Hunting Attributes: Context Prototype-Aware Learning for Weakly Supervised Semantic Segmentation

Feilong Tang[1,2*] Zhongxing Xu[3*] Zhaojun Qu[4] Wei Feng[1,2]
Xingjian Jiang[5] Zongyuan Ge[1,2†]

[1]AIM Lab, Faculty of IT, Monash University [2]Faculty of IT, Monash University
[3]Weill Cornell Medicine, Cornell University [4]Xi'an Jiaotong-Liverpool University
[5]Ann Arbor, University of Michigan
{feilong.tang,zongyuan.ge}@monash.edu

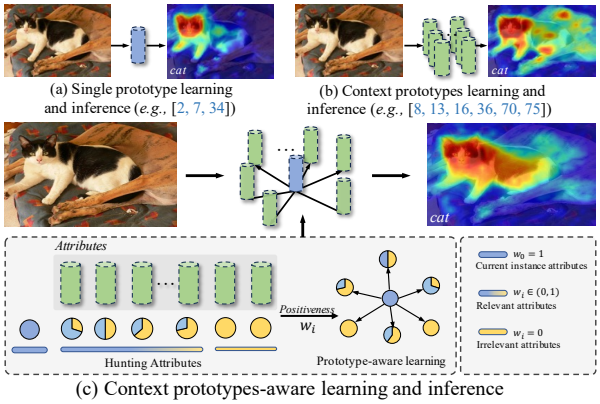
Abstract

Recent weakly supervised semantic segmentation (WSSS) methods strive to incorporate contextual knowledge to improve the completeness of class activation maps (CAM). In this work, we argue that the knowledge bias between instances and contexts affects the capability of the prototype to sufficiently understand instance semantics. Inspired by prototype learning theory, we propose leveraging prototype awareness to capture diverse and fine-grained feature attributes of instances. The hypothesis is that contextual prototypes might erroneously activate similar and frequently co-occurring object categories due to this knowledge bias. Therefore, we propose to enhance the prototype representation ability by mitigating the bias to better capture spatial coverage in semantic object regions. With this goal, we present a Context Prototype-Aware Learning (CPAL) strategy, which leverages semantic context to enrich instance comprehension. The core of this method is to accurately capture intra-class variations in object features through context-aware prototypes, facilitating the adaptation to the semantic attributes of various instances. We design feature distribution alignment to optimize prototype awareness, aligning instance feature distributions with dense features. In addition, a unified training framework is proposed to combine label-guided classification supervision and prototypes-guided self-supervision. Experimental results on PASCAL VOC 2012 and MS COCO 2014 show that CPAL significantly improves off-the-shelf methods and achieves state-of-the-art performance. The project is available at https://github.com/Barrett-python/CPAL.

1. Introduction

Semantic segmentation serves as a fundamental task in the field of computer vision. Weakly Supervised Semantic Segmentation (WSSS) has become a popular approach in the community, learning from weak labels such as image-level labels [25, 30], scribbles [37, 56], or bounding boxes [11, 31, 49], instead of pixel-level annotations. Most WSSS approaches utilize Class Activation Mapping (CAM) [74] to provide localization cues for target objects, thereby mapping visual concepts to pixel regions.

The key in WSSS is generating CAM with better coverage on the complete object. Recent studies [3, 51, 59, 71] primarily aim to optimize model segmentation accuracy and stability by integrating contextual knowledge. Inspired by the progress of representation learning [15, 62], some studies [36, 50, 69, 70] introduce semantic context and instance knowledge for global-scale context modeling to more ac-



(c) Context prototypes-aware learning and inference

Figure 1. The main idea promoted throughout the paper is that semantic context prototype-aware underpins localization of individual objects in WSSS. Our CPAL performs adaptive perception of diverse attributes (*e.g.,* `cat`) with attribute hunting (c) rather than from single prototype (a) and plain context prototypes (b). This attribute-specific adaptation not only mitigates the risk of errors where (b) mistakenly identifies similar categories (*e.g.,* `dog`) but also ensures accurate activation of the complete object region.

* The first two authors contribute equally to this work.
† Corresponding author: Zongyuan Ge

curately parse the semantic features of instances. But they ignore the challenge of large intra-class variation i.e., regions belonging to the same class may exhibit a very different appearance even in the same picture. The bias between contextual knowledge (global in-class features) and instance-specific knowledge (unique features) makes the labels propagation hard from image-level to pixel-level. In this work, we argue that alleviating the knowledge bias between instances and contexts can capture more accurate and complete regions. Further, we incorporate extra supervised signals to expedite the alleviation of knowledge biases.

Class prototype representation, by diminishing the bias, has shown its potential reveal feature patterns in few-shot learning algorithms such as BDCSPN [40]. Prototype learning theory [58, 75] states that prototypes can represent local features, global features, or specific attributes of an object. Based on intra-class variation in object features, an instance prototype [7] can dynamically characterize the discriminative features of the specific image. As shown in Fig. 1 (a), only few pixels are activated in warm colors, indicating that a large amount of pixels representing the object are mistakenly classified as background. Furthermore, prototypes that integrate contextual knowledge [76] have the ability to capture more specific and accurate category semantic patterns. They enable a more complete capture of the object area compared to a single instance prototype (Fig. 1 (b)). Though the introduction of contextual knowledge enhances the ability of prototypes to process semantic information, knowledge bias between instances and contexts result in prototypes erroneously activating similar or highly co-occurring categories (*e.g.,* `cat` and `dog` in Fig. 1 (b)).

In this work, we propose a learning strategy named Context Prototype-Aware Learning (CPAL) to mine effective feature attributes from the cluster structure of contexts (Fig. 1 (c)). Specifically, we explore other instances related to the specific image to construct contextual prototypes as candidate neighbors. Then, in-class attribute hunting is conducted in the candidate neighbor set, locating the current instance prototype as an anchor. Meanwhile, we design a pairwise positiveness score indicative of the correlation between attributes, aiming to identify contextual prototypes (*i.e.,* soft neighbors) highly related to the current attribute. After applying respective positiveness score, the contributions of these prototypes to the anchored instance were dynamically adjusted, thus explicitly mitigating biases associated with intra-class diversity and instance attributes.

The core of our method is prototype awareness. We softly measure the distance between the instance prototype and the contextual prototype to perceive the instance attributes. For robust estimation, categorical support banks are proposed to overcome the limitations on mini-batch, so intra-class feature diversity can be observed in a feature-to-bank manner where class distribution can be globally approximated. However, due to the limited quantity of instance features, there is a bias relative to the feature distribution of the context, affecting the precise awareness of instance. Therefore, we propose feature distribution alignment by introducing a shifting term δ to the sparse instance features, pushing them towards the dense feature distribution of the categorical support bank.

