SFC: Shared Feature Calibration in Weakly Supervised Semantic Segmentation

Xinqiao Zhao[1,2*], Feilong Tang[1*], Xiaoyang Wang[1,2,3], Jimin Xiao[1†]

[1]Xi'an Jiaotong-Liverpool University
[2]University of Liverpool
[3]Metavisioncn
xqz@liverpool.ac.uk, Feilong.Tang19@xjtlu.edu.cn, wangxy@liverpool.ac.uk, Jimin.Xiao@xjtlu.edu.cn

Abstract

Image-level weakly supervised semantic segmentation has received increasing attention due to its low annotation cost. Existing methods mainly rely on **C**lass **A**ctivation **M**apping (CAM) to obtain pseudo-labels for training semantic segmentation models. In this work, we are the first to demonstrate that long-tailed distribution in training data can cause the CAM calculated through classifier weights over-activated for head classes and under-activated for tail classes due to the shared features among head- and tail- classes. This degrades pseudo-label quality and further influences final semantic segmentation performance. To address this issue, we propose a **S**hared **F**eature **C**alibration (SFC) method for CAM generation. Specifically, we leverage the class prototypes that carry positive shared features and propose a **M**ulti-**S**caled **D**istribution-**W**eighted (MSDW) consistency loss for narrowing the gap between the CAMs generated through classifier weights and class prototypes during training. The MSDW loss counterbalances over-activation and under-activation by calibrating the shared features in head-/tail-class classifier weights. Experimental results show that our SFC significantly improves CAM boundaries and achieves new state-of-the-art performances. The project is available at https://github.com/Barrett-python/SFC.

Introduction

Semantic segmentation (Minaee et al. 2021) assigns semantic labels to image pixels and is crucial for applications like autonomous driving, robotics (Zhang et al. 2022). Obtaining accurate pixel annotations for training deep learning models is laborious and time-consuming. One alternative approach is to adopt **W**eakly **S**upervised **S**emantic **S**egmentation (WSSS) with only image-level labels provided (Ahn, Cho, and Kwak 2019; Wang et al. 2020; Zhang et al. 2020; Lee, Kim, and Yoon 2021; Xu et al. 2022; Zhang et al. 2023a, 2021a). Generally, these methods employ **C**lass **A**ctivation **M**apping (CAM) (Zhou et al. 2016) to generate discriminative semantic masks from a classification model. Then, a series of post-processing methods (Krähenbühl and Koltun 2011) are adopted to refine the masks to obtain pixel-level pseudo labels, which are then used to train a semantic segmentation model (Chen et al. 2016).

However, we find the training data of WSSS are naturally long-tailed distributed (Fig. 1(a)), which makes the shared feature components (Li and Monga 2020) tend to be positive in head-class classifier weight and negative in tail-class classifier weight because the head-class weight receives more positive gradients (denoted as $\oplus$) than the negative ones (denoted as $\ominus$) and the tail-class weight receives more negative gradients than the positive ones (Fig. 1(b)). This makes the pixels containing shared features activated by the head-class classifier weight (*i.e.*, the dot product (denoted as $\cdot$) of feature and weight > 0) while the pixels containing tail-class feature not activated by the tail-class weight (*i.e.*, the dot product of feature and weight < 0) as shown in Fig. 1(c). Thus, the CAM calculated through classifier weights inevitably becomes over-activated for head classes and under-activated for tail classes (Fig. 1(d)). This degrades the qualities of pseudo labels and further influences the final WSSS performances. On the other hand, as shown in Fig. 1(d), the CAM activated by the head-class prototype (Chen et al. 2022a) is less-activated compared to the CAM activated by the head-class classifier weight, and the CAM activated by the tail-class prototype is more-activated compared to the CAM activated by the tail-class classifier weight.

Inspired by the above findings (a detailed theoretical analysis is provided in **Analysis On SFC** section of main paper), we propose a **S**hared **F**eature **C**alibration (SFC) method to reduce shared feature proportions in head-class classifier weights and increase the ones in tail-class classifier weights, avoiding shared-feature-caused over-/under-activation issues. Particularly, a **M**ulti-**S**caled **D**istribution-**W**eighted (MSDW) consistency loss is calculated on the CAMs generated through class prototypes and classifier weights, where the consistency loss magnitude on one class is re-weighted by the total sample number gaps between this class and other classes. The theories behind this re-weighting strategy is also demonstrated, proving that pseudo-labels with better boundaries can be achieved through our SFC. The contributions of this work include:

- We first point out that the features shared by head and tail classes can enlarge the classifier-weight-generated CAM for the head class and shrink it for the tail class under a long-tailed scenario.

[*]These authors contributed equally.
[†]Corresponding author.

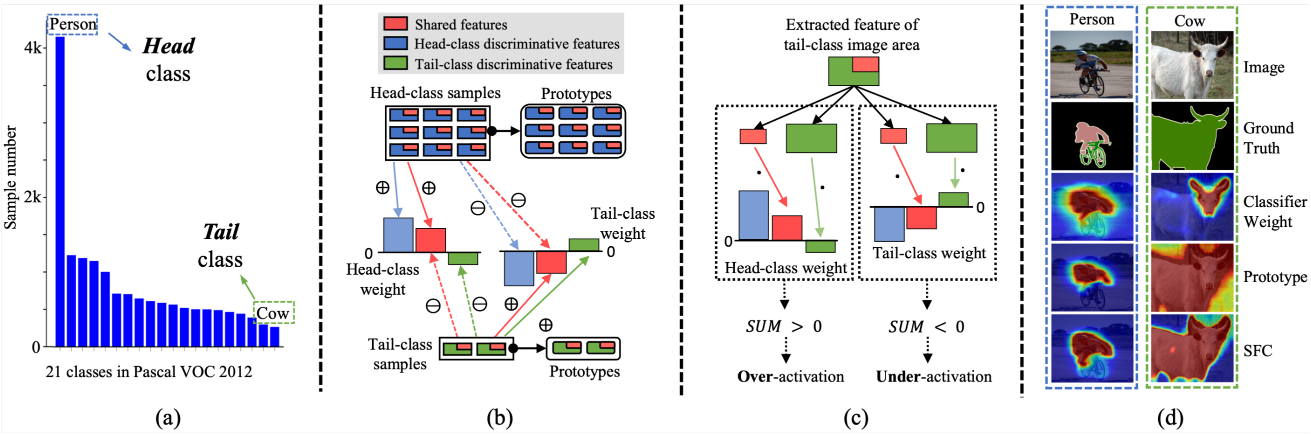

Figure 1: Illustration of how shared features influence CAMs under a long-tailed scenario and the effects of our proposed SFC. (a) shows Pascal VOC 2012 (Everingham et al. 2010) is a naturally long-tailed distributed dataset. (b) explains the shared feature components in head-/tail-class classifier weights and prototypes. (c) shows how over-/under-activations happen. (d) shows the CAMs of head-/tail-class examples. Our SFC achieves better results with appropriate activation areas.

