Weakly Supervised Semantic Segmentation via Adversarial Learning of Classifier and Reconstructor

Hyeokjun Kweon*, Sung-Hoon Yoon*, and Kuk-Jin Yoon
Visual Intelligence Lab., KAIST, Korea
{0327june,yoon307,kjyoon}@kaist.ac.kr

Abstract

*In Weakly Supervised Semantic Segmentation (WSSS), Class Activation Maps (CAMs) usually 1) do not cover the whole object and 2) be activated on irrelevant regions. To address the issues, we propose a novel WSSS framework via adversarial learning of a classifier and an image reconstructor. When an image is perfectly decomposed into class-wise segments, information (*i.e.*, color or texture) of a single segment could not be inferred from the other segments. Therefore, inferability between the segments can represent the preciseness of segmentation. We quantify the inferability as a reconstruction quality of one segment from the other segments. If one segment could be reconstructed from the others, then the segment would be imprecise. To bring this idea into WSSS, we simultaneously train two models: a classifier generating CAMs that decompose an image into segments and a reconstructor that measures the inferability between the segments. As in GANs, while being alternatively trained in an adversarial manner, two networks provide positive feedback to each other. We verify the superiority of the proposed framework with extensive ablation studies. Our method achieves new state-of-the-art performances on both PASCAL VOC 2012 and MS COCO 2014. The code is available at https://github.com/sangrockEG/ACR.*

1. Introduction

Over the past decade, learning-based semantic segmentation has made significant advancements. However, the high labeling cost remains a major challenge when applying existing methods to real cases. In response to this challenge, without relying on pixel-wise supervision, Weakly Supervised Semantic Segmentation (WSSS) has been proposed to learn semantic segmentation with weak labels only.

The field of WSSS has studied several types of weak labels such as scribbles [29, 36], bounding boxes [16, 23, 31], and image-level classification labels [1–3, 7, 12, 19, 21, 28, 37, 42, 46, 47, 49, 52], which are relatively inexpensive to acquire. Among them, using image-level labels is the most widely studied setting due to its high accessibility and efficiency. Different from the other supervisions that roughly notify the locations of the things in the image, image-level labels only contain categorical information. Therefore, the main challenge in WSSS using image-level labels is localizing the regions of each class.

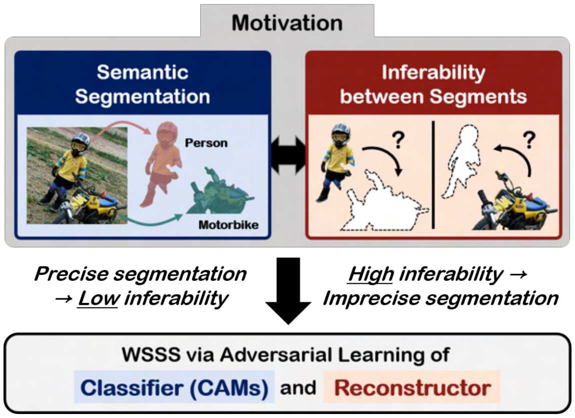
Motivation
Semantic Segmentation
Person
Motorbike
Inferability between Segments
?
?
Precise segmentation → Low inferability
High inferability → Imprecise segmentation
WSSS via Adversarial Learning of
Classifier (CAMs) and Reconstructor

Figure 1. Demonstration of our method. We are motivated by the relation between semantic segmentation and inferability. We realize it as adversarial learning of a classifier and a reconstructor.

To dispel this challenge, most of the existing literature employs Class Activation Maps (CAMs) [51] that highlight the regions highly contributing to the prediction of a classifier. Intuitively, the classifier learns shared patterns among the images including the same class, and thereby the CAM of each class is activated on the image regions corresponding to the class. However, the CAMs usually show a tendency to focus on the most discriminative regions of each class (*e.g.* face for the *cat* class), which leads to incomplete segmentation. Also, due to the absence of pixel-wise supervision, the CAMs are rather imprecise at the boundary, which is a critical issue from the perspective of segmentation. Technically, the goal of WSSS can be summarized as obtaining better CAMs, which can serve as precise pseudo-labels for learning semantic segmentation.

*The first two authors contributed equally. In alphabetical order.

In this paper, we propose a novel method inspired by simple but meaningful intuition as in Fig. 1. If semantic segmentation is perfectly accomplished, each of the objects in the image is perfectly segmented into mutually independent segments in terms of color and texture. In this case, each of the segments does not include any clue about the rest of the image. Therefore, if segmentation of the image is perfectly performed, no single segment can "infer" colors or textures about the other segments. As a contra-positive statement, if any information such as color or textures about a segment could be inferred from the other segments, the semantic segmentation could be regarded as imperfect. Accordingly, it might be possible to measure the quality of semantic segmentation based on the inferability between the segments. However, how could we quantify the degree of "inferability"? To this end, we propose to employ the image reconstruction task, which reconstructs one image segment from the other segments. Then, the quality of reconstruction could be regarded as a measure of inferability. Here, note that the image reconstruction task does not introduce any additional supervision.

We formulate the aforementioned intuition as an adversarial learning of a classifier and a reconstructor. In specific, according to the CAMs obtained by the classifier, we decompose an image into two segments: a segment of the target class and a segment of the non-target class (the other classes). The reconstructor is trained to reconstruct one segment by using the other segment as the only input. On the other hand, we promote the CAMs to decompose an image into segments that **reduce** the inferability of the reconstructor. In other words, the classifier is trained to not only classify the image but also generate CAMs correctly segmenting the image, while competing with the reconstructor. Ultimately, we improve the quality of the CAMs by jointly training the two competitors (*i.e*., the classifier and the reconstructor) in an adversarial manner.

The adversarial learning strategy of our framework is similar to Generative Adversarial Networks (GANs) [14]. Like the discriminator in GANs is specialized to discriminate the real/fake samples, the reconstructor in our framework is trained to fully exploit the **remnant** contained in the given segment for reconstructing the other segment. Similarly, the classifier in our framework learns to generate precise CAMs, using the reconstructor as a measure of the inferability between the segments, like the generator getting feedback from the discriminator in GANs. Consequently, our adversarial learning framework can achieve WSSS using only the supervision that comes from the image-level classification labels and the input images themselves.