In the PASCAL VOC 2012 [14] and MS COCO 2014 [38] datasets, we evaluate our method in various WSSS settings, where our approach achieves state-of-the-art performance. The contributions are summarized as follows:

- We propose a context prototype-aware learning (CPAL) strategy that generates more accurate and complete localization maps by alleviating the knowledge bias between instances and contexts.
- We propose a feature alignment module combined with dynamic support banks to accurately perceive the attribute of object instances.
- We propose a unified learning framework consisting of self-supervised learning and context prototype-aware learning, in which two schemes complement each other. Experiments show that our method brings significant improvement and achieves state-of-the-art performances.

2. Related Work

Weakly Supervised Semantic Segmentation using image-level labels typically generates CAM as a seed for generating pixel-level pseudo labels. A typical drawback of CAM is its incomplete and inaccurate activation. To address this drawback, recent work has proposed various training schemes, such as adversarial erasing [27, 28, 52, 68], region growing [22, 61], exploring boundary constraints [4, 34, 44]. The single-image learning and inference model [2, 34] focuses on a deeper understanding of the features within an individual image to generate more complete CAM. SIPE [7] extract customized prototypes multi-scale features to extend coarse object localization maps to obtain the complete extent of object regions.

While past efforts only considered each image individually, recent work focuses on obtaining rich semantic context between different images in the dataset. Recent works [16, 51] address cross-image semantic mining by focusing on capturing pairwise relationships between images. And [13, 36, 70] further perform high-order semantic mining of more complex relationships within a set of images. At the same time, in order to strengthen the representation relationship of the feature space (explore object patterns on the entire data set), RCA [76] introduces a memory bank to store high-quality category features and perform context modeling. CPSPAN [24] proposed to align the feature representation of paired instances under different views, and this alignment was also introduced in the data distribution under different contexts [73]. Unlike previous work on con-

textual knowledge application, our method can adaptively perceive the semantic attributes and intra-class variations, resulting in more complete CAM activation regions.

Prototype-based Learning has been well studied in few-shot [48, 48], zero-shot [19] and unsupervised learning [67]. It is worth noting that many segmentation models can be regarded as prototype-based learning networks [17, 41, 58, 65, 75], revealing the possibility of application in image segmentation. [13] proposed a prototype-based metric learning method that enforces feature-level consistency in interviews and intra-view regularization. LPCAM [8] uses prototype learning to also extract rich features of objects. In our work, we learn effective feature attributes within the clustering structure of the context to model diverse object features at a fine-grained level.

3. Methodology

WSSS first trains the classification network to identify the object region corresponding to each category, then refined to generate pseudo-labels as supervisors of the semantic segmentation network. Fig. 2 illustrates the overview of the proposed method. The framework is built upon the foundation of a classification network, shown in Fig. 2 (a) and described in Section 3.1. It consists of two supervisory signals: classification loss and self-supervised loss. Our approach encourages consistency between the CAM predicted through prototype-aware learning and the classifier, implicitly motivating the model to learn more discriminative features. We model the instance prototype as an anchor and extract context prototypes from the support bank as the candidate neighbor set, which is described in Section 3.2. The core of our method is prototype awareness to capture the intra-class variations, illustrated in Fig. 2 (b) and detailed in Section 3.3. We softly measure the positiveness of each candidate neighbor on the current instance, selectively filter, and adjust their contributions. Meanwhile, feature distribution alignment guides the current instance features toward the cluster center of dense features in the bank.

3.1. Self-Supervised Optimization Paradigm

Network Optimization. Our framework is built upon a classification network, utilizing this network θ to extract effective supervision from image labels, capturing object regions for each category ($i.e.$, CAMs). We propose context prototype-aware learning to generate more complete prototype-aware CAM (PACAM), providing additional supervisory signals for the initial CAM and forming a self-supervised paradigm. The key element of this paradigm is consistency regularization, implicitly reducing the feature distance between discriminative and missing pixels, encouraging the model to learn more consistent and distinctive features. This simple modification leads to significant improvements. A unified loss function optimizes the model:

$$\mathcal{L} = \lambda_{BCE}\mathcal{L}^{BCE} + \lambda_{Self}\mathcal{L}^{Self} \tag{1}$$

where λ_{BCE} and λ_{Self} are coefficients, $\mathcal{L}^{BCE}$ is the classification loss, and $\mathcal{L}^{Self}$ is the self-supervised loss. Losses are described in detail in the following sections.

Classification Loss and Class Activation Maps. Each training image $I \in \mathbb{R}^{w\times h\times 3}$ in the dataset $\mathcal{I}$ is associated with only an image-level label vector $\boldsymbol{y} = \{y_n\}_{n=1}^{N} \in \{0,1\}^N$ for N is pre-specified categories. CAM is proposed to locate the foreground objects by training a classification network. CAM takes a mini-batch image I as the input to extract feature maps $f \in \mathbb{R}^{W\times H\times D}$, with D channels and $H \times W$ spatial size. To bridge the gap between the classification task and the segmentation task, a classifier weight $\mathbf{w}_n$ and a global average pooling (GAP) layer are employed to produce the logits prediction $\hat{y}_i \in \mathbb{R}^N$. During training, binary cross-entropy loss is used as follows:

$$\mathcal{L}^{BCE} = \frac{1}{N}\sum_{i=1}^{N} y_i \log\sigma\left(\hat{y}_i\right) + \left(1-y_i\right)\log\left(1-\sigma\left(\hat{y}_i\right)\right), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. To get rough location information about foreground and background. The class activation map $M_{\boldsymbol{f}} = \{M_n\}_{n=1}^{N}$ over N foreground classes can be represented as follows:

$$M_n = \frac{\mathrm{ReLU}\left(\mathbf{w}_{\boldsymbol{n}}^{\top}\boldsymbol{f}\right)}{\max\left(\mathrm{ReLU}\left(\mathbf{w}_{\boldsymbol{n}}^{\top}\boldsymbol{f}\right)\right)}, \quad \forall n \in N. \tag{3}$$