- We propose a **S**hared **F**eature **C**alibration (SFC) CAM generation method aiming at balancing the shared feature proportions in different classifier weights, which can improve the CAM quality.
- Our method achieves new state-of-the-art WSSS performances with only image-level labels on Pascal VOC 2012 and COCO 2014.

Related Works

Weakly Supervised Semantic Segmentation

The generation of pseudo-labels in WSSS is based on attention mapping (Wang et al. 2020; Zhang et al. 2021a). The key step is the produce of high-quality CAMs (Sun et al. 2020; Yoon et al. 2022). Several works have designed heuristic approaches, such as erasing and accumulation (Zhang et al. 2021b; Yoon et al. 2022), to force the network to mine novel regions rather solely focusing on discriminative regions. Moreover, other strategies include self-supervised learning (Wang et al. 2020; Chen et al. 2022a), contrastive learning (Du et al. 2022), and cross-image information (Xu et al. 2023) have been proposed to generate accurate and complete CAMs. Recently, vision-language pre-training has emerged as the prevalent approach for addressing downstream vision-language tasks (Zhu et al. 2023), including WSSS (Lin et al. 2023). Due to the rough boundary of the initial map, refinement methods like CRF (Krähenbühl and Koltun 2011) and IRN (Ahn, Cho, and Kwak 2019) are employed for further enhancements. However, to the best of our knowledge, no previous work aims at solving the over-/under-activation issue caused by long-tailed distributed training data. This paper analyzes the reasons behind the over-/under-activation and tackles this issue through a **S**hared **F**eature **C**alibration (SFC) method.

Shared Feature in Classification

Classification is an upstream task for semantic segmentation (Zhang et al. 2023b) and shared feature has been actively studied in this task (Li and Monga 2020). Most existing methods (Zheng et al. 2017; Yao et al. 2017; Peng, He, and Zhao 2017) tend to only extract discriminative partial features for classification and prevent the shared features from influencing the classification performance. Although both training under classification loss, unlike the classification task, WSSS cannot solely rely on discriminative features to construct intact CAM, and existing methods (Lee, Kim, and Shim 2022; Chen et al. 2022a) freeze several layers of pre-trained encoder for avoiding catastrophic forgetting of indiscriminative features (Vasconcelos, Birodkar, and Dumoulin 2022). In this work, we focus on balancing the shared feature proportions in classifier weights under a long-tailed scenario for a better WSSS performance.

Methodology

The pipeline of our SFC is illustrated in Fig. 2. It involves an **I**mage **B**ank **R**e-sampling (IBR) and a **M**ulti-**S**caled **D**istribution-**W**eighted (MSDW) consistency loss.

Preliminary

Classifier Weight CAM Given an input image $\mathbf{I}$, the features extracted from $\mathbf{I}$ by the encoder is denoted as $\mathbf{F} \in \mathbb{R}^{H\times W\times D}$, and the classifier weight of class c is denoted as $\mathbf{W}_c \in \mathbb{R}^{D\times 1}$, where $H \times W$ is the feature map size and D is the feature dimension. The classification loss, which is a multi-label soft margin loss, is calculated as:

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathbf{C}|}\sum_{c=1}^{|\mathbf{C}|}\Big(\mathbf{y}_c \log\big(sigmoid(GAP(\mathbf{F}\mathbf{W}_c))\big) + (1-\mathbf{y}_c)\log\big(1-sigmoid(GAP(\mathbf{F}\mathbf{W}_c))\big)\Big), \tag{1}$$

where $\mathbf{C}$ is the foreground class set and $|\mathbf{C}|$ denotes its size; $\mathbf{y}_c$ denotes the binary label on class c; $GAP(\cdot)$ denotes the global average pooling. Then, the CAM generated through

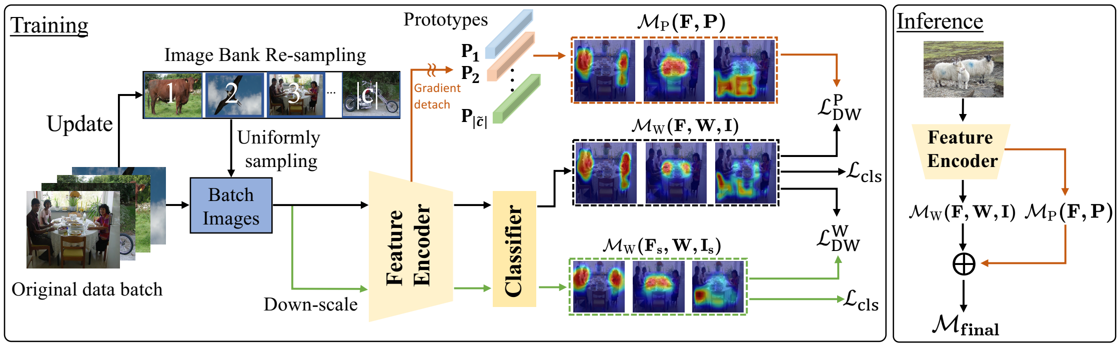

Figure 2: The overall structure of our proposed SFC. For each training image, two distribution-weighted consistency losses ($\mathcal{L}^{\mathrm{P}}_{\mathrm{DW}}$ and $\mathcal{L}^{\mathrm{W}}_{\mathrm{DW}}$) are calculated, where $\mathcal{L}^{\mathrm{P}}_{\mathrm{DW}}$ is calculated between the prototype CAM ($\mathcal{M}_{\mathrm{P}}$) and classifier weight CAM ($\mathcal{M}_{\mathrm{W}}$) of original image and $\mathcal{L}^{\mathrm{W}}_{\mathrm{DW}}$ is calculated between the classifier weight CAMs of down-scaled and original images. In addition, an image bank that stores the latest shown images for different classes is maintained, and images are uniformly sampled from it to complement the original training batch, increasing the consistency loss optimization frequency for tail classes. Finally, the classifier weight CAM is complemented with prototype CAM in inference.

classifier weights $\mathbf{W}$ (*i.e.*, classifier weight CAM) on the extracted feature $\mathbf{F}$ of input image $\mathbf{I}$ is calculated as follows:

$$\mathcal{M}_{\mathbf{W}}(\mathbf{F},\mathbf{W},\mathbf{I}) = \underbrace{\left\{f\left(\mathbf{F}\mathbf{W}_1,\mathbf{I},\mathbf{F}\right),...,f\left(\mathbf{F}\mathbf{W}_{|\mathbf{C}|},\mathbf{I},\mathbf{F}\right)\right\}}_{|\mathbf{C}|\ foreground\ classes} \cup \underbrace{\left(\mathbf{1}-\max_{c\in\mathbf{C}}\left(f(\mathbf{F}\mathbf{W}_c,\mathbf{I},\mathbf{F})\right)\right)}_{background\ class}, \tag{2}$$

where $\mathcal{M}_{\mathbf{W}}$ denotes the classifier weight CAM, $f(\cdot)$ denotes a function that feeds the normalized $ReLU$ activated $\mathbf{FW}$, $\mathbf{I}$, and $\mathbf{F}$ into a **P**ixel **C**orrelation **M**odule (PCM) (Wang et al. 2020) to refine the CAM based on the relationships among the low-level features of different pixels.