The proposed method has methodological similarity to the existing Adversarial Erasing (AE) methods of WSSS in that it erases (or spatially decomposes) the image according to CAMs. However, the insights behind our method and the AE methods are far different. AE methods mask the highly activated regions of the CAMs from the image and impose classification loss on the remained image. Therefore, due to the lack of regularization for the erasing process, the CAMs usually suffer from undesirable expansion. On the other hand, the proposed method is inspired by the relation between segmentation and reconstruction. And we formulate it as adversarial learning between two networks performing each task. This realization not only provides reliable guidance for CAMs from the perspective of segmentation, but also enables each network to improve while training proceeds, based on the positive feedback from its counterpart.

To verify the superiority of our method, we conduct extensive ablation studies and comparisons with the other state-of-the-art (SoTA) WSSS methods. Further, on both PASCAL VOC 2012 [11] and MS COCO [30] datasets, the proposed framework achieves a new SoTA.

The contribution of this paper is threefold:

- We formulate the problem of WSSS as minimizing inferability between the segments decomposed by the CAMs.
- We propose a novel WSSS framework based on adversarial learning of the classifier and reconstructor.
- We achieve state-of-the-art performance on both the PASCAL VOC 2012 *val/test* set and MS COCO *val* set.

2. Related Works

With their localization capability, Class Activation Maps (CAMs) have been widely employed in WSSS to generate pixel-level pseudo-labels. However, the original CAMs do not fully cover the whole object and have imprecise boundaries. To relieve these, WSSS studies have focused on 1) improving the CAMs (seeds) or 2) post-processing the acquired CAMs into more reliable pseudo-labels (masks).

2.1. CAMs Improvements

To explicitly expand the CAMs, various WSSS studies have been conducted while exploring the sub-category classification [3], cross-image relationships [13, 26, 34], information bottleneck [20], intra-class boundaries [12], mutually exclusive patches [49], and attention mechanisms [32, 37,39]. Along them, the others have tried to relieve the issue from the perspective of data, using hard Out-of-Distribution (OoD) data [22] or specialized augmentation for foreground and background [33]. Recently, studies based on contrastive learning [7, 42, 52] proposed to learn the feature while minimizing/maximizing the distance to the prototypes.

Adversarial Erasing (AE) methods [19,25,35,46,50] expand the CAMs while exploring the object from the erased images. The AE methods share a degree of methodological similarity with the proposed method in that they spatially decompose the image/feature according to CAMs; however, our method has a novel and distinct insight. Since AE methods impose classification loss on the remained image and there is no regularization for the erasing phase, it is inevitable to suffer from undesirable expansion, well known as an over-erasing problem. Recently, OC-CSE [19] proposes to handle the problem by using the guidance of a pre-trained classifier; however, the guidance is fixed, and thereby the achievable performance is strongly bounded. On the other hand, in our method, we formulate WSSS as adversarial learning of the classifier and the reconstructor. As far as we know, it is the first approach to utilize the reconstruction task for guiding the CAMs. Further, since the two networks provide positive feedback to each other, the reconstructor could provide an effective regularization, free from the over-erasing that has plagued the AE methods.

2.2. Mask Refinements

In addition to the method of improving the CAMs themselves, several post-processing methods have been proposed to improve the quality of the pseudo-labels, based on the semantic affinities among adjacent pixels [1, 2], anti-adversarial manipulation [21], and boundary information [5]. The other approaches target the noisy nature of the CAMs and relieve it while refinement, by using an underfitting strategy with reweighting [28] or an uncertainty of the CAMs [27]. Several studies [12,15,24,26,34,43,45,52] have employed saliency detection module to provide precise object boundary information when generating pseudo-labels. However, such modules require additional dataset (and labels) for training. Considering the goal of WSSS, we abstain to use the saliency module in this paper.

3. Motivation

A semantic segmentation task can be regarded as a decomposition of an image into class-wise segments. For the given C classes, an image $\mathbf{I} \in [0,1]^{3\times H\times W}$ could be decomposed into C class-wise segments as follows:

$$\mathbf{I} = \sum_{k=1}^{C} \mathbf{I} \odot \mathbf{M}_k, \tag{1}$$

where $\mathbf{M}_k$ is a $H \times W$ size occupancy mask of the class k and $\odot$ denotes an element-wise multiplication.

In this paper, we interpret WSSS as a task to infer the masks from an image, by using the image-level classification labels only. For this, the most straightforward approach is directly employing the CAM of a certain class as the mask of that class. Since the CAMs usually highlight the objects of the corresponding class from an image, they could serve as noisy targets (i.e. pseudo-labels) for semantic segmentation. However, the CAMs not only fail to localize the whole regions of the corresponding class but also usually invade the regions of the other classes.

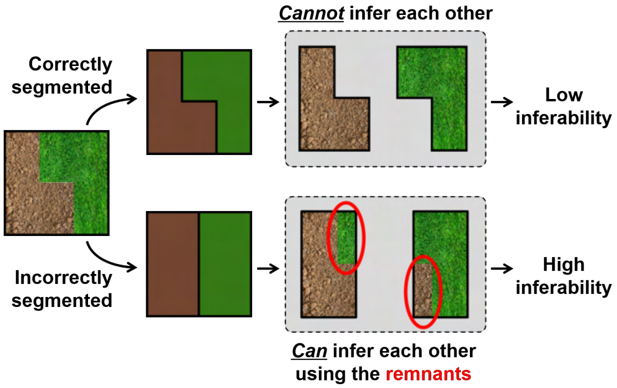

Figure 2. Our motivation. If segmentation is correctly done, no single segment can infer information about the other segment. On the other hand, if the segmentation is incorrectly performed, then the remnants (the incorrectly segmented regions circled by red) can be the clue for the inference between the segments.

