Sparsely Annotated Semantic Segmentation with Adaptive Gaussian Mixtures

Linshan Wu[1], Zhun Zhong[2], Leyuan Fang *[1], Xingxin He[1], Qiang Liu[1], Jiayi Ma[3], and Hao Chen[4]

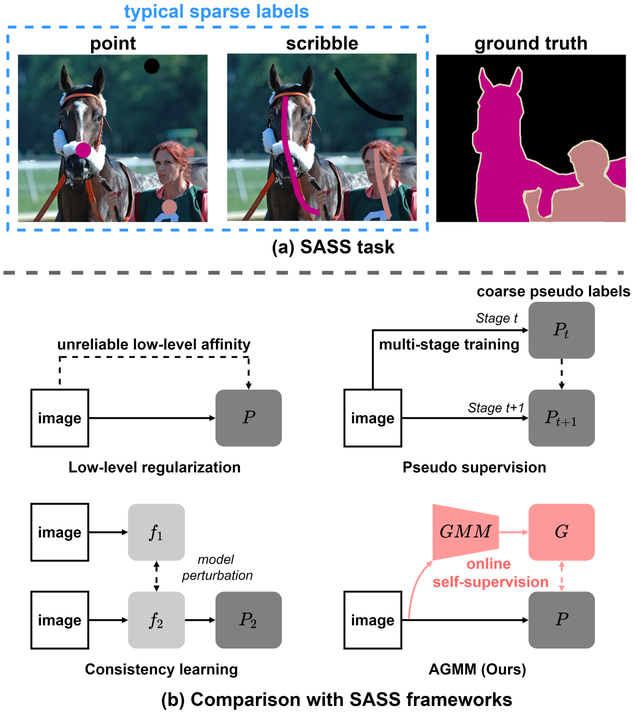
[1]College of Electrical and Information Engineering, Hunan University
[2]Department of Information Engineering and Computer Science, University of Trento
[3]School of Electronic Information, Wuhan University
[4]Department of Computer Science and Engineering, Hong Kong University of Science and Technology

Abstract

Sparsely annotated semantic segmentation (SASS) aims to learn a segmentation model by images with sparse labels (i.e., points or scribbles). Existing methods mainly focus on introducing low-level affinity or generating pseudo labels to strengthen supervision, while largely ignoring the inherent relation between labeled and unlabeled pixels. In this paper, we observe that pixels that are close to each other in the feature space are more likely to share the same class. Inspired by this, we propose a novel SASS framework, which is equipped with an Adaptive Gaussian Mixture Model (AGMM). Our AGMM can effectively endow reliable supervision for unlabeled pixels based on the distributions of labeled and unlabeled pixels. Specifically, we first build Gaussian mixtures using labeled pixels and their relatively similar unlabeled pixels, where the labeled pixels act as centroids, for modeling the feature distribution of each class. Then, we leverage the reliable information from labeled pixels and adaptively generated GMM predictions to supervise the training of unlabeled pixels, achieving online, dynamic, and robust self-supervision. In addition, by capturing category-wise Gaussian mixtures, AGMM encourages the model to learn discriminative class decision boundaries in an end-to-end contrastive learning manner. Experimental results conducted on the PASCAL VOC 2012 and Cityscapes datasets demonstrate that our AGMM can establish new state-of-the-art SASS performance. Code is available at https://github.com/Luffy03/AGMM-SASS.

1. Introduction

Semantic segmentation [2, 8, 42] aims to assign the corresponding pixel-wise semantic labels for a given image, which is a fundamental computer vision task. Previous deep learning based semantic segmentation methods [3, 9, 43] trained on large