Prototype CAM Following (Chen et al. 2022a, 2023), class prototype is calculated through masked average pooling of extracted features. Specifically, the hierarchical features extracted from different layers of feature extractor are denoted as $\mathbf{F_1},\mathbf{F_2},\mathbf{F_3},\mathbf{F_4}$; $L(\cdot)$ denotes the linear projection and it stops gradients to the feature extractor. Then, the prototype $\mathbf{P}_{\tilde{c}}$ of class $\tilde{c}$ ($\tilde{c}$ can be either foreground or background class) is calculated as follows:

$$\mathbf{P}_{\tilde{c}} = MAP\Big(\big(\widehat{\mathcal{M}}_{\mathbf{W}}(\mathbf{F},\mathbf{W},\mathbf{I})\big)_{\tilde{c}} \odot L(\mathbf{F_1},\mathbf{F_2},\mathbf{F_3},\mathbf{F_4})\Big), \tag{3}$$

where $\big(\widehat{\mathcal{M}}_{\mathbf{W}}(\mathbf{F},\mathbf{W},\mathbf{I})\big)_{\tilde{c}}$ is a binary mask for class $\tilde{c}$, highlighting the pixels whose activation values are higher than the set threshold with 1; $MAP(\cdot)$ denotes masked average pooling. Finally, the CAM calculated through the prototype of class $\tilde{c}$ (*i.e.*, prototype CAM) is calculated as follows:

$$\big(\mathcal{M}_{\mathbf{P}}(\mathbf{F},\mathbf{P})\big)_{\tilde{c}} = ReLU\big(\cos\langle\mathbf{P}_{\tilde{c}}, L(\mathbf{F_1},\mathbf{F_2},\mathbf{F_3},\mathbf{F_4})\rangle\big), \tag{4}$$

where $\cos\langle\cdot,\cdot\rangle$ denotes the cosine similarity between the two terms within it.

Shared Feature Calibration

Image Bank Re-sampling (IBR) We maintain an image bank $\mathcal{B} = (\mathbf{b}_1,...,\mathbf{b}_c)$ that stores $|\mathbf{C}|$ images for $|\mathbf{C}|$ foreground classes. For each image $\mathbf{I}$ in the current training batch, we update $\mathbf{b}_c$ with $\mathbf{I}$ when the c-th class appears in $\mathbf{I}$. Otherwise, we keep $\mathbf{b}_c$ as it was. After the bank updating, we uniformly sample N_{IBR} images from the current bank and concatenate them with the original training batch as final training inputs. The uniform sampling does not bring further shared feature issues caused by long-tailed distribution, as the sample numbers of different classes are nearly balanced.

Our proposed IBR is used for increasing the tail-class sampling frequency, thus the MSDW loss will be enforced on the tail classes more frequently, effectively calibrating the shared features in the tail-class classifier weights.

Multi-Scaled Distribution-Weighted Consistency Loss To address the over-activation issues on head classes and under-activation issues on tail classes, we propose two **D**istribution-**W**eighted (DW) consistency losses $\mathcal{L}^{\mathrm{P}}_{\mathrm{DW}}$ and $\mathcal{L}^{\mathrm{W}}_{\mathrm{DW}}$. $\mathcal{L}^{\mathrm{P}}_{\mathrm{DW}}$ is calculated between the prototype CAM and classifier weight CAM as:

$$\mathcal{L}^{\mathrm{P}}_{\mathrm{DW}} = \sum_{c=1}^{|\mathbf{C}|}\Big(DC_c \cdot \underbrace{\big\|(\mathcal{M}_{\mathbf{W}})_c - (\mathcal{M}_{\mathbf{P}})_c\big\|_1}_{\ell 1\ loss\ of\ foreground\ class}\Big) + \underbrace{\big\|(\mathcal{M}_{\mathbf{W}})_{|\mathbf{C}|+1} - (\mathcal{M}_{\mathbf{P}})_{|\mathbf{C}|+1}\big\|_1}_{\ell 1\ loss\ of\ background\ class}, \tag{5}$$

where DC_c denotes the scaled **D**istribution **C**oefficient, calculated for each foreground class c as:

$$DC_c = \underbrace{\frac{|\mathbf{C}|}{\sum_{j=1}^{|\mathbf{C}|}\Big(\frac{\sum_{i=1}^{|\mathbf{C}|}|n_j - n_i|}{(n_j+\mathcal{N})}\Big)}}_{scaling\ factor} \cdot \underbrace{\frac{\sum_{i=1}^{|\mathbf{C}|}|n_c - n_i|}{n_c + \mathcal{N}}}_{total\ demand}, \tag{6}$$

where n_c denotes the sample number of class c; $\mathcal{N}$ denotes the estimated increased sample number of our IBR on each

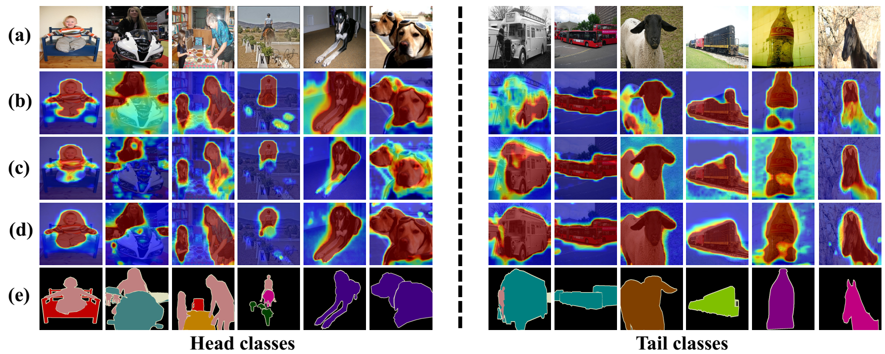

Figure 3: CAM visualization results on PASCAL VOC 2012, demonstrating **Conclusion 2** and **Conclusion 3**. (a) input images; (b) classifier weight CAMs; (c) prototype CAMs; (d) final CAMs generated through our SFC; (e) ground truth.

Method	PASCAL VOC		
	CAM	CRF	Mask
IRN (Ahn, Cho, and Kwak 2019)	48.8	54.3	66.3
SEAM (Wang et al. 2020)	55.4	56.8	63.6
CONTA (Zhang et al. 2020)	48.8	-	67.9
AdvCAM (Lee, Kim, and Yoon 2021)	55.6	62.1	68.0
RIB (Lee et al. 2021a)	56.5	62.9	70.6
CLIMS (Xie et al. 2022)	56.6	62.4	70.5
ESOL (Li et al. 2022)	53.6	61.4	68.7
SIPE (Chen et al. 2022a)	58.6	64.7	68.0
AMN (Lee, Kim, and Shim 2022)	62.1	65.3	72.2
SFC (Ours)	**64.7**	**69.4**	**73.7**

Table 1: Evaluation (mIoU (%)) of different pseudo labels on PASCAL VOC 2012 training set.