To relieve the issues, we propose a novel WSSS framework inspired by the motivation visualized in Fig. 2. When an image is correctly segmented (upper branch), each segment does not include information about the other segment. Therefore, a segment of a certain class could not be inferred from the segments of the other class. In other words, correct segmentation leads to low "***inferability***" between the segments. On the other hand, if the segmentation results are incorrect (lower branch in Fig. 2), the remnants (*i.e.*, miss-segmented regions denoted by red circles) could serve as clues for inferring one segment from the other segment. Therefore, if the segments have high inferability between them, then the semantic segmentation would be incorrect. It implies that the inferability between the segments can role as a measure of the quality of the semantic segmentation.

In order to incorporate this concept into the learning of CAMs, it is necessary to quantitatively measure the inferability. To quantify it, we need a **reconstructor** such that: (1) when segmentation is imprecise, which means there exist some miss-segmented regions (*i.e.*, remnants), the reconstructor should be able to reconstruct one segment from the other segments using remnants and (2) when the segmentation is perfect, the reconstructor should fail to reconstruct one segment from the other segments, due to the lack of remnants. However, in a weakly-supervised setting, it is challenging to obtain such an appropriate reconstructor.

As a remedy, we formulate this intuition as **an adversarial learning of a classifier and a reconstructor**. The goal of the classifier is not only classifying the image, but also generating the CAMs that can correctly segment the image. For this, we first sample a target class among the classes existing in the image. Then, using the CAM of the target class, we decompose the image into two segments: the target segment (the regions activated by the CAM) and the non-target segment (the regions not activated by the CAM). Here, the

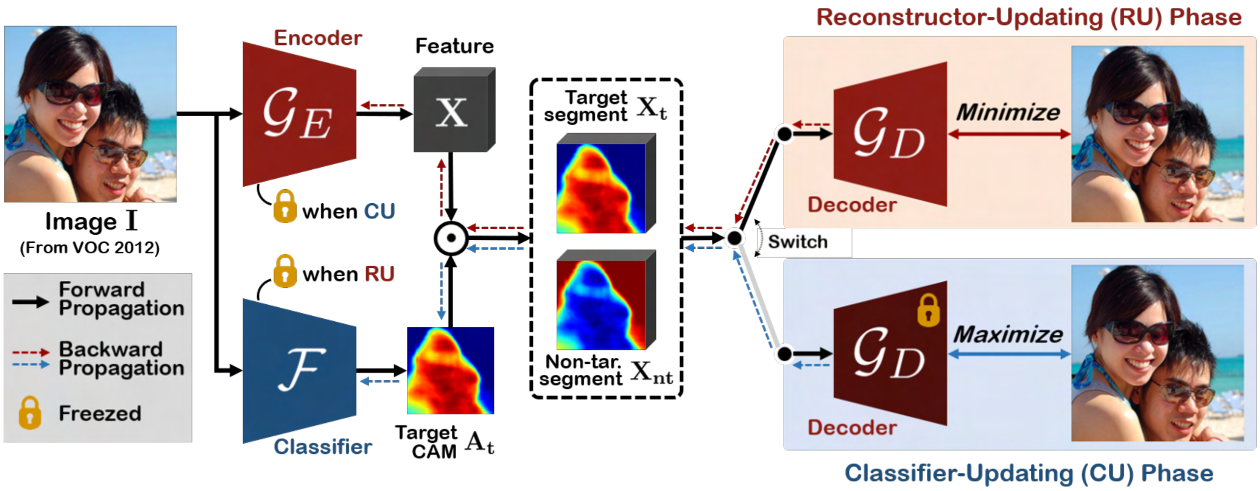

Figure 3. Visualization of the proposed framework. We input an image $\mathbf{I}$ to the Reconstructor Encoder $\mathcal{G}_E$ and the classifier $\mathcal{F}$ to acquire a feature $\mathbf{X}$ and a target CAM $\mathbf{A}_t$, respectively. Then, according to the target CAM, we decompose the feature into target segment $\mathbf{X}_t$ and non-target segment $\mathbf{X}_{nt}$. The segments are fed to the Reconstructor-Updating (RU) Phase and Classifier-Updating (CU) Phase, in an alternative manner. Using the Reconstructor Decoder $\mathcal{G}_D$, the RU and CU phases reconstruct images from the segments and compute the loss between the reconstructed results and the input image. Note that the red and blue dashed lines denote the back-propagation from RU phase and CU phase, respectively. We omit the classification branch of the classifier and class-specific sampling process for simplicity.

classifier and the reconstructor compete on the quality of the reconstruction of one segment from the other segment. The classifier learns to generate CAMs that could correctly segment the images (*i.e.*, make the segments have low inferability between them), and thereby make the reconstructor fail to reconstruct. On the contrary, the reconstructor learns to correctly reconstruct the segments, by exploiting the incorrectly segmented remnants. As a result, while training the networks to achieve the opposite goal, we can obtain CAMs that can precisely segment the image.

4. Methods

4.1. Overall Framework

In this paper, we propose a novel WSSS framework via Adversarial learning of the Classifier and the Reconstructor (**ACR**). As explained in Section 3, the insight behind our method is independency between the segments of classes. The overall framework is visualized in Fig. 3.

In the proposed framework, the classifier $\mathcal{F}$ and the reconstructor $\mathcal{G}$ are jointly trained. The reconstructor is a combination of feature encoder $\mathcal{G}_E$ and decoder $\mathcal{G}_D$. We first obtain CAMs $\mathbf{A} \in \mathbb{R}^{C\times h\times w}$ and feature $\mathbf{X} \in \mathbb{R}^{d\times h\times w}$ from the classifier and the encoder, respectively, as follows:

$$\mathbf{A}, \mathbf{p} = \mathcal{F}(I) \quad \text{and} \quad \mathbf{X} = \mathcal{G}_E(I), \tag{2}$$

where $\mathbf{p}$ is a class prediction for the image $\mathbf{I}$. Then, similar to the class-specific erasing [19], we sample one target class t among the classes existing in the image, and regard the corresponding CAM as a target CAM ($\mathbf{A}_t$). Using the target CAM, we decompose the feature $\mathbf{X}$ into a target segment $\mathbf{X}_t$ and a non-target segment $\mathbf{X}_{nt}$, as follows:

$$\mathbf{X}_t = \mathbf{A}_t \odot \mathbf{X} \quad \text{and} \quad \mathbf{X}_{nt} = (1 - \mathbf{A}_t) \odot \mathbf{X}. \tag{3}$$

Here, the element-wise multiplication ($\odot$) is differentiable, and thereby either the target CAM or feature can be trained with gradients back-propagated through $\mathbf{X}_t$ or $\mathbf{X}_{nt}$. As aforementioned, if the target CAM is precise, then the inferability between the target segment and the non-target segment would be low. In other words, if one segment can be reconstructed from the other segment, then the CAM could be regarded as imprecise. Therefore, the reconstructor is trained to correctly reconstruct one segment by using the other segment, while the classifier is trained to generate the target CAM that makes the reconstructor fail to reconstruct. For this, similar to the generator and the discriminator in GANs [14], the classifier and the reconstructor are competitively trained in every iteration.