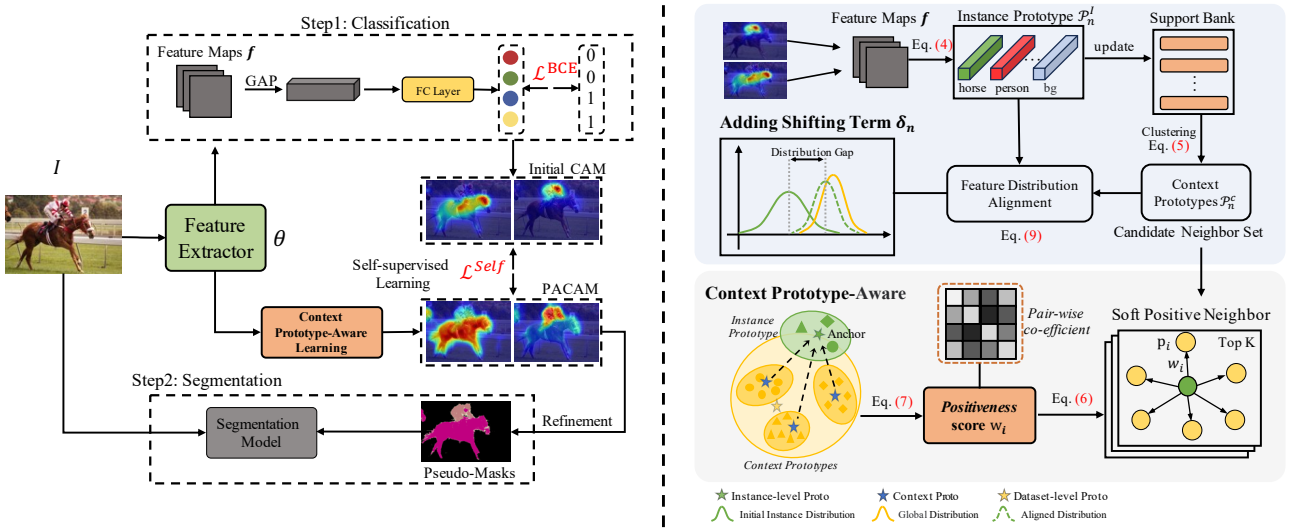
Considering the importance of background in the segmentation task, we follow [60] to estimate the background activation map $M_b = 1 - \max_{1\le n\le N} M_n$ based on M_f. We combine the processed background activation map with the foreground activation map as a whole, *i.e.* $M = M_f \cup M_b$, to help model background knowledge.

3.2. Prototype Modeling

Inspired by prototype-based learning, our prototype awareness strategy aims to effectively explore features within the candidate neighbor set. We propose to conduct a prototype search within the context prototype set for each class, locating the current instance prototype as an anchor to enhance comprehension of the instance features.

Modeling Instance Prototype as Anchor. For each image I, feature maps are mapped to the projection space $z = v(f)$ by a projection head v for instance prototyping. Each instance prototype represents the regional semantics of the categories observed in I based on M. Specifically, for the n-th category that appears in I ($i.e.$, $y_c = 1$), its projected features is summarized to a vector $\mathcal{P}_n^I \in \mathbb{R}^D$ by masked average pooling (MAP) [47]:

$$\mathcal{P}_n^I = \frac{\sum_{x=1,y=1}^{W,H} \mathbf{P}_n(x,y) * z(x,y)}{\sum_{x=1,y=1}^{W,H} \mathbf{P}(x,y)}, \tag{4}$$



(a) The Pipeline of Self-Supervised Optimization Paradigm

(b) Context Prototype-Aware Learning

Figure 2. Overview of the proposed unified learning framework. (a) shows image label-guided WSSS (*from classification to segmentation*). The upper branch describes the classification network θ identifying object regions corresponding to each category to minimize $\mathcal{L}^{BCE}$. Introduce a self-supervised learning paradigm using context prototype-aware learning to provide a more complete CAM, supervising the initial CAM and minimizing $\mathcal{L}^{Self}$. The lower branch refines these CAMs (*e.g.,* DenceCRF [26]) to form pseudo-labels for supervising the semantic segmentation network. (b) outlines our strategy based on context prototype-aware learning. In mini-batches, instance prototypes $\mathcal{P}_n^I$ are generated using CAM and extracted features f, updating the support bank. Then, the bank is used to construct a context prototype set $\mathcal{P}_n^c$. Feature distribution alignment is then applied to the current instance features, adding a shift term δ_n to guide them toward clusters of dense features in the bank. Next, soft neighbors are softly measured for $\mathcal{P}_n^I$ based on $\mathcal{P}_n^c$, with $\mathcal{P}_n^I$ serving as anchors. Finally, *positiveness* value w_i can be computed between two specific attributes. This mechanism selects K soft positive neighbors $\tilde{\mathcal{P}}_n^c$ to generate PACAM.

where $\mathbf{P}_n = \mathbb{1}(M_n > \tau) \in \{0,1\}^{W \times H}$ is a binary mask, emphasizing only strongly-activated pixels of class n in its activation map. $\mathbb{1}(\cdot)$ is an indicator function, and the threshold τ is a hyper-parameter and denotes the threshold of the reliability score. Here, $\mathcal{P}_n^I$ is compact and lightweight, allowing feasible exploration of its relationships with numerous other samples and positioning it as an anchor.

Modeling Context Prototypes as Candidate Neighbors. We assume that categorical features within images or batches only provide a limited view of the class. Therefore, we utilize a support bank as a candidate set $\mathcal{C}$, where each element is the context prototype of different categories. When using sample batches for network training, we store their instance prototypes $\mathcal{P}_n^I$ in $\mathcal{C}$ and employ a first-in-first-out strategy to update the candidate set. This set maintains a relatively large length for each prototype category to sufficiently provide potential context prototypes. Based on this set, k-means online clustering is applied to refine each category into clustered prototype groups $\mathcal{G} = \{G_i\}_{i=1}^{N_p}$ to deeply reveal attributes of each category. We perform averaging operations on each clustered prototype group of $\mathcal{G}$ to generate N_p candidate neighbors $\mathbf{p}_i$ as follows:

$$\mathbf{p}_i = \frac{1}{|G_i|} \sum_{\mathbf{r}_j \in G_i} \mathbf{r}_j, \tag{5}$$

where $\mathbf{r}_j$ refers to the j-th instance prototype belonging to the i-th cluster group G_i. $\mathbf{p}_i$ represents the i-th context prototype of the candidate neighbor set $\mathcal{P}_n^c = \{\mathbf{p}_i\}_{i=1}^{N_p}$.

3.3. Context Prototype-Aware Learning

With the anchor prototypes and the candidate neighbor set from Section 3.2, the candidate neighbor set further perceives or supports the anchor feature. Context prototype-aware learning can measure and adjust this support extent.