Method	Backbone	Val	Test
Image-level supervision + Saliency maps.			
AuxSegNet (Xu et al. 2021)	ResNet38	69.0	68.6
NSROM (Yao et al. 2021)	ResNet101	70.4	70.2
EPS (Lee et al. 2021b)	ResNet101	71.0	71.8
Image-level supervision only.			
SEAM (Wang et al. 2020)	ResNet38	64.5	65.7
PPC+SEAM (Du et al. 2022)	ResNet38	67.7	67.4
ReCAM (Chen et al. 2022b)	ResNet38	68.5	68.4
SIPE (Chen et al. 2022a)	ResNet38	68.2	69.5
SIPE (Chen et al. 2022a)	ResNet101	68.8	69.7
ESOL (Li et al. 2022)	ResNet101	69.9	69.3
AMN (Lee, Kim, and Shim 2022)	ResNet101	70.7	70.6
SFC (Ours)	ResNet38	**70.2**	**71.4**
SFC (Ours)	ResNet101	**71.2**	**72.5**

Table 2: Comparison of semantic segmentation performance on PASCAL VOC 2012 validation and test sets.

class and $\mathcal{N} = \frac{N_{\text{IBR}} \cdot C_{\text{IBR}} \cdot N_{\text{iter}}}{|\mathbf{C}|}$. Here, N_{iter} denotes the number of iterations in one training epoch; N_{IBR} is the sampling number from the image bank; C_{IBR} is the average number of classes covered in one image.

We calculate the sum of sample number gaps between class c and other classes and regard this sum as the *total demand* on the consistency loss for class c. Next, this total reward is averaged by $n_c + \mathcal{N}$ (*i.e.*, the estimated total sample number of class c) and scaled with the *scaling factor*, obtaining the scaled distribution coefficient (*i.e.*, DC_c). The *scaling factor* is to scale the $\ell 1$ loss magnitude of foreground class to the same level as the background class.

DC_c is finally used to re-weight the consistency loss on class c, assigning higher consistency loss to the class with higher total demand, as the severity of over-/under-activation issue is positively related to the *total demand*.

Meanwhile, all images in the current training batch are further down-scaled with 0.5 through bilinear interpolation algorithm and are used to calculate the loss $\mathcal{L}_{\text{DW}}^{\text{W}}$:

$$\begin{aligned}\mathcal{L}_{\text{DW}}^{\text{W}} = &\sum_{c=1}^{|\mathbf{C}|} \Bigg(DC_c \cdot \bigg\| s\Big(\big(\mathcal{M}_{\mathbf{W}}(\mathbf{F}, \mathbf{W}, \mathbf{I})\big)_c \Big) \\ &- \big(\mathcal{M}_{\mathbf{W}}(\mathbf{F}_s, \mathbf{W}, \mathbf{I}_s)\big)_c \bigg\|_1 \Bigg),\end{aligned} \tag{7}$$

where $s(\cdot)$ denotes the bilinear down-sampling operation; $\mathbf{I}_s$ denotes the down-scaled image and $\mathbf{F}_s$ denotes its extracted feature. Similar to Eq. (5), we re-weight the consistency loss by DC coefficient. Considering the prototype CAM on the down-scaled image is less accurate than the down-scaled classifier weight CAM calculated on the original image (see **$\mathcal{L}_{\text{DW}}^{\text{W}}$ with Multi-Scaled Scheme** in appendix), we calculate the consistency loss between the down-scaled classifier weight CAM on the original image and the classifier weight CAM on the down-scaled image. $\mathcal{L}_{\text{DW}}^{\text{W}}$ further boosts the performance improvement. Our multi-scaled distribution-weighted consistency loss $\mathcal{L}_{\text{MSDW}}$ is formulated as follows:

$$\mathcal{L}_{\text{MSDW}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{DW}}^{\text{P}} + \mathcal{L}_{\text{DW}}^{\text{W}}. \tag{8}$$

Inference The final CAM for inference is calculated as:

$$(\mathcal{M}_{\text{final}})_{\tilde{c}} = \big(\mathcal{M}_{\mathbf{W}}(\mathbf{F}, \mathbf{W}, \mathbf{I})\big)_{\tilde{c}} + \big(\mathcal{M}_{\mathbf{P}}(\mathbf{F}, \mathbf{P})\big)_{\tilde{c}}, \tag{9}$$

where $(\mathcal{M}_{\text{final}})_{\tilde{c}}$ denotes the final CAM of class $\tilde{c}$; $\tilde{c}$ can be foreground or background class. In this way, the classifier weight CAM is complemented by the prototype CAM, jointly solving the over-/under-activation issue.

Method	Backbone	Val
Image-level supervision + Saliency maps.		
G-WSSS (Li et al. 2021)	ResNet38	28.4
AuxSegNet (Xu et al. 2021)	ResNet38	33.9
EPS (Lee et al. 2021b)	ResNet101	35.7
Image-level supervision only.		
IRN (Ahn, Cho, and Kwak 2019)	ResNet50	32.6
SEAM (Wang et al. 2020)	ResNet38	31.9
RIB (Lee et al. 2021a)	ResNet38	43.8
SIPE (Chen et al. 2022a)	ResNet38	43.6
CONTA (Zhang et al. 2020)	ResNet101	33.4
SIPE (Chen et al. 2022a)	ResNet101	40.6
ESOL (Li et al. 2022)	ResNet101	42.6
AMN (Lee, Kim, and Shim 2022)	ResNet101	44.7
SFC (Ours)	ResNet101	**46.8**

Table 3: Comparison of semantic segmentation performance on MS COCO 2014 validation set.

Analysis on SFC

This section demonstrates how shared features in classifier weights cause over-/under-activation issues under a long-tailed scenario and the working mechanism behind SFC.

Shared Feature Distribution in Classifier Weights

In image-level WSSS, a multi-label soft margin loss (denoted as $\mathcal{L}$) is commonly used for the classification model training (Chen et al. 2022a). Following the definitions in (Tan et al. 2021), the positive and negative gradients caused by $\mathcal{L}$ are formulated as follows:

$$\begin{aligned}(\mathcal{L}_c'^{pos})_i &= -\,\mathbf{y}_{i,c}\big(sigmoid(\mathbf{z}_{i,c})-1\big)>0, \quad \mathbf{y}_{i,c}=1\\ (\mathcal{L}_c'^{neg})_i &= -\,(1-\mathbf{y}_{i,c})sigmoid(\mathbf{z}_{i,c})<0, \quad \mathbf{y}_{i,c}=0,\end{aligned} \tag{10}$$

where $\mathbf{z}_{i,c}$ denotes the model predicted logit for the i-th sample on class c, and $sigmoid(\mathbf{z}_{i,c})$ indicates its sigmoid activated value; $\mathbf{y}_{i,c}$ denotes the label of the i-th sample on class c (either 0 or 1); $(\mathcal{L}_c'^{pos})_i$ and $(\mathcal{L}_c'^{neg})_i$ denote the positive and negative gradients on $\mathbf{z}_{i,c}$.

