Z., Tan, T., Song, C., Xiao, J.: Cian: Cross-image affinity net for weakly supervised semantic segmentation. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 34, pp. 10762–10769 (2020)
21. Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., Zhou, B., Ye, Q.: Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In: IEEE International Conference on Computer Vision (ICCV). pp. 2886–2895 (2021)
22. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Neural Information Processing Systems (NeurIPS)**33**, 21271–21284 (2020)
23. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: IEEE International Conference on Computer Vision (ICCV). pp. 991–998. IEEE (2011)
24. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 447–456 (2015)
25. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
26. Jiang, P.T., Yang, Y., Hou, Q., Wei, Y.: L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 16886–16896 (2022)
27. Ke, T.W., Hwang, J.J., Yu, S.: Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In: International Conference on Learning Representations (ICLR)(2020)
28. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 876–885 (2017)
29. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Neural Information Processing Systems (NeurIPS)**24** (2011)
30. Kweon, H., Yoon, S.H., Kim, H., Park, D., Yoon, K.J.: Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV). pp. 6994–7003 (2021)

31. Kweon, H., Yoon, S.H., Yoon, K.J.: Weakly supervised semantic segmentation via adversarial learning of classifier and reconstructor. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 11329–11339 (2023)
32. Lee, J., Kim, E., Lee, S., Lee, J., Yoon, S.: Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 5267–5276 (2019)
33. Lee, J., Oh, S.J., Yun, S., Choe, J., Kim, E., Yoon, S.: Weakly supervised semantic segmentation using out-of-distribution data. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 16897–16906 (2022)
34. Lee, J., Yi, J., Shin, C., Yoon, S.: Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2643–2652 (2021)
35. Lee, S., Lee, M., Lee, J., Shim, H.: Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 5495–5505 (2021)
36. Li, J., Fan, J., Zhang, Z.: Towards noiseless object contours for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 16856–16865 (2022)
37. Li, J., Jie, Z., Wang, X., Ma, L., et al.: Expansion and shrinkage of localization for weakly-supervised semantic segmentation. In: Neural Information Processing Systems (NeurIPS)(2022)
38. Li, Y., Duan, Y., Kuang, Z., Chen, Y., Zhang, W., Li, X.: Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 36, pp. 1447–1455 (2022)
39. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
40. Lin, Y., Chen, M., Wang, W., Wu, B., Li, K., Lin, B., Liu, H., He, X.: Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 15305–15314 (2023)
41. Liu, S., Liu, K., Zhu, W., Shen, Y., Fernandez-Granda, C.: Adaptive early-learning correction for segmentation from noisy annotations. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2606–2616 (2022)
42. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE International Conference on Computer Vision (ICCV). pp. 10012–10022 (2021)
43. Liu, Z., Qi, X., Fu, C.W.: One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1726–1736 (2021)
44. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
45. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
46. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
47. Polyak, B.T., Juditsky, A.B.: Acceleration of stochastic approximation by averaging. SIAM journal on control and optimization **30**(4), 838–855 (1992)

48. Rong, S., Tu, B., Wang, Z., Li, J.: Boundary-enhanced co-training for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 19574–19584 (2023)
49. Rossetti, S., Zappia, D., Sanzari, M., Schaerf, M., Pirri, F.: Max pooling with vision transformers reconciles class and shape in weakly supervised semantic segmentation. In: European Conference on Computer Vision (ECCV). pp. 446–463. Springer (2022)
50. Ru, L., Zhan, Y., Yu, B., Du, B.: Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 16846–16855 (2022)
51. Ru, L., Zheng, H., Zhan, Y., Du, B.: Token contrast for weakly-supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3093–3102 (2023)
52. Ruppert, D.: Efficient estimations from a slowly convergent robbins-monro process. Tech. rep., Cornell University Operations Research and Industrial Engineering (1988)
53. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017)
54. Shimoda, W., Yanai, K.: Self-supervised difference detection for weakly-supervised semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV). pp. 5208–5217 (2019)
55. Song, C., Huang, Y., Ouyang, W., Wang, L.: Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3136–3145 (2019)
56. Sun, K., Shi, H., Zhang, Z., Huang, Y.: Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 7283–7292 (2021)
57. Wang, C., Xu, R., Xu, S., Meng, W., Zhang, X.: Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV). pp. 755–765 (2023)
58. Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 12275–12284 (2020)
59. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition **90**, 119–133 (2019)
60. Xie, J., Hou, X., Ye, K., Shen, L.: Clims: Cross language image matching for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4483–4492 (2022)
61. Xie, J., Xiang, J., Chen, J., Hou, X., Zhao, X., Shen, L.: C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 989–998 (2022)
62. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Xu, D.: Multi-class token transformer for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4310–4319 (2022)
63. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Xu, D.: Learning multi-modal class-specific tokens for weakly supervised dense object localization. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 19596–19605 (2023)

64. Xu, R., Wang, C., Sun, J., Xu, S., Meng, W., Zhang, X.: Self correspondence distillation for end-to-end weakly-supervised semantic segmentation. In: AAAI Conference on Artificial Intelligence (AAAI). AAAI Press (2023)
65. Yang, X., Burghardt, T., Mirmehdi, M.: Dynamic curriculum learning for great ape detection in the wild. International Journal of Computer Vision (IJCV) **131**(5), 1163–1181 (2023)
66. Yang, Z., Fu, K., Duan, M., Qu, L., Wang, S., Song, Z.: Separate and conquer: Decoupling co-occurrence via decomposition and representation for weakly supervised semantic segmentation. arXiv preprint arXiv:2402.18467 (2024)
67. Yoon, S.H., Kweon, H., Cho, J., Kim, S., Yoon, K.J.: Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation. In: European Conference on Computer Vision (ECCV). pp. 326–344. Springer (2022)
68. Yu, L., Xiang, W., Fang, J., Chen, Y.P.P., Chi, L.: ex-vit: A novel explainable vision transformer for weakly supervised semantic segmentation. Pattern Recognition p. 109666 (2023)
69. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision (ECCV). pp. 818–833. Springer (2014)
70. Zhang, B., Xiao, J., Wei, Y., Sun, M., Huang, K.: Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In: AAAI Conference on Artificial Intelligence (AAAI). vol. 34, pp. 12765–12772 (2020)
71. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 1325–1334 (2018)
72. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 2921–2929 (2016)
73. Zhou, T., Zhang, M., Zhao, F., Li, J.: Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 4299–4309 (2022)