In this paper, we devise the alternative training scheme with Reconstructor-Updating (RU) Phase and Classifier-Updating (CU) Phase, as in Fig. 3. In **RU Phase**, the reconstructor is trained to correctly reconstruct one segment by using the other segment only. The goal is achieved by **minimizing** the loss between the reconstructed image segment and the original image segment. This loss is back-propagated to the reconstructor ($\mathcal{G}_E$ and $\mathcal{G}_D$) only, and the classifier is not updated. During the RU Phase, we expect that the reconstructor learns to reconstruct one segment from the other, using the remnants that are miss-segmented due to the imprecise CAMs.

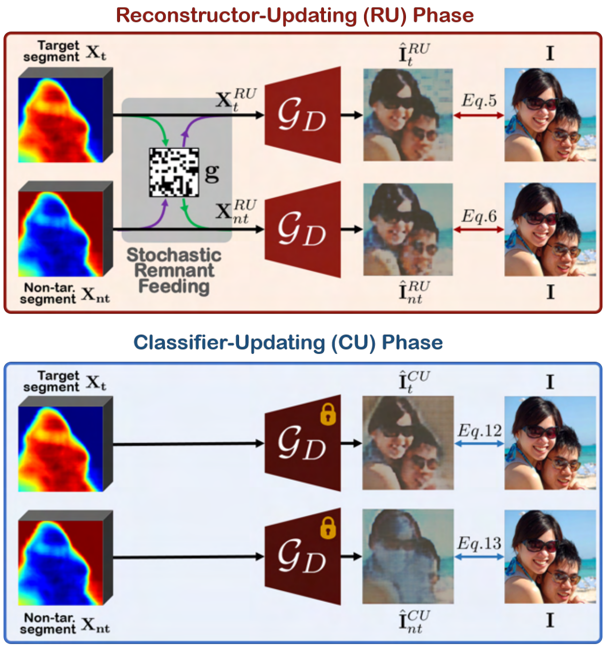
Reconstructor-Updating (RU) Phase
Target segment $\mathbf{X}_t$
$\hat{\mathbf{I}}_t^{RU}$
I
$\mathbf{X}_t^{RU}$
$\mathcal{G}_D$
Eq.5
g
$\mathbf{X}_{nt}^{RU}$
$\mathcal{G}_D$
Eq.6
Stochastic Remnant Feeding
Non-tar. segment $\mathbf{X}_{nt}$
$\hat{\mathbf{I}}_{nt}^{RU}$
I
Classifier-Updating (CU) Phase
Target segment $\mathbf{X}_t$
$\hat{\mathbf{I}}_t^{CU}$
I
$\mathcal{G}_D$
Eq.12
$\mathcal{G}_D$
Eq.13
Non-tar. segment $\mathbf{X}_{nt}$
$\hat{\mathbf{I}}_{nt}^{CU}$
I

Figure 4. Visualization of Reconstructor-Updating (RU) Phase and Classifier-Updating (CU) Phase. In RU phase, with the proposed Stochastic Remnant Feeding (SRF), we prevent the memorization of the training dataset. Note that the Reconstructor is updated in RU phase only and the classifier is updated in CU Phase.

On the other hand, in **CU Phase**, the classifier is trained to generate the target CAM which makes the reconstructor fail. For this, opposite to the RU phase, we **maximize** the difference between the reconstructed image and the original image within a certain segment. In this phase, the loss only penalizes the classifier, while the reconstructor is frozen. To achieve the goal of spoiling the reconstruction result, the classifier learns to generate the target CAM more precisely without remaining any miss-segmented remnants.

4.2. Reconstructor-Updating Phase

4.2.1 Basic Formulation

We visualize the RU Phase as the red box in Fig. 4. The RU phase originally includes Stochastic Remnant Feeding (SRF) strategy which will be explained in Sec. 4.2.2. However, for better understanding, we first explain the basic formulation without SRF in this subsection.

In the RU phase, note that we only optimize the reconstructor (*i.e.*, the encoder and the decoder) while the classifier remains frozen. Therefore, in Eq. 3, the loss is backpropagated to the reconstructor only, through the feature $\mathbf{X}$. We input the segments ($\mathbf{X}_t$ and $\mathbf{X}_{nt}$) to the decoder $\mathcal{G}_\mathcal{D}$ and obtain reconstruction results as:

$$\hat{\mathbf{I}}_t^{RU} = \mathcal{G}_\mathcal{D}(\mathbf{X}_t) \quad \text{and} \quad \hat{\mathbf{I}}_{nt}^{RU} = \mathcal{G}_\mathcal{D}(\mathbf{X}_{nt}). \tag{4}$$

The goal of the reconstructor is to correctly reconstruct the target segment by using the non-target segment only, and vice versa. For the case of the target segment, we reduce the difference between the reconstructed result $\hat{\mathbf{I}}_t^{RU}$ and the input image $\mathbf{I}$, within the non-target region. This constraint can be imposed by minimizing the following loss:

$$\mathcal{L}_t^{RU} = |\mathbf{V}_t^{RU} \odot (\mathbf{I} - \hat{\mathbf{I}}_t^{RU})|_1, \tag{5}$$

where $\mathbf{V}_t^{RU} = (1 - \mathbf{A}_t)$ is the soft validation mask indicating the non-target region and $|\cdot|_1$ denotes L1 loss. Similarly, for the non-target segment, we minimize the following loss:

$$\mathcal{L}_{nt}^{RU} = |\mathbf{V}_{nt}^{RU} \odot (\mathbf{I} - \hat{\mathbf{I}}_{nt}^{RU})|_1. \tag{6}$$

Here, we set $\mathbf{V}_{nt}^{RU} = \mathbf{A}_t$, since the validation of reconstruction should be conducted on the target region. To sum up, the total loss for training the reconstructor is

$$\mathcal{L}^{RU} = \lambda_t^{RU} \mathcal{L}_t^{RU} + \lambda_{nt}^{RU} \mathcal{L}_{nt}^{RU}, \tag{7}$$

where the lambdas are weighting parameters. Please note that the classifier is **not** optimized within this phase.