Considering a simplified case with one head class H and one tail class T, based on the conclusion that different classes have shared features (Hou, Yu, and Tao 2022; Liu et al. 2021; Li and Monga 2020), the head-class image feature can be decomposed as $\alpha_H\mathbf{f}_H+\eta_H\mathbf{f}_0$, where $\alpha_H\mathbf{f}_H$ and $\eta_H\mathbf{f}_0$ indicate the discriminative and shared feature parts respectively, with α_H and η_H indicating their proportions. Similarly, the tail-class image feature can be decomposed as $\alpha_T\mathbf{f}_T+\eta_T\mathbf{f}_0$. Then, the head-class classifier weight $\mathbf{W}_H$ can be represented as:

$$\begin{aligned}\mathbf{W}_H =& n_H E(\alpha_H)E(\mathcal{L}_H'^{pos})\mathbf{f}_H + n_T E(\alpha_T)E(\mathcal{L}_H'^{neg})\mathbf{f}_T\\ &+\underbrace{\big(n_H E(\eta_H)E(\mathcal{L}_H'^{pos})+n_T E(\eta_T)E(\mathcal{L}_H'^{neg})\big)\mathbf{f}_0}_{\mathbf{f}_0^H:\ shared\ feature\ component\ in\ \mathbf{W}_H},\end{aligned} \tag{11}$$

where n_H and n_T indicate the sample numbers of head and tail classes, with $n_H \gg n_T$ under a long-tailed scenario; $E(\cdot)$ denotes the expectation operation. Similarly, we have the tail-class classifier weight $\mathbf{W}_T$ as:

$$\begin{aligned}\mathbf{W}_T =& n_T E(\alpha_T)E(\mathcal{L}_T'^{pos})\mathbf{f}_T + n_H E(\alpha_H)E(\mathcal{L}_T'^{neg})\mathbf{f}_H\\ &+\underbrace{\big(n_T E(\eta_T)E(\mathcal{L}_T'^{pos})+n_H E(\eta_H)E(\mathcal{L}_T'^{neg})\big)\mathbf{f}_0}_{\mathbf{f}_0^T:\ shared\ feature\ component\ in\ \mathbf{W}_T}.\end{aligned} \tag{12}$$

The proofs of Eq. (11) and Eq. (12) are provided in **Proof 1** of appendix.

Then, as demonstrated in **Gradient Magnitude Analysis** of appendix, the magnitude of $E(\mathcal{L}'^{pos})$ is larger than that of $E(\mathcal{L}'^{neg})$ and the gap is not significant. Combining with the precondition that $n_H \gg n_T$, it can be concluded that in Eq. (11) we have $n_H E(\mathcal{L}_H'^{pos}) + n_T E(\mathcal{L}_H'^{neg}) > 0$. Similarly, in Eq. (12) we have $n_T E(\mathcal{L}_T'^{pos}) + n_H E(\mathcal{L}_T'^{neg}) < 0$. Then, based on Eq. (11) and Eq. (12), we can have:

Conclusion 1. *When $n_H \gg n_T$ and the difference between $E(\eta_H)$ and $E(\eta_T)$ is not as significant as that between n_H and n_T, the shared feature component in $\mathbf{W}_H$* (i.e., $\mathbf{f}_0^H$) *tends to be positive with a large magnitude, and the one in $\mathbf{W}_T$* (i.e., $\mathbf{f}_0^T$) *tends to be negative with a large magnitude. However, when $n_H \approx n_T$, the class with a higher $E(\eta)$ will have a larger shared feature in its classifier weight, and the shared feature magnitude will be much lower than that when $n_H \gg n_T$.*

Over-activation and Under-activation

One extracted feature of tail-class image area can be decomposed as $\mathbf{A}_T = \alpha_T^A\mathbf{f}_T + \eta_T^A\mathbf{f}_0$ (α_T^A and η_T^A are the proportions of $\mathbf{f}_T$ and $\mathbf{f}_0$). Similarly, one extracted feature of head-class image area can be decomposed as $\mathbf{A}_H = \alpha_H^A\mathbf{f}_H + \eta_H^A\mathbf{f}_0$. Under long-tailed scenario (*i.e.*, $n_H \gg n_T$), the head-/tail-class classifier weight activations on $\mathbf{A}_T$ and $\mathbf{A}_H$ can be formulated as follows:

$$\begin{aligned}&\mathbf{A}_T\mathbf{W}_H\\ =&\big(n_H E(\eta_H)E(\mathcal{L}_H'^{pos})+n_T E(\eta_T)E(\mathcal{L}_H'^{neg})\big)\eta_T^A\left\|\mathbf{f}_0\right\|_2^2\\ &+n_T\alpha_T^A E(\alpha_T)E(\mathcal{L}_H'^{neg})\left\|\mathbf{f}_T\right\|_2^2,\end{aligned} \tag{13}$$

$$\begin{aligned}&\mathbf{A}_H\mathbf{W}_H\\ =&\big(n_H E(\eta_H)E(\mathcal{L}_H'^{pos})+n_T E(\eta_T)E(\mathcal{L}_H'^{neg})\big)\eta_H^A\left\|\mathbf{f}_0\right\|_2^2\\ &+n_H\alpha_H^A E(\alpha_H)E(\mathcal{L}_H'^{pos})\left\|\mathbf{f}_H\right\|_2^2,\end{aligned} \tag{14}$$

	IBR	$\mathcal{L}_{\text{DW}}^{\text{P}}$	$\mathcal{L}_{\text{DW}}^{\text{W}}$	mIoU (%)
Base				55.1
I	✓			55.9
II		✓	✓	62.4
III	✓	✓		62.4
IV	✓		✓	58.1
V	✓	✓	✓	64.7

Table 4: Ablation of SFC components. Base and I report the mIoU of $\mathcal{M}_{\mathbf{W}}$ and others report the mIoU of $\mathcal{M}_{\text{final}}$.

	$\mathcal{L}_{\text{DW}}^{\text{P}}$	$\mathcal{L}_{\text{DW}}^{\text{W}}$	mIoU (%)
VI			62.0
VII	✓		63.7
VIII		✓	63.6
V	✓	✓	64.7

Table 5: Effect of DC. ✓: the presence of DC. VI: 'Base': plain consistency loss. mIoU of $\mathcal{M}_{\text{final}}$ is reported.

$$\begin{aligned}&\mathbf{A}_T\mathbf{W}_T\\ =&\big(n_T E(\eta_T)E(\mathcal{L}_T'^{pos})+n_H E(\eta_H)E(\mathcal{L}_T'^{neg})\big)\eta_T^A\left\|\mathbf{f}_0\right\|_2^2\\ &+n_T\alpha_T^A E(\alpha_T)E(\mathcal{L}_T'^{pos})\left\|\mathbf{f}_T\right\|_2^2,\end{aligned} \tag{15}$$

$$\begin{aligned}&\mathbf{A}_H\mathbf{W}_T\\ =&\big(n_T E(\eta_T)E(\mathcal{L}_T'^{pos})+n_H E(\eta_H)E(\mathcal{L}_T'^{neg})\big)\eta_H^A\left\|\mathbf{f}_0\right\|_2^2\\ &+n_H\alpha_H^A E(\alpha_H)E(\mathcal{L}_T'^{neg})\left\|\mathbf{f}_H\right\|_2^2.\end{aligned} \tag{16}$$