Soft Positive Neighbor Identification. Prototype selection is critical in our proposed approach since it largely determines the quality of supervision. Instance prototypes can specifically represent the categorical attributes of the current image, while context prototypes show more comprehensive and diverse category patterns. Our awareness strategy employs positiveness scores w_i to measure the relevance of candidate neighbors in the category to the current instance attributes. We propose selecting the top-K neighbors adjusted by positiveness scores, located in close proximity to the anchor. The soft positive neighbor can be formulated as:

$$\tilde{\mathcal{P}}_n^c = \left\{ w_i \mathbf{p_i} : i \in \underset{i \in N_p}{\arg\max} \left(d\left(w_i \mathbf{p_i}, \mathcal{P}_n^I\right), \text{ top } = K \right) \right\} \tag{6}$$

where $d()$ denotes the cosine similarity as the measured metric, and $\tilde{\mathcal{P}}_n^c$ represents the top-K context-aware prototypes tailored to the current instance.

Positiveness Predictions. We have designed pair-wise positiveness scores to softly measure (in a non-binary form) the relevance between the instance prototype and the candidate neighbors in the same category. For the prototype pair ($\mathbf{p}_i$, $\mathcal{P}_n^I$), the positiveness score w_i can be calculated as:

$$w_i = \frac{1}{\gamma_i} \texttt{softmax}\left[l_1\left(\mathcal{P}_n^I\right) \times l_2\left(\mathbf{p}_i\right)^\top\right], \quad \mathbf{p}_i \in \mathcal{P}_n^c, \tag{7}$$

where $l_1(\cdot)$ and $l_2(\cdot)$ are parameter-free identity mapping layers in feature transformation. γ_i is a scaling factor to adjust the positiveness score w_i. Various structures for the score w_i have been explored in Section 4.2.

Claim 1. *Assume we train a model θ using the proposed optimization method, $\mathcal{P}_n^I$ and $\tilde{\mathcal{P}}_n^c$ are n-th class current instance prototype and context prototypes, respectively. The optimal value of similarity measure s_i^* can be expressed as $\frac{w_i}{\sum_{k=1}^{K} w_k}$, where w_i is the corresponding positiveness score for the prototype pair ($\mathcal{P}_n^I$, $\mathbf{p}_i \in \tilde{\mathcal{P}}_n^c$) in Eq. 7.*

The proof can be found in Appendix A. Claim 1 indicates that we optimize the model to maximize the similarity between the context prototype and the current instance of the same category in direct proportion to the corresponding positiveness score. We effectively transfer knowledge from the self-supervised branch to the model, as well as the generalization performance of the model.

Feature Distribution Alignment. The sparse features [21] and intra-class diversity pose challenges to accurately representing consistent category-specific features, impeding category distinction. Thus, we posit bias between instance and intra-class features. To tackle this, we guide features to align their category-specific densely gathered features to enhance intra-class feature compactness. Considering that mini-batch normalization [23] or instance normalization [54] follows the trend of batch learning, the mini-batch features are aligned by introducing shift terms δ_n to push them towards the cluster centers. The derivation is as follows.

We define the Optimal Cosine Similarity Evaluation Metric (OCSEM) to assess the cosine similarity between the current sample and others, aiming to boost model accuracy by maximizing this metric. The optimization objective is defined as:

$$\text{OCSEM} = \frac{1}{N_p Q_n} \sum_{i=1}^{N_p} \sum_{q=1}^{Q_n} \cos(\mathbf{p}_i, \mathcal{P}_{n,q}^I) > \max_{h \neq i} \{\cos(\mathbf{p}_h, \mathcal{P}_{n,q}^I)\}, \tag{8}$$

where $\mathbf{p}_i$ is the context prototype in the candidate neighbors set $\mathcal{P}_n^c = \{\mathbf{p}_i\}_{i=1}^{N_p}$ for the n-th class, and $\mathcal{P}_{n,q}^I$ is its corresponding instance prototype in the set $\mathcal{P}_n^b = \{\mathcal{P}_{n,q}^I\}_{q=1}^{Q_n}$ in the mini-batch. Q_n denotes the number of prototypes for the n-th class in the mini-batch. We assume the bias can be diminished by adding a shifting term δ_n to the instance feature. The term δ_n should follow the objective:

$$\arg\max_{\delta_n} \frac{1}{N_p Q_n} \sum_{i=1}^{N_p} \sum_{q=1}^{Q_n} \cos\left(\mathbf{p}_i, \mathcal{P}_{n,q}^I + \delta_n\right). \tag{9}$$

We assume that each prototype features $\mathcal{P}_{n,q}^I$ can be represented as $\mathbf{p}_i + \epsilon_{i,q}$. Eq. 9 can be further formalized as:

$$\arg\max_{\delta_n} \frac{1}{N_p Q_n} \sum_{i=1}^{N_p} \sum_{q=1}^{Q_n} \cos\left(\mathbf{p}_i, \mathbf{p}_i + \delta_n + \epsilon_{i,q}\right). \tag{10}$$

To maximize the cosine similarity, we should minimize the following objective:

$$\min \frac{1}{N_p Q_n} \sum_{i=1}^{N_p} \sum_{q=1}^{Q_n} (\epsilon_{i,q} + \delta_n). \tag{11}$$

The term δ_n is thus computed:

$$\delta_n = -\mathbb{E}\left[\epsilon_{i,q}\right] = \frac{1}{N_p Q_n} \sum_{i=1}^{N_p} \sum_{q=1}^{Q_n} \left(\mathbf{p}_i - \mathcal{P}_{n,q}^I\right). \tag{12}$$

3.4. Prototype-Aware CAM and Self-Supervise Loss

Prototype-Aware CAM. With the clear meaning of the prototypes, the predicted CAM procedure can be intuitively understood as retrieving the most similar prototypes. For each prototype $\tilde{\mathcal{P}}_n^c$ in Eq. 6, we compute the cosine similarity between features at each position and the corresponding category prototype. These similarity maps are then aggregated as follows:

$$\tilde{M}_n(j) = \ ReLU\left(\frac{1}{K} \sum_{\mathbf{p}_i \in \tilde{\mathcal{P}}_n^c} \frac{f(j) \cdot \mathbf{p}_i}{\|f(j)\| \cdot \|\mathbf{p}_i\|}\right), \tag{13}$$

where $\|\cdot\|$ denotes the L2-norm of a vector. $\tilde{M}_n(j)$ represents the PACAM for the n-th class at pixel j.