4.2.2 Stochastic Remnant Feeding

With the described process, the reconstructor can learn to exploit the remnant in one segment for reconstructing the other segment. However, as training proceeds, we observe that forcing to minimize the loss of Eq. 7 leads our framework to an undesirable local minimum. When the CAMs become more precise, the regional features (X_t, X_{nt}) are more perfectly decomposed, and thereby each segment contains smaller remnants of the other segment. In this case, the reconstructor tends to minimize the loss of Eq. 7 by memorizing the training dataset and "generating" the original image, rather than reconstructing the segments based on the remnants, unlike our design intention. When this happens, the reconstruction can be correctly done even when the classifier achieves precise CAMs, and thereby the condition (2) in Section 3 is violated.

To relieve this, we devise a strategy named **Stochastic Remnant Feeding (SRF)**, visualized as a grey box in Fig. 4. Instead of perfectly decomposing the features into the target and non-target features as in Eq. 3, we make synthetic remnants of each segment and feed them to the feature of the other segment. Then, the reconstructor could always exploit the remnants for reconstructing segments. Therefore, with the proposed SRF strategy, we can regularize the undesirable memorization of the reconstructor, while keeping the capability to exploit the remnants for reconstruction. For this, we define a random binary grid $\mathbf{g} \in [0, 1]^{h \times w}$ composed of $s \times s$ patches. Each $\frac{h}{s} \times \frac{w}{s}$ size cell has a value of 0 or 1, sampled from independent Bernoulli distribution with a probability of q. With this grid, we define a target feature for RU phase ($\mathbf{X}_t^{RU}$) as follows:

$$\mathbf{X}_t^{RU} = \mathbf{X}_t + \mathbf{g} \odot \mathbf{X}_{nt}. \tag{8}$$

Here, the term $\mathbf{g} \odot \mathbf{X}_{nt}$ represents the synthetic remnants sampled by SRF. Similarly, the non-taget feature for RU phase ($\mathbf{X}_{nt}^{RU}$) is as

$$\mathbf{X}_{nt}^{RU} = \mathbf{X}_{nt} + \mathbf{g} \odot \mathbf{X}_t. \quad (9)$$

With the processes described above, we update the features as $\mathbf{X}_t, \mathbf{X}_{nt} \xrightarrow{SRF} \mathbf{X}_t^{RU}, \mathbf{X}_{nt}^{RU}$. Therefore, Eq. 4 is modified into $\hat{\mathbf{I}}_t^{RU} = \mathcal{G}_{\mathcal{D}}(\mathbf{X}_t^{RU})$ and $\hat{\mathbf{I}}_{nt}^{RU} = \mathcal{G}_{\mathcal{D}}(\mathbf{X}_{nt}^{RU})$. Accordingly, we also update the validation masks $\mathbf{V}_t^{RU}$ and $\mathbf{V}_{nt}^{RU}$ as

$$\mathbf{V}_t^{RU} = (1-\mathbf{A}_t)\odot(1-\mathbf{g}) \text{ and } \mathbf{V}_{nt}^{RU} = \mathbf{A}_t\odot(1-\mathbf{g}). \quad (10)$$

Note that Eq. 5-7 are remained the same. We use the SRF strategy by default of our setting, and conduct an ablation study on it, which will be explained in Section 5.3.

4.3. Classifer-Updating Phase

The visualization of the CU Phase is shown in the blue box of Fig. 4. Here, in contrast to the RU Phase, only the classifier is trained while the reconstructor is not updated. Therefore, in Eq. 3, the loss is back-propagated to the classifier only, through the target CAM $\mathbf{A}_t$. Similar to the RU phase, we first obtain the reconstructed images as follows:

$$\hat{\mathbf{I}}_t^{CU} = \mathcal{G}_{\mathcal{D}}(\mathbf{X}_t) \quad \text{and} \quad \hat{\mathbf{I}}_{nt}^{CU} = \mathcal{G}_{\mathcal{D}}(\mathbf{X}_{nt}). \quad (11)$$

Here, note that we do not use the SRF strategy, which is devised only for the RU phase. As described in Section 3, in CU phase, the goal of the classifier is to generate a target CAM that prevents the reconstructor from restoring the original image. This strategy is similar to optimizing the generator to fool the discriminator in GANs [14]. For the target segment, we train the classifier to maximize the difference between the reconstruction result $\hat{\mathbf{I}}_t^{CU}$ and the image $\mathbf{I}$ on the non-target region, while fixing the reconstructor. We impose this by minimizing the following equation:

$$\mathcal{L}_t^{CU} = -|\mathbf{V}_t^{CU} \odot (\mathbf{I} - \hat{\mathbf{I}}_t^{CU})|_1, \quad (12)$$

where $\mathbf{V}_t^{CU} = (1 - \mathbf{A}_t)$ indicates non-target region. Please note the **minus sign**, which means that minimizing the loss leads the framework to maximize the difference between the $\hat{\mathbf{I}}_t^{CU}$ and $\mathbf{I}$. Similarly, we minimize the following loss also for the non-target ($\cdot_{nt}$).

$$\mathcal{L}_{nt}^{CU} = -|\mathbf{V}_{nt}^{CU} \odot (\mathbf{I} - \hat{\mathbf{I}}_{nt}^{CU})|_1. \quad (13)$$

Here, we set $\mathbf{V}_{nt}^{CU} = \mathbf{A}_t$ to indicate target region. Finally, the total loss for training the classifier is as follows:

$$\mathcal{L}^{CU} = \mathcal{L}_{cls}^{CU} + \lambda_t^{CU}\mathcal{L}_t^{CU} + \lambda_{nt}^{CU}\mathcal{L}_{nt}^{CU}, \quad (14)$$

where $\mathcal{L}_{cls}^{CU}$ denotes the binary cross-entropy loss between the class prediction (p in Eq. 2) and image-level classification labels. Note that only the classifier is trained in this phase, and the reconstructor is not optimized.