Based on **Conclusion 1**, proved through **Proof 2** in appendix, we have **Conclusion 2**:

Conclusion 2. *When $n_H \gg n_T$, $\mathbf{A}_H\mathbf{W}_T$ and $\mathbf{A}_T\mathbf{W}_T$ tend to be unactivated, and $\mathbf{W}_T$ has* ***under-activated*** *tail-class image area compared with the ground truth (as shown in the tail classes of Fig. 3(b)). On the contrary, $\mathbf{A}_H\mathbf{W}_H$ and $\mathbf{A}_T\mathbf{W}_H$ tend to be activated, and $\mathbf{W}_H$ has* ***over-activated*** *head-class image area compared with the ground truth (as shown in the head classes of Fig. 3(b)).*

On the other hand, for the class prototype extracted through averaging its classifier weight activated features, it only has positive shared features. Thus, based on **Conclusion 2**, proved through **Proof 3** in appendix, we have **Conclusion 3**:

Conclusion 3. *Let $\mathbf{P}_H$ and $\mathbf{P}_T$ denote the prototypes of head class H and tail class T, $\mathbf{A}$ denotes the image area including $\mathbf{A}_H$ and $\mathbf{A}_T$. When $n_H \gg n_T$, $\mathbf{AP}_H$ is* ***less-activated*** *compared with $\mathbf{AW}_H$ (as shown in the head classes of Fig. 3(b) and Fig. 3(c)). On the contrary, $\mathbf{AP}_T$ is* ***more-activated*** *compared with $\mathbf{AW}_T$ (as shown in the tail classes of Fig. 3(b) and Fig. 3(c)).*

How SFC Works

As described in Eq. (5), DW consistency loss pulls closer prototype CAM and classifier weight CAM for pairs: $\{\mathbf{A}_T\mathbf{W}_H, \mathbf{A}_T\mathbf{P}_H\}$, $\{\mathbf{A}_T\mathbf{W}_T, \mathbf{A}_T\mathbf{P}_T\}$, $\{\mathbf{A}_H\mathbf{W}_H, \mathbf{A}_H\mathbf{P}_H\}$, $\{\mathbf{A}_H\mathbf{W}_T, \mathbf{A}_H\mathbf{P}_T\}$. Thereby, $\mathbf{W}_H$ or $\mathbf{W}_T$ are enforced to learn towards features activated by $\mathbf{P}_H$ or $\mathbf{P}_T$. When $n_H \gg n_T$, based on **Conclusion 3**, we have:

CASE 1: For $\{\mathbf{A}_T\mathbf{W}_H, \mathbf{A}_T\mathbf{P}_H\}$, $\mathbf{A}_T\mathbf{P}_H$ is **less-activated** compared with $\mathbf{A}_T\mathbf{W}_H$. As $\mathbf{A}_T$ contains $\mathbf{f}_0$ and $\mathbf{f}_T$, $\mathbf{W}_H$ is optimized towards $-\mathbf{f}_0$ and $-\mathbf{f}_T$, bringing *positive* effects for $\mathbf{W}_H$ to shrink its CAM on tail-class areas.

CASE 2: For $\{\mathbf{A}_T\mathbf{W}_T, \mathbf{A}_T\mathbf{P}_T\}$, $\mathbf{A}_T\mathbf{P}_T$ is **more-activated** compared with $\mathbf{A}_T\mathbf{W}_T$. Since $\mathbf{A}_T$ contains $\mathbf{f}_0$ and $\mathbf{f}_T$, $\mathbf{W}_T$ is optimized towards $\mathbf{f}_0$ and $\mathbf{f}_T$, bringing *positive* effects for $\mathbf{W}_T$ to expand its CAM on tail-class areas.

CASE 3: For $\{\mathbf{A}_H\mathbf{W}_T, \mathbf{A}_H\mathbf{P}_T\}$, $\mathbf{A}_H\mathbf{P}_T$ is **more-activated** compared with $\mathbf{A}_H\mathbf{W}_T$. Since $\mathbf{A}_H$ contains $\mathbf{f}_0$ and $\mathbf{f}_H$, $\mathbf{W}_T$ is optimized towards $\mathbf{f}_0$ and $\mathbf{f}_H$. As $\mathbf{W}_T$ has $-\mathbf{f}_0$ and $-\mathbf{f}_H$ with large magnitudes (**Conclusion 1**), optimizing $\mathbf{W}_T$ towards positive $\mathbf{f}_0$ and $\mathbf{f}_H$ hardly brings negative effects.

CASE 4: For $\{\mathbf{A}_H\mathbf{W}_H, \mathbf{A}_H\mathbf{P}_H\}$, $\mathbf{A}_H\mathbf{P}_H$ is **less-activated** compared with $\mathbf{A}_H\mathbf{W}_H$. Since $\mathbf{A}_H$ contains $\mathbf{f}_0$ and $\mathbf{f}_H$, $\mathbf{W}_H$ is optimized towards $-\mathbf{f}_0$ and $-\mathbf{f}_H$. As $\mathbf{W}_H$ has $\mathbf{f}_0$ and $\mathbf{f}_H$ with large magnitudes (**Conclusion 1**), optimizing $\mathbf{W}_H$ towards $-\mathbf{f}_0$ and $-\mathbf{f}_H$ hardly brings negative effects.

In summary, classifier weights with severe over-/under-activation issues can benefit from **CASE 1** and **CASE 2**, while they are not negatively affected for **CASE 3** and **CASE 4**, improving the overall CAMs as shown in Fig. 3(d).

However, when $n_H \approx n_T$, the consistency *negatively* affects the CAM generation. For example, by pulling closer the pair of $\{\mathbf{A}_H\mathbf{W}_T, \mathbf{A}_H\mathbf{P}_T\}$, as $\mathbf{P}_T$ contains $\mathbf{f}_0$ and it activates $\mathbf{A}_H$ which contains $\mathbf{f}_0$ and $\mathbf{f}_H$, $\mathbf{W}_T$ will be optimized towards $\mathbf{f}_0$ and $\mathbf{f}_H$. However, $\mathbf{W}_T$ does not have $-\mathbf{f}_0$ or $-\mathbf{f}_H$ with large magnitude when $n_H \approx n_T$ (**Conclusion 1**), increasing $\mathbf{f}_H$ and $\mathbf{f}_0$ in $\mathbf{W}_T$ brings a *negative* effect.

Considering the consistency loss brings *positive* effects when $n_H \gg n_T$ and brings *negative* effects when $n_H \approx n_T$, we define the total demand on the consistency loss for each class by adding up the sample number gaps between this class and all other classes, and then regard this total demand as the weight of consistency loss on this class (*i.e.*, DC coefficient in Eq. (5)), maximizing the consistency loss effect.