Self-Supervise Loss. To further leverage contextual knowledge, we introduce a self-supervised learning paradigm that encourages consistency between outputs from prototype-aware predictions and a supervised classifier. This promotes the model to recognize more discriminative features and injects prototype-aware knowledge into the feature representation, fostering collaborative optimization throughout training cycles. The L1 normalization of two CAMs defines the consistency regularization:

$$\mathcal{L}^{self} = \frac{1}{N+1} \|M - \tilde{M}\|_1, \tag{14}$$

where M and $\tilde{M}$ represent the original CAM and PACAM, respectively.

Table 1. Ablation study on main components of the proposed framework. The mIoU values are evaluated on the PASCAL VOC 2012 [14] `train` set. $\mathcal{L}^{BCE}$: Baseline classification network. Vanilla: Plain context learning, where context prototypes are clustered from the support bank. Proto-aware: prototype-aware learning involving the top-K candidate neighbor set and positiveness prediction in Eq. 6. Align: Feature alignment, adding shift term within the mini-batch feature in Eq. 12. $\mathcal{L}^{Self}$: Self-supervised loss used as an additional supervised signal in Eq. 14.

	$\mathcal{L}^{BCE}$	Vanilla	Proto-Aware	Align	$\mathcal{L}^{Self}$	mIoU
I	✔					50.1
II	✔	✔				51.2
III	✔	✔	✔			54.5
IV	✔	✔	✔	✔		56.8
V	✔	✔	✔	✔	✔	62.5

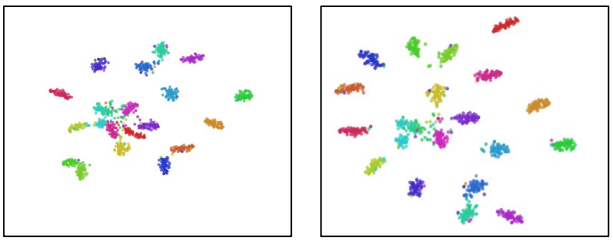
4. Experiments

4.1. Datasets and Implementation Details

Dataset and Evaluation Metric. Experiments are conducted on two benchmarks: PASCAL VOC 2012 [14] with 21 classes and MS COCO 2014 [38] with 81 classes. For PASCAL VOC 2012, following [7, 30, 35, 60], we use the augmented SBD [18] with 10,582 annotated images. We evaluate CPAL in terms of i) quality of generated pseudo segmentation labels on VOC 2012 `train`, and ii) semantic segmentation on VOC 2012 `val/test` and COCO 2014 `val`. Mean intersection over union (mIoU) [42] is used as the metric in both cases. The scores on the VOC 2012 `test` are obtained from the official evaluation server.

Implementation Details. In our experiments, the ImageNet [12] pre-trained ResNet50 [20] is adopted as the backbone with an output stride of 16, where a classifier replaces the fully connected layer with output channels of 20. The augmentation strategy is the same as [1, 7, 8], including random flipping, scaling, and crop. The model is trained with a batch size 16 on 8 Nvidia 4090 GPUs. SGD optimizer is adopted to train our model for 5 epochs, with a momentum of 0.9 and a weight decay of 1e-4. The learning rates for the backbone and the newly added layers are set as 0.1 and 1, respectively. We use a poly learning scheduler decayed with a power of 0.9 for the learning rate.

The loss coefficients λ_{BCE} and λ_{Self} are both set as 1 in Eq. 1. For VOC 2012, the threshold τ in Eq. 4 is set to 0.1. Support bank size for each class to store region embeddings, with the size set to 1000 to avoid significant support consumption. The k-means prototype clustering in Section 3.2 is performed only once at the beginning of each epoch, and the per-class prototype number N_p is set to 50, and the top-K candidate neighbors is set to 20 in Eq. 6. For the segmentation network, we experimented with DeepLab-v2 [6] with the ResNet101 and ResNet38 backbone. *More details (including COCO) are in the appendix.*



(a) ***w/o*** Feature Alignment. (b) Our Method

Figure 3. Feature embedding visualizations of (a) our method without feature distribution alignment, and (b) our method on the Pascal VOC 2012 `val` images using t-SNE [55]. Feature distribution alignment improves the compactness of intra-class features.

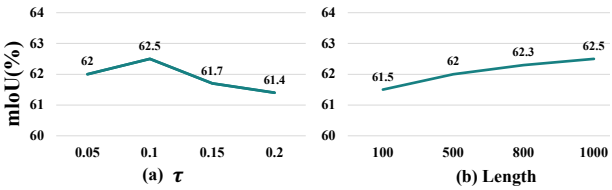


Figure 4. Sensitivity analysis on PASCAL VOC 2012 `train` set, in terms of (a) the threshold τ used to generate 0-1 seed masks from heatmaps. (b) the length of the support set. The results show that CPAL is not sensitive to them.

4.2. Ablation study

To study the contributions of each component of our method, we conducted ablation studies on the VOC 2012 dataset. All experiments use Resnet-50 as the backbone.

Effectiveness of each component. In Table 1, we conduct ablation studies to demonstrate the effectiveness of our approach. We utilize a model trained with only classification supervision training (Experiment I) as the baseline. Then, a plain context prototype learning strategy is introduced in Experiment II and only brings limited gains in mIoU on the `train` set. Experiment III demonstrates that introducing context prototype-aware learning (top-K candidate neighbor set and positiveness prediction) to generate PACAM significantly boosts performance by +3.3%. In Experiment IV, when introducing the feature alignment module, performance further increased by +2.3%. In Experiment V, the performance is further improved by +5.7% when introduced for self-supervised training as complement supervision, indicating its importance in our framework. The consistency loss compels the model to concentrate on fine-grained semantic details, enhancing its perception of the intrinsic structure and semantic features.