Table 1. Ablation study on the learning strategy. We evaluate mIoU performance on the PASCAL VOC 2012 *train* set. **Bold** numbers represent the best results.

Learning strategy for reconstructor	SRF	mIoU (%)
Baseline		48.4
Pre-trained		52.9
Pre-trained	✓	54.6
Adversarial		55.8
Adversarial	✓	**60.3**

5. Experimental Results

5.1. Dataset and Evaluation Metric

As conventional WSSS works, we evaluate our method on two benchmarks: PASCAL VOC 2012 dataset [11] and MS-COCO dataset [30]. VOC 2012 and MS-COCO datasets contain 21 and 81 categories including background, respectively. We follow the official *train/val/test* split for WSSS on both datasets. As an evaluation metric, we use the mean Intersection over Union (mIoU) between the prediction and the GT semantic map. Note that, for training, we only use image-level class labels.

5.2. Implementation Details

Architectures In our framework, we use ResNet38 [40] as the backbone of the classifier. We attach a 1×1 convolution layer as the classification head to generate CAMs, as in [50]. For the image reconstruction, UNet-based network is used. In the Encoder, we aggregate the multiscale features from multiple different layers. This enables the reconstructor to use primitive details (which come from low-level features) more easily. For the segmentation network, we use Deeplab [4] with ResNet38 backbone as in [2, 19, 28, 35, 44, 46, 49]. More details on the architecture can be found in the *Supp. Materials*.

Data Augmentation Random cropping, resizing, and horizontal flipping are applied to the input. The crop size is set to 256 with [0.5, 1.3] resizing range. Color jittering [18] is widely used for augmentation; however, we do not use it due to the unstable convergence of reconstructor.

Training Policy As in [6], we use a poly learning rate that multiplies $(1 - \frac{iter}{max\ iter})^{0.9}$ to the initial learning rate (0.01 for our framework and 0.001 for Deeplab). The whole framework is trained for 40 epochs, which took around 12 hours with a single RTX 3090 ti. For the weighting parameters in Eq. 7, λ_t^{RU} and λ_{nt}^{RU} are set equally to 0.5. For the λ_t^{CU} and λ_{nt}^{CU} in Eq. 14, we set the values to 0.8 and 0.3 respectively to balance the magnitude of each term. Additional details (including COCO) are in the *Supp. Material*.

Table 2. Ablation study on the loss function in Eq. 14. We evaluate precision, recall, and mIoU performance on the PASCAL VOC 2012 *train* set. **Bold** numbers represent the best results.

	Loss function			Metrics		
	$\mathcal{L}_{cls}^{CU}$	$\mathcal{L}_{t}^{CU}$	$\mathcal{L}_{nt}^{CU}$	Precision	Recall	mIoU (%)
	✓			0.61	0.72	48.4
(a)	✓	✓		0.73 (+0.12)	0.68 (-0.04)	53.4 (+5.0)
(b)	✓		✓	0.58 (-0.03)	**0.77 (+0.05)**	51.5 (+3.1)
(c)	✓	✓	✓	**0.75 (+0.14)**	0.76 (+0.04)	**60.3 (+13.6)**

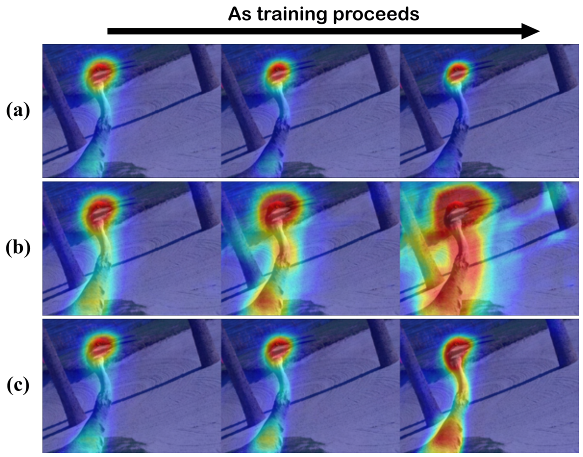

Figure 5. Qualitative comparison between the ablated settings and ours. The change of CAMs as training proceeds is shown for each case (a), (b), and (c) of Table 2.

5.3. Ablation Studies

We conduct an ablation study to demonstrate the benefit of adversarial learning of the classifier and the reconstruction, which is our main idea. We first solely train a classifier with a mere classification loss. The performance of the CAMs of this baseline is shown in the first row of Table 1.

Then, with the baseline CAMs, a reconstructor is trained as in our framework. Note that, unlike the proposed framework, we freeze the baseline classifier and optimize the reconstructor only. After that, we train a new classifier with the pre-trained reconstructor. In this phase, we freeze the reconstructor and train the classifier only. Compared with ours, the reconstructor and the classifier are always solely trained in this ablated setting. Therefore, unlike the proposed adversarial framework that is alternatively trained in every iteration, they have no chance to give and take positive feedback from competitors.

We provide the quantitative result of the ablation study in Table 1. We can observe that the proposed adversarial learning outperforms the ablated setting (denoted as pre-trained) by a large margin. Also, the proposed SRF strategy provides great gains while preventing undesirable over-fitting in both settings. These results imply two insights: 1) our motivation still works even when we use the pre-trained reconstructor while freezing it, 2) and adversarial learning can even further exploit the potential of our motivation.