	Classifier weight	Prototype	mIou (%)
IX	✓		64.2
X		✓	62.5
V	✓	✓	64.7

Table 6: IX: Inference with $\mathcal{M}_{\mathbf{W}}$. X: Inference with $\mathcal{M}_{\mathbf{P}}$. V: Inference with $\mathcal{M}_{\text{final}}$.

Experiments

Dataset and Evaluation Metric. Experiments are conducted on two benchmarks: PASCAL VOC 2012 (Everingham et al. 2010) with 21 classes and MS COCO 2014 (Lin et al. 2014) with 81 classes. For PASCAL VOC 2012, following (Wang et al. 2020; Lee, Kim, and Yoon 2021; Chen et al. 2022a; Li et al. 2022), we use the augmented SBD set (Hariharan et al. 2011) with 10,582 annotated images. **M**ean **I**ntersection **o**ver **U**nion (mIoU) (Long, Shelhamer, and Darrell 2015) is used to evaluate segmentation results.

Implementation Details. For pseudo label generation, we adopt the ImageNet (Deng et al. 2009) pretrained ResNet50 (He et al. 2016). Random cropping size 512×512 is adopted for training data augmentation. The N_{IBR} is set to 4. $\mathcal{M}_{final}$ from our method is further post-processed by DenceCRF (Krähenbühl and Koltun 2011) and IRN (Ahn, Cho, and

Class sets	Overall	Many	Medium	Few
VI	6.9	5.6	6.5	8.6
V	9.6	9.5	6.5	12.9

Table 7: Average mIoU gains of different class sets. The class sets definitions follow (Wu et al. 2020).

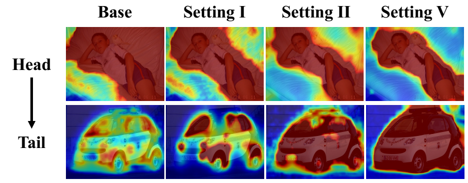

Figure 4: CAM visualization results under different settings.

Kwak 2019) to generate the final pseudo labels, which are used to train the segmentation model: ResNet101-based DeepLabV2 (Ahn, Cho, and Kwak 2019; Chen et al. 2022a). More details can be found in appendix.

Comparison of Pseudo-label Quality

To validate the effectiveness of our SFC, we evaluate the quality of intermediate and final results in the pseudo-label generation process in Table 1. Specifically, we first compare the initial CAM generated by the classification model (denoted as CAM). Then, we compare various post-processed CAMs to show the consistent improvements by our SFC. Particularly, the original CAM is first refined by CRF (Krähenbühl and Koltun 2011)(denoted as CRF) and further processed by IRN (Ahn, Cho, and Kwak 2019) to generate the final pseudo masks (denoted as Mask). Experimental results in Table 1 show that the CAM from SFC is significantly better than the previous works on datasets with different class numbers and long-tailed degrees, and our method outperforms state-of-the-art methods by 2.6% on PASCAL VOC. Regarding the CRF-post-processed CAM, we achieve 69.4% mIoU on the PASCAL VOC, and further with IRN, our SFC improves the mIoU to 73.7%, achieving 1.5% gain compared to AMN (Lee, Kim, and Shim 2022).

Comparison of WSSS Performance

In WSSS, the CRF and IRN post-processed pseudo masks obtained from the initial CAM are treated as ground truth to train semantic segmentation model in a fully supervised manner. Table 2 reports the mIoU scores of our method and recent WSSS methods on the validation and test sets of PASCAL VOC 2012. On this dataset, we achieve 71.2% and 72.5% mIoU using an ImageNet pre-trained backbone, outperforming all other WSSS methods that use only image-level labels or both image-level labels and saliency maps (Xu et al. 2021; Yao et al. 2021; Lee et al. 2021b). Table 3 reports the performance comparison on MS COCO 2014. Using the same training scheme as the PASCAL VOC 2012 experiment, our method achieves 46.8% mIoU on the validation set with a ResNet101 backbone, outperforming AMN (Lee, Kim, and Shim 2022) by 2.1%.

Ablation Studies

In Table 4, we first verify the effectiveness of SFC components, *i.e.*, Image Bank Re-sampling (IBR) and Multi-scaled Distribution-Weighted (MSDW) consistency Loss (including $\mathcal{L}_{\mathrm{DW}}^{\mathrm{P}}$ and $\mathcal{L}_{\mathrm{DW}}^{\mathrm{W}}$). 'Base' is the classifier weight CAM in Eq.(2). In Setting I, IBR increases the mIoU of 'Base' CAM by 0.8%, showing increasing the tail-class sampling frequency can alleviate the over-/under-activation issues. In Setting II, the CAM generated by SFC without IBR has a lower mIoU score than SFC (Setting V) by 2.3%, showing increasing the tail-class sampling frequency can boost the effectiveness of $\mathcal{L}_{\mathrm{MSDW}}$. The result of setting III shows $\mathcal{L}_{\mathrm{DW}}^{\mathrm{W}}$ can boost the performance improvement brought by $\mathcal{L}_{\mathrm{DW}}^{\mathrm{P}}$. However, setting IV indicates that using $\mathcal{L}_{\mathrm{DW}}^{\mathrm{W}}$ alone fails to calibrate the shared features in the down-scaled feature space and its performance drops significantly. Table 5 studies the effectiveness of DC in Eq. (6). Setting VI shows the SFC performance without DC in both of $\mathcal{L}_{\mathrm{DW}}^{\mathrm{P}}$ and $\mathcal{L}_{\mathrm{DW}}^{\mathrm{W}}$. Settings VII and VIII show the performances of removing DC only from $\mathcal{L}_{\mathrm{DW}}^{\mathrm{W}}$ or $\mathcal{L}_{\mathrm{DW}}^{\mathrm{P}}$. The results show that DC effectively adjusts the consistency loss weights of each class, bringing significant improvement.

Table 6 shows that the CAM combination $\mathcal{M}_{\mathrm{final}}$ in Eq. (9) (setting V) has the highest performance compared with those using $\mathcal{M}_{\mathbf{W}}$ or $\mathcal{M}_{\mathbf{P}}$ alone during inference, demonstrating it is better to complement $\mathcal{M}_{\mathbf{P}}$ to $\mathcal{M}_{\mathbf{W}}$ for SFC. Table 7 shows the average performance gains on different class sets with or without our DC coefficient. The plain consistency loss (setting VI) achieves almost the same gains across 'Many', 'Medium', and 'Few' classes. However, head and tail classes (*i.e.*, 'Many' and 'Few' classes) actually need higher magnitudes of consistency loss to overcome the over-/under-activation issues. With the help of DC coefficient (setting V), head and tail classes achieve more mIoU gains.

Besides, we also study the qualitative effects of SFC components in Fig. 4. It can be seen that Base with IBR (setting I) improves the CAM as it increases the tail-class sampling frequency and calibrates the shared features in classifier weights. However, the improvements are limited (*e.g.*, the shared feature 'wheel' is not activated) as IBR cannot balance the training data completely. When only using $\mathcal{L}_{\mathrm{MSDW}}$ (setting II), the CAMs are improved significantly but still not perfect, as the tail-class sampling frequencies and optimizing frequency of $\mathcal{L}_{\mathrm{MSDW}}$ are low. By using the complete SFC (setting V), we can achieve decent CAM results.