Effectiveness of candidate neighbors and positiveness. We analyze the importance of candidate neighbors and positiveness, as shown in Table 2. Removing positiveness and utilizing all neighbors for prediction, Miou accuracy decrease in CAM from 62.5% to 60.3%. It indicates that positiveness is not merely a simple embellishment but rather provides an effective mechanism for the model. It enables

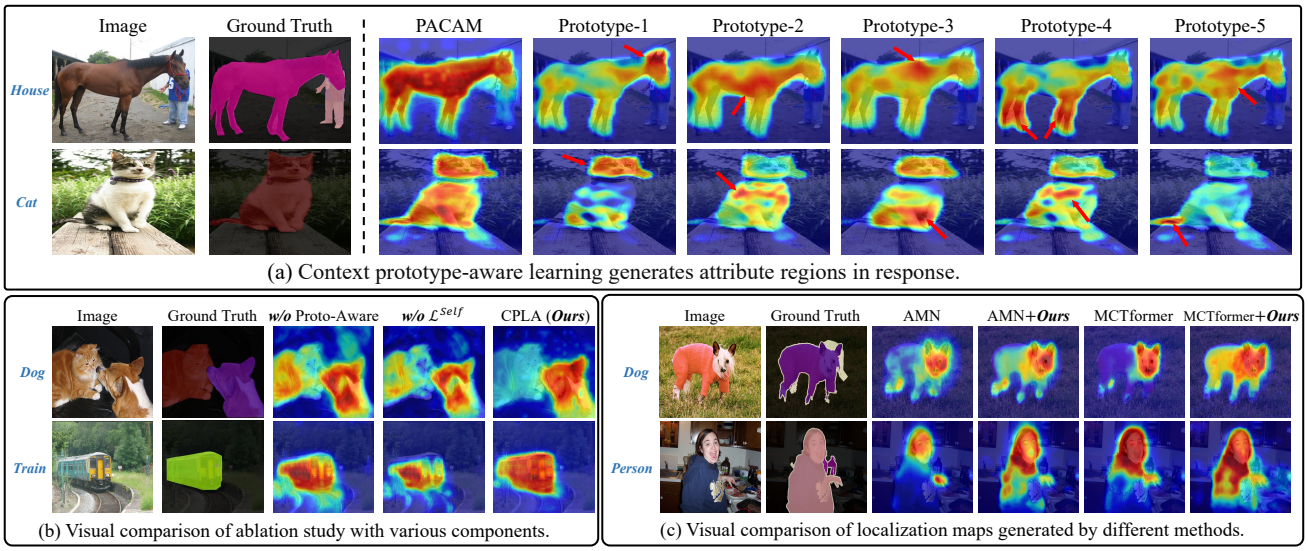


Figure 5. Qualitative visualization on the PASCAL VOC `train` set. (a) PACAM is obtained using various soft positive prototypes to enhance the comprehension of our model. (b) Visual comparison of ablation study two main components: our model without prototype-aware learning (top-K candidate neighbor set and positiveness prediction) or self-supervised loss. (c) The impact of our method as a plug-in to AMN [33] and MCTformer [66] significantly improves the object localization ability of networks.

Table 2. Analysis of the positiveness and number of candidate neighbors K. The mIoU values are evaluated on the PASCAL VOC 2012 `train` set.

Neighbor	Positiveness	K	mIou(%)
✔	✔	20	62.5
✗	✗	-	59.2
✔	✗	20	60.3
✔	✔	10	61.3
✔	✔	20	62.5
✔	✔	50	60.1

the model to adaptively and selectively focus on neighbors that contribute most significantly to the task during the learning process while disregarding neighbors that are uninformative for predictions. In the third block of Table 2, we also conduct experiments to analyze the influence of the number of neighbors. On one hand, having a sufficient number of neighbors enhances the diversity of features. On the other hand, including less-correlated prototypes may introduce too much noise during the training process and diminish the ability of the model to perceive discriminative features. The proposed soft measure introduces a pair-wise positiveness to adjust the contribution of different prototypes to the anchor instance in Eq. 1. We apply various similarity metrics to calculate the positiveness score. As illustrated in Table 3, four options were explored: Manhattan distance (L_1), Euclidean distance (L_2), Cosine similarity, and Dot product. The Dot product demonstrates significantly superior performance compared to the other strategies and is used as our method to measure positiveness.

Table 3. Quantitative comparison of different distance measure strategies in *positiveness* on PASCAL VOC `train` set. The best results are shown in bold.

Function	L_1	L_2	Cosine	Dot
mIou (%)	59.6	58.7	61.9	**62.5**

Effectiveness of feature alignment. In Table 1, we present the performance improvement results achieved by diminishing distribution bias. Additionally, we conducted a visual comparison using t-SNE [55] in Fig. 3. The results indicate that after aligning the feature distributions, the model can generate more compact clusters with higher inter-cluster separability. Adjusting the dynamic shift variable helps alleviate differences between instance features of the same class, making instances belonging to the same class more similar. This, in turn, facilitates the model in distinguishing instances from different categories more accurately.

Analysis of Hyper-parameters. We conduct a hyperparameter sensitivity analysis, varying values such as (a) the threshold τ for generating the 0-1 seed mask. Fig. 4 (a) indicates that the optimal τ value is 0.1. Additionally, we examine (b) the length of the support set, finding that a larger set enhances model performance. Fig. 4 (b) shows that the encoder trained with the largest set achieves the highest accuracy of 62.5%, suggesting that increasing capacity enables the model to find more correlated neighbors for support.

Qualitative Analysis: We visualize the response regions and prediction outcomes of prototype awareness in Fig. 5 (a). It clearly demonstrates that prototypes are associated with specific instance attributes. Specifically, For example, given images (*e.g.,* `horse` and `cat`), each prototype

Table 4. Comparisons between our method and the other WSSS methods. We evaluate mIoU (%) on the PASCAL VOC 2012 `train` set at levels: CAM, w/ CRF, and pseudo Mask.

Method	Seed	w/ CRF	Mask
SEAM [CVPR20] [60]	55.4	56.8	63.6
AdvCAM [CVPR21] [30]	55.6	62.1	68.0
CLIMS [CVPR22] [64]	56.6	-	70.5
SIPE [CVPR22] [7]	58.6	64.7	68.0
ESOL [NeurIPS22] [35]	53.6	61.4	68.7
AEFT [ECCV22] [68]	56.0	63.5	71.0
PPC [CVPR22] [13]	61.5	64.0	64.0
ReCAM [CVPR22] [9]	54.8	60.4	69.7
Mat-Label [ICCV23] [57]	62.3	65.8	72.9
FPR [ICCV23] [5]	63.8	66.4	68.5
LPCAM [CVPR23] [8]	62.1	-	72.2
ACR [CVPR23] [28]	60.3	65.9	72.3
SFC [AAAI24] [72]	64.7	69.4	73.7
IRN [CVPR19] [1]	48.8	53.7	66.5
+CPAL (Ours)	**62.5** ↑13.7	**66.2** ↑12.5	**72.7** ↑6.2
AMN [CVPR22] [33]	62.1	66.1	72.2
+CPAL (Ours)	**65.7** ↑3.6	**68.2** ↑2.1	**74.1** ↑1.9
MCTformer [CVPR22] [66]	61.7	64.5	69.1
+CPAL (Ours)	**66.8** ↑5.1	**69.3** ↑4.8	**74.7** ↑5.6
CLIP-ES [CVPR23] [39]	70.8	-	75.0
+CPAL (Ours)	**71.9** ↑1.1	-	**75.8** ↑0.8