Table 3. Comparisons between our method and the other WSSS methods. We evaluate mIoU (%) on the PASCAL VOC 2012 *train* set at three levels: CAM, w/ CRF, and Mask. For a fair comparison, we split the methods into two groups upon the backbones (ResNet and ViT [9]). (W)RN denotes (wide)ResNet. **Bold** and underlined numbers represent the best and the second best results.

Methods	Backbone	seed	w/ CRF	Mask
CONTA [48] $_{NeurIPS20}$	WRN38	56.2	65.4	66.1
EDAM [39] $_{CVPR21}$	WRN38	52.8	58.2	68.1
AdvCAM [21] $_{CVPR21}$	RN50	55.6	62.1	68.0
ECS [35] $_{ICCV21}$	WRN38	56.6	58.6	-
OC-CSE [19] $_{ICCV21}$	WRN38	56.0	62.8	66.9
CDA [33] $_{ICCV21}$	WRN38	58.4	-	66.4
PMM [28] $_{ICCV21}$	WRN38	58.2	61.5	61.0
RIB [20] $_{NeurIPS}$	RN50	56.5	62.9	70.6
AMR [32] $_{AAAI22}$	RN50	56.8	-	69.7
ReCAM [8] $_{CVPR22}$	RN50	54.8	-	70.5
SIPE [7] $_{CVPR22}$	RN50	58.6	64.7	-
CLIMS [41] $_{CVPR22}$	WRN38	56.6	-	70.5
W-OoD [22] $_{CVPR22}$	RN50	53.3	58.4	
PPC [10] $_{CVPR22}$	WRN38	**61.5**	64.0	70.1
AEFT [46] $_{ECCV22}$	WRN38	56.0	63.5	71.0
Ours (ACR)	WRN38	60.3	**65.9**	**72.3**
MCT [44] $_{CVPR22}$	ViT	61.7	-	69.1
Ours (ACR + ViT [9])	ViT	**65.5**	-	**70.9**

We also ablate the loss function for training the classifier (Eq. 14). In specific, while keeping the classification loss, we ablate each term and observe the change in mIoU performance. The precision and recall achieved by each setting are also provided in Table 2. If we ablate the non-target term as in (a), the target term could be minimized by overly reducing the target CAM (high precision and low recall). On the contrary, when we ablate the target term as in (b), then the classifier tends to overly expand the target CAM (low precision and high recall) to minimize the non-target term. Finally, in (c), our method using all terms outperforms the other settings by a large margin. The qualitative comparison visualized in Fig. 5 further clarifies the main insight of this ablation study.

5.4. Comparisons to State-of-The-Arts

For training the semantic segmentation model, we refine the CAMs generated by our method to the pseudo-labels using IRN [1] as previous WSSS methods. In Table 3, we compare the mIoU performance of the proposed method with the other WSSS methods on three levels: CAMs, CAMs refined by denseCRF [17], and the pseudo-labels (Mask). The result implies not only that the proposed framework generates high-quality CAMs, but also that the achieved gain is not overlapped with the gain of refinement techniques widely used in WSSS. Note that, since the use of refinement techniques is common in WSSS, it is important to obtain CAM that largely benefits from them.

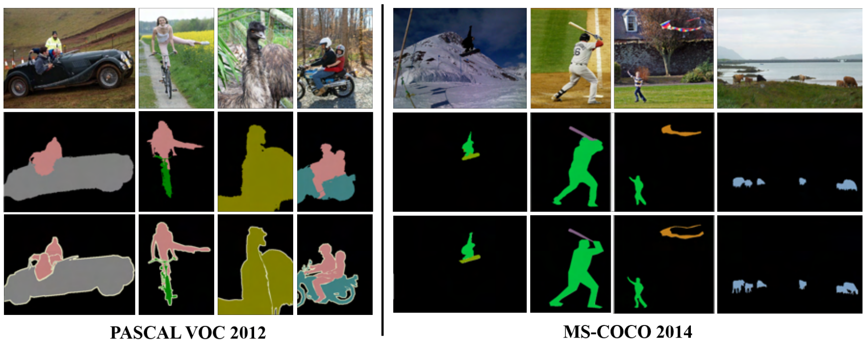

Figure 6. Our semantic segmentation results on VOC 2012 (left) and COCO 2014 (right). From top to bottom: Image, Ours, GT.

Further, for a fair comparison with the recent works using a Vision Transformer (ViT) [9], we incorporate the proposed method with it. For this, we refine the CAMs obtained by adversarial learning, using the patch attention of ViT as in MCTformer-V2 [44]. The pseudo-labels quality of Ours(+ViT) is also shown in Table 3. Compared with the baseline, the proposed framework still provides a meaningful gain, even when we use a different backbone. The results show that the proposed method was originally designed for CNN, but can be integrated with various backbones like ViT. Additional details regarding of Ours(+ViT) can be found in the *Supp. Material*.

In Table 4, the performance of the semantic segmentation model trained by our pseudo-labels is shown. To show the superiority of the proposed method, we compare our performance with that of the other SoTA WSSS methods. The proposed framework achieves a new SoTA in terms of semantic segmentation. The results strongly support the superiority of the proposed framework. Along with the quantitative comparison results, qualitative segmentation results are provided in Fig. 6. Since the semantic segmentation network is trained with the precise pseudo-labels generated by the proposed methods, it captures not only fine details but also the global semantic context. Additional CAMs and segmentation results can be found in the *Supp. Material*.

Table 4. Comparison in mIoU (%) performance between the proposed method and the existing WSSS methods. Evaluation is conducted on the PASCAL VOC 2012 and MS-MOCO 2014. For pair comparison, we list the methods using image-level classification labels only in this table. **Bold** numbers represent the best results.