Conclusion

In this paper, we first demonstrate that shared features can cause over-/under-activation issues in CAM generation under a long-tailed scenario and then propose a novel Shared Feature Calibration (SFC) method for solving such issues, achieving new state-of-the-art performances. Our work provides a new perspective for improving CAM accuracy in image-level weakly supervised semantic segmentation, and other possible solutions will be investigated in future work.

Acknowledgements

This work was supported by the National Key R&D Program of China (No.2022YFE0200300), the National Natural Science Foundation of China (No. 61972323, 62331003), Suzhou Basic Research Program (SYG202316) and XJTLU REF-22-01-010, XJTLU AI University Research Centre, Jiangsu Province Engineering Research Centre of Data Science and Cognitive Computation at XJTLU and SIP AI innovation platform (YZCXPT2022103), Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering (SZS2022004).

References

Ahn, J.; Cho, S.; and Kwak, S. 2019. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*.

Chen, J.; Cong, R.; Yuxuan, L.; Ip, H.; and Kwong, S. 2023. Saving 100x Storage: Prototype Replay for Reconstructing Training Sample Distribution in Class-Incremental Semantic Segmentation. In *NeurIPS*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K. P.; and Yuille, A. L. 2016. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *PAMI*, 40: 834–848.

Chen, Q.; Yang, L.; Lai, J.-H.; and Xie, X. 2022a. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In *CVPR*.

Chen, Z.; Wang, T.; Wu, X.; Hua, X.-S.; Zhang, H.; and Sun, Q. 2022b. Class re-activation maps for weakly-supervised semantic segmentation. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Du, Y.; Fu, Z.; Liu, Q.; and Wang, Y. 2022. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV*, 88: 303–338.

Hariharan, B.; Arbeláez, P.; Bourdev, L.; Maji, S.; and Malik, J. 2011. Semantic contours from inverse detectors. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hou, Z.; Yu, B.; and Tao, D. 2022. Batchformer: Learning to explore sample relationships for robust representation learning. In *CVPR*.

Krähenbühl, P.; and Koltun, V. 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*.

Lee, J.; Choi, J.; Mok, J.; and Yoon, S. 2021a. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*.

Lee, J.; Kim, E.; and Yoon, S. 2021. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*.

Lee, M.; Kim, D.; and Shim, H. 2022. Threshold matters in WSSS: manipulating the activation for the robust and accurate segmentation model against thresholds. In *CVPR*.

Lee, S.; Lee, M.; Lee, J.; and Shim, H. 2021b. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*.

Li, J.; Jie, Z.; Wang, X.; Wei, X.; and Ma, L. 2022. Expansion and shrinkage of localization for weakly-supervised semantic segmentation. In *NeurIPS*.

Li, X.; and Monga, V. 2020. Group based deep shared feature learning for fine-grained image classification. *arXiv preprint arXiv:2004.01817*.

Li, X.; Zhou, T.; Li, J.; Zhou, Y.; and Zhang, Z. 2021. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Lin, Y.; Chen, M.; Wang, W.; Wu, B.; Li, K.; Lin, B.; Liu, H.; and He, X. 2023. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *CVPR*.

Liu, F.; Gao, C.; Sun, Y.; Zhao, Y.; Yang, F.; Qin, A.; and Meng, D. 2021. Infrared and Visible Cross-Modal Image Retrieval Through Shared Features. *TCSVT*, 31: 4485–4496.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; and Terzopoulos, D. 2021. Image segmentation using deep learning: A survey. *PAMI*, 44(7): 3523–3542.

Peng, Y.; He, X.; and Zhao, J. 2017. Object-part attention model for fine-grained image classification. *TIP*, 27(3): 1487–1500.

Sun, G.; Wang, W.; Dai, J.; and Van Gool, L. 2020. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*.

Tan, J.; Lu, X.; Zhang, G.; Yin, C.; and Li, Q. 2021. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *CVPR*.

Vasconcelos, C.; Birodkar, V.; and Dumoulin, V. 2022. Proper reuse of image classification features improves object detection. In *CVPR*, 13628–13637.

Wang, Y.; Zhang, J.; Kan, M.; Shan, S.; and Chen, X. 2020. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*.

Wu, T.; Huang, Q.; Liu, Z.; Wang, Y.; and Lin, D. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*.

Xie, J.; Hou, X.; Ye, K.; and Shen, L. 2022. CLIMS: cross language image matching for weakly supervised semantic segmentation. In *CVPR*.

Xu, L.; Ouyang, W.; Bennamoun, M.; Boussaid, F.; Sohel, F.; and Xu, D. 2021. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *CVPR*.

Xu, L.; Ouyang, W.; Bennamoun, M.; Boussaid, F.; and Xu, D. 2022. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*.

Xu, R.; Wang, C.; Sun, J.; Xu, S.; Meng, W.; and Zhang, X. 2023. Self correspondence distillation for end-to-end weakly-supervised semantic segmentation. In *AAAI*.

Yao, H.; Zhang, S.; Yan, C.; Zhang, Y.; Li, J.; and Tian, Q. 2017. AutoBD: Automated bi-level description for scalable fine-grained visual categorization. *TIP*, 27(1): 10–23.

Yao, Y.; Chen, T.; Xie, G.-S.; Zhang, C.; Shen, F.; Wu, Q.; Tang, Z.; and Zhang, J. 2021. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*.

Yoon, S.-H.; Kweon, H.; Cho, J.; Kim, S.; and Yoon, K.-J. 2022. Adversarial erasing framework via triplet with gated pyramid pooling layer for weakly supervised semantic segmentation. In *ECCV*.

Zhang, B.; Xiao, J.; Jiao, J.; Wei, Y.; and Zhao, Y. 2021a. Affinity attention graph neural network for weakly supervised semantic segmentation. *PAMI*, 44(11): 8082–8096.

Zhang, B.; Xiao, J.; Wei, Y.; and Zhao, Y. 2023a. Credible Dual-Expert Learning for Weakly Supervised Semantic Segmentation. *IJCV*, 131: 1892 – 1908.

Zhang, D.; Zhang, H.; Tang, J.; Hua, X.-S.; and Sun, Q. 2020. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*.

Zhang, F.; Gu, C.; Zhang, C.; and Dai, Y. 2021b. Complementary patch for weakly supervised semantic segmentation. In *ICCV*.

Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2023b. SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. *arXiv preprint arXiv:2303.05118*.

Zhang, Z.; Gao, G.; Fang, Z.; Jiao, J.; and Wei, Y. 2022. Mining Unseen Classes via Regional Objectness: A Simple Baseline for Incremental Segmentation. *NeurIPS*, 35: 24340–24353.

Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *ICCV*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*.

Zhu, H.; Wei, Y.; Liang, X.; Zhang, C.; and Zhao, Y. 2023. CTP: Towards Vision-Language Continual Pretraining via Compatible Momentum Contrast and Topology Preservation. In *ICCV*, 22257–22267.