corresponds to different parts of the instance, enabling better modeling of intra-class variations in semantic objects. In Fig. 5 (b), we conduct visualizations of ablation studies on different components of our method. When removing prototype awareness (positiveness and top-K neighbors), the model erroneously activates regions that strongly co-occur (*e.g.,* `train` and `railroad`) or exhibit similar appearances (*e.g.,* `cat` and `dog`), indicating a lack of accurate learning and discriminative capabilities for instance-specific features. Without self-supervised loss $\mathcal{L}^{Self}$, CAM shows under-activation, indicating insufficient learning of category features. These findings suggest that our method, with the introduction of these components, can more accurately perceive and distinguish various category attributes.

4.3. Comparisons with State-of-the-Art Methods

Improved Localization Maps: Since the proposed CPAL does not modify the architecture of the CAM network, it simply integrates the CPAL branch as supervision into multiple methods. Table 4 presents the results of applying CPAL to various well-known methods (IRN [1], AMN [33], MCTformer [66], and CLIP-ES [39]) and show improvements in localization maps on VOC 2012. For instance, incorporating CPAL into AMN improves performance by 3.6% in seed and 2.1% in pseudo masks. When plugging CPAL into the CLIP-ES model, there is a 1.1% gain in the seed. Fig. 5 visualizes the comparison with baseline AMN and MCTformer, showing that CPAL can effectively capture high-quality localization maps.

Table 5. The mIoU results (%) based on DeepLabV2 on PASCAL VOC and MS COCO. $\mathcal{I}$ denotes using image-level labels. $\mathcal{S}$ denotes using saliency maps. $\mathcal{L}$ denotes using Language supervision.

	Methods	Sup.	VOC		COCO
			Val	Test	Val
Trans.	AFA [CVPR22] [45]	$\mathcal{I}$	66.0	66.3	-
	BECO [CVPR23] [44]	$\mathcal{I}$	73.7	73.5	42.0
	ToCo [CVPR23] [46]	$\mathcal{I}$	71.1	72.2	42.3
ResNet38	Spatial-BCE [ECCV22] [63]	$\mathcal{I}+\mathcal{S}$	70.0	71.3	35.2
	MCTformer [CVPR22] [66]	$\mathcal{I}$	71.9	71.6	42.0
	USAGE [ICCV23] [43]	$\mathcal{I}$	71.9	72.8	42.7
	OCR [CVPR23]+SEAM [10]	$\mathcal{I}$	67.8	68.4	33.2
	ACR [ICCV23] [53]	$\mathcal{I}$	71.9	71.9	45.3
	MCTformer +CPAL (Ours)	$\mathcal{I}$	**72.8**	**73.5**	**46.5**
ResNet-101	EPS [CVPR21] [34]	$\mathcal{I}+\mathcal{S}$	70.9	71.0	-
	RIB [NeurIPS21] [29]	$\mathcal{I}+\mathcal{S}$	68.3	68.6	44.2
	EDAM [CVPR21] [62]	$\mathcal{I}+\mathcal{S}$	70.9	71.8	-
	ESOL [NeurIPS22] [35]	$\mathcal{I}+\mathcal{S}$	69.9	69.3	42.6
	RCA [CVPR22] [76] +OOA	$\mathcal{I}+\mathcal{S}$	72.2	72.8	36.8
	IRN [CVPR19] [1]	$\mathcal{I}$	63.5	64.8	42.0
	SEAM [CVPR20] [60]	$\mathcal{I}$	64.5	65.7	32.8
	ReCAM [CVPR22] [9]	$\mathcal{I}$	68.5	68.4	42.9
	OOD [CVPR22] [32] +Adv	$\mathcal{I}$	69.8	69.9	-
	AMN [CVPR22] [33]	$\mathcal{I}$	69.5	69.6	44.7
	SIPE [CVPR22] [7]	$\mathcal{I}$	68.8	69.7	40.6
	LPCAM [CVPR23] [8] +AMN	$\mathcal{I}$	70.1	70.4	45.5
	CLIMS [CVPR22] [64]	$\mathcal{I}+\mathcal{L}$	70.4	70.0	-
	CLIP-ES [CVPR23] [39]	$\mathcal{I}+\mathcal{L}$	73.8	73.9	45.4
	IRN +CPAL (Ours)	$\mathcal{I}$	**71.8**	**72.1**	**42.9**
	AMN +CPAL (Ours)	$\mathcal{I}$	**72.5**	**72.9**	**46.3**
	CLIP-ES +CPAL (Ours)	$\mathcal{I}+\mathcal{L}$	**74.5**	**74.7**	**46.8**

Improved Segmentation Results: Table 5 shows the performance of the semantic segmentation model trained with pseudo-labels generated by our method. Pseudo-labels are utilized to train the DeepLabV2 segmentation model. Comparisons with related works. Our AMN+CPAL achieves state-of-the-art results on VOC (mIoU of 72.5% on the validation set and 72.9% on the test set). On the more challenging MS COCO dataset, our MCTformer+CPAL (with ResNet-38 as the backbone) outperforms the state-of-the-art result AMN and all related works based on ResNet-38. For CLIP-ES, CPAL improves performance (+1.4% mIoU on the COCO val). These superior results on both datasets confirm the effectiveness of our CPAL, which accurately captures the semantic features and object structures.

5. Conclusion

In this work, we propose a novel context prototype-aware learning (CPAL) strategy for WSSS methods, which aims to alleviate the knowledge bias between instances and contexts. This method mines effective feature attributes in context clusters and adaptively selects and adjusts context prototypes to enhance representation capabilities. The core of our method is prototype awareness, which is achieved by context-aware prototypes to accurately capture the intra-class variation and feature distribution alignment. Extensive experiments under various settings show that the proposed method outperforms existing state-of-the-art methods, and ablation studies reveal the effectiveness of our CPAL.