Methods	*Backbone*	VOC *val*	VOC *test*	COCO *val*
AffinityNet [2]$_{CVPR18}$	WRN38	61.7	63.7	-
IRNet [1]$_{CVPR19}$	RN50	63.5	64.8	41.4
SEAM [37]$_{CVPR20}$	WRN38	64.5	65.7	31.9
OC-CSE [19]$_{ICCV21}$	WRN38	68.4	68.2	36.4
CPN [49]$_{ICCV21}$	WRN38	67.8	68.5	-
RIB [20]$_{NeurIPS21}$	RN101	68.3	68.6	43.8
PMM [28]$_{ICCV21}$	WRN38	68.5	69.0	36.7
ReCAM [8]$_{CVPR22}$	RN101	68.5	68.4	42.9
SIPE [7]$_{CVPR22}$	RN101	68.8	69.7	-
SIPE [7]$_{CVPR22}$	WRN38	-	-	43.6
CLIMS [41]$_{CVPR22}$	RN50	69.3	68.7	
W-OoD [22]$_{CVPR22}$	WRN38	70.7	70.1	
Spatial-BCE [38]$_{ECCV22}$	RN101	70.0	71.3	-
Spatial-BCE [38]$_{ECCV22}$	VGG16	-	-	35.2
AEFT [46]$_{ECCV22}$	WRN38	70.9	71.7	44.8
Ours (ACR)	WRN38	**71.9**	**71.9**	**45.3**
MCT [44] $_{CVPR22}$	WRN38	71.9	71.6	42.0
Ours (ACR + ViT [9])	WRN38	**72.4**	**72.4**	**-**

6. Conclusion

Weakly Supervised Semantic Segmentation (WSSS) studies have utilized Class Activation Maps (CAMs) for localization; however, the CAMs usually provide imprecise activation. To dispel this problem, we draw a simple but powerful motivation: the high inferability between the segments implies the low semantic segmentation quality. We formulate this motivation as adversarial learning of a classifier and a reconstructor, where the inferability between the segments is quantified as a reconstruction quality. Once the classifier decomposes the image into segments according to its CAMs, then the reconstructor is trained to reconstruct a single segment from the other segments correctly. On the other hand, the classifier is trained to generate high-quality CAMs, which makes the reconstructor fail to cross-reconstruct the segments. Similar to GANs, the classifier and the reconstructor provide positive feedback to each other while being alternatively trained in an adversarial manner. We verify the superiority of our method with extensive ablation studies. Further, we achieve new SoTA performances on PASCAL VOC 2012 and MS COCO 2014, outperforming the other WSSS methods.

Acknowledgements This work was supported by Institute of Information and Communications Technology Planning & Evaluation(IITP) grant funded by Korea Government (MSIT) (No.2020-0-00440, Development of Artificial Intelligence Technology that Continuously Improves Itself as the Situation Changes in the Real World and No.2014-3-00123, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis).

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2019. 1, 3, 7, 8

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. 1, 3, 6, 8

[3] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8991–9000, 2020. 1, 2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015. 6

[5] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *European Conference on Computer Vision*, pages 347–362. Springer, 2020. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 6

[7] Qi Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4288–4298, 2022. 1, 2, 7, 8

[8] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2022. 7, 8

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7, 8

[10] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4329, 2022. 7

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 2, 6

[12] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4283–4292, 2020. 1, 2, 3

[13] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10762–10769, 2020. 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2, 4, 6

[15] Peng-Tao Jiang, Yuqi Yang, Qibin Hou, and Yunchao Wei. L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16886–16896, 2022. 3

[16] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017. 1

[17] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 7

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 6

[19] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Daehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6994–7003, 2021. 1, 2, 3, 4, 6, 7, 8

[20] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 34:27408–27421, 2021. 2, 7, 8

[21] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4071–4080, 2021. 1, 3, 7

[22] Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, Eunji Kim, and Sungroh Yoon. Weakly supervised semantic segmentation using out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16897–16906, 2022. 2, 7, 8

[23] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2643–2652, 2021. 1

[24] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel su-

pervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5495–5505, 2021. 3

[25] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018. 2

[26] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. *arXiv preprint arXiv:2012.05007*, 2020. 2, 3

[27] Yi Li, Yiqun Duan, Zhanghui Kuang, Yimin Chen, Wayne Zhang, and Xiaomeng Li. Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation. *arXiv preprint arXiv:2112.07431*, 2021. 3

[28] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6964–6973, 2021. 1, 3, 6, 7, 8

[29] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016. 1

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 6

[31] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1742–1750, 2015. 1

[32] Jie Qin, Jie Wu, Xuefeng Xiao, Lujun Li, and Xingang Wang. Activation modulation and recalibration scheme for weakly supervised semantic segmentation. *arXiv preprint arXiv:2112.08996*, 2021. 2, 7

[33] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. *arXiv preprint arXiv:2103.01795*, 2021. 2, 7

[34] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. *arXiv preprint arXiv:2007.01947*, 2020. 2, 3

[35] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7283–7292, 2021. 2, 6, 7

[36] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7158–7166, 2017. 1

[37] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020. 1, 2, 8

[38] Tong Wu, Guangyu Gao, Junshi Huang, Xiaolin Wei, Xiaoming Wei, and Chi Harold Liu. Adaptive spatial-bce loss for weakly supervised semantic segmentation. In *European Conference on Computer Vision*, pages 199–216. Springer, 2022. 8

[39] Tong Wu, Junshi Huang, Guangyu Gao, Xiaoming Wei, Xiaolin Wei, Xuan Luo, and Chi Harold Liu. Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16765–16774, 2021. 2, 7

[40] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019. 6

[41] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Cross language image matching for weakly supervised semantic segmentation. *arXiv preprint arXiv:2203.02668*, 2022. 7, 8

[42] Jinheng Xie, Jianfeng Xiang, Junliang Chen, Xianxu Hou, Xiaodong Zhao, and Linlin Shen. C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–998, 2022. 1, 2

[43] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6984–6993, 2021. 3

[44] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4310–4319, 2022. 6, 7, 8

[45] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2623–2632, 2021. 3

[46] Sung-Hoon Yoon, Hyeokjun Kweon, Jegyeong Cho, Shinjeong Kim, and Kuk-Jin Yoon. Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 326–344. Springer Nature Switzerland Cham, 2022. 1, 2, 6, 7, 8

[47] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end

weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12765–12772, 2020. 1

[48] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 2020. 7

[49] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7242–7251, 2021. 1, 2, 6, 8

[50] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018. 2, 6

[51] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1

[52] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4299–4309, 2022. 1, 2, 3