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 6, 8

[2] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, 2020. 2

[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 *CVPR*, 2020. 1

[4] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *ECCV*, 2020. 2

[5] Liyi Chen, Chenyang Lei, Ruihuang Li, Shuai Li, Zhaoxiang Zhang, and Lei Zhang. Fpr: False positive rectification for weakly supervised semantic segmentation. In *ICCV*, 2023. 8

[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. *TPAMI*, 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 *CVPR*, 2022. 2, 6, 8

[8] Zhaozheng Chen and Qianru Sun. Extracting class activation maps from non-discriminative features as well. In *CVPR*, 2023. 3, 6, 8

[9] Zhaozheng Chen, Tan Wang, Xiongwei Wu, Xian-Sheng Hua, Hanwang Zhang, and Qianru Sun. Class re-activation maps for weakly-supervised semantic segmentation. In *CVPR*, 2022. 8

[10] Zesen Cheng, Pengchong Qiao, Kehan Li, Siheng Li, Pengxu Wei, Xiangyang Ji, Li Yuan, Chang Liu, and Jie Chen. Out-of-candidate rectification for weakly supervised semantic segmentation. In *CVPR*, 2023. 8

[11] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[13] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *CVPR*, 2022. 2, 3, 8

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 2, 6

[15] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *CVPR*, 2020. 1

[16] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *AAAI*, 2020. 2

[17] Chongjian Ge, Jiangliu Wang, Zhan Tong, Shoufa Chen, Yibing Song, and Ping Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *ICLR*, 2023. 3

[18] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[19] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *CVPR*, 2019. 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[21] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *JMLR*, 2021. 5

[22] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, 2018. 2

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[24] Jiaqi Jin, Siwei Wang, Zhibin Dong, Xinwang Liu, and En Zhu. Deep incomplete multi-view clustering with cross-view partial sample and prototype alignment. In *CVPR*, 2023. 2

[25] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1

[26] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. 4

[27] 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 *ICCV*, 2021. 2

[28] Hyeokjun Kweon, Sung-Hoon Yoon, and Kuk-Jin Yoon. Weakly supervised semantic segmentation via adversarial learning of classifier and reconstructor. In *CVPR*, 2023. 2, 8

[29] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sungroh Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In *NeurIPS*, 2021. 8

[30] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 1, 6, 8

[31] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In *CVPR*, 2021. 1

[32] Jungbeom Lee, Seong Joon Oh, Sangdoo Yun, Junsuk Choe, Eunji Kim, and Sungroh Yoon. Weakly supervised semantic segmentation using out-of-distribution data. In *CVPR*, 2022. 8

[33] Minhyun Lee, Dongseob Kim, and Hyunjung Shim. Threshold matters in wsss: manipulating the activation for the robust and accurate segmentation model against thresholds. In *CVPR*, 2022. 7, 8

[34] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, 2021. 2, 8

[35] Jinlong Li, Zequn Jie, Xu Wang, Xiaolin Wei, and Lin Ma. Expansion and shrinkage of localization for weakly-supervised semantic segmentation. In *NeurIPS*, 2022. 6, 8

[36] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*, 2021. 1, 2

[37] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 1

[38] 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 *ECCV*, 2014. 2, 6

[39] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *CVPR*, 2023. 8

[40] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *ECCV*, 2020. 2

[41] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 2020. 3

[42] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 6

[43] Zelin Peng, Guanchun Wang, Lingxi Xie, Dongsheng Jiang, Wei Shen, and Qi Tian. Usage: A unified seed area generation paradigm for weakly supervised semantic segmentation. In *ICCV*, 2023. 8

[44] Shenghai Rong, Bohai Tu, Zilei Wang, and Junjie Li. Boundary-enhanced co-training for weakly supervised semantic segmentation. In *CVPR*, 2023. 2, 8

[45] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *CVPR*, 2022. 8

[46] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. Token contrast for weakly-supervised semantic segmentation. In *CVPR*, 2023. 8

[47] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *ICCV*, 2019. 3

[48] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017. 3

[49] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *CVPR*, 2019. 1

[50] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *ICCV*, 2021. 1

[51] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 1, 2

[52] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *ICCV*, 2021. 2

[53] Weixuan Sun, Yanhao Zhang, Zhen Qin, Zheyuan Liu, Lin Cheng, Fanyi Wang, Yiran Zhong, and Nick Barnes. All-pairs consistency learning forweakly supervised semantic segmentation. In *ICCV*, 2023. 8

[54] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 5

[55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 6, 7

[56] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 2017. 1

[57] Changwei Wang, Rongtao Xu, Shibiao Xu, Weiliang Meng, and Xiaopeng Zhang. Treating pseudo-labels generation as image matting for weakly supervised semantic segmentation. In *ICCV*, 2023. 8

[58] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019. 2, 3

[59] Xiaoyang Wang, Bingfeng Zhang, Limin Yu, and Jimin Xiao. Hunting sparsity: Density-guided contrastive learning for semi-supervised semantic segmentation. In *CVPR*, 2023. 1

[60] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 3, 6, 8

[61] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 2018. 2

[62] 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 *CVPR*, 2021. 1, 8

[63] Tong Wu, Guangyu Gao, Junshi Huang, Xiaolin Wei, Xiaoming Wei, and Chi Harold Liu. Adaptive spatial-bce loss for weakly supervised semantic segmentation. In *ECCV*, 2022. 8

[64] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: cross language image matching for weakly supervised semantic segmentation. In *CVPR*, 2022. 8

[65] Haiming Xu, Lingqiao Liu, Qiuchen Bian, and Zhen Yang. Semi-supervised semantic segmentation with prototype-based consistency regularization. In *NeurIPS*, 2022. 3

[66] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, 2022. 7, 8

[67] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. In *NeurIPS*, 2020. 3

[68] 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 *ECCV*, 2022. 2, 8

[69] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*, 2020. 1

[70] Meijie Zhang, Jianwu Li, and Tianfei Zhou. Multi-granular semantic mining for weakly supervised semantic segmentation. In *ACM MM*, 2022. 1, 2

[71] Xiaolin Zhang, Yunchao Wei, and Yi Yang. Inter-image communication for weakly supervised localization. In *ECCV*, 2020. 1

[72] Xinqiao Zhao, Feilong Tang, Xiaoyang Wang, and Jimin Xiao. Sfc: Shared feature calibration in weakly supervised semantic segmentation. In *AAAI*, 2024. 8

[73] Yifan Zhao, Tong Zhang, Jia Li, and Yonghong Tian. Dual adaptive representation alignment for cross-domain few-shot learning. *TPAMI*, 2023. 2

[74] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1

[75] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022. 2, 3

[76] Tianfei Zhou, Meijie Zhang, Fang Zhao, and Jianwu Li. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In *CVPR*, 2022. 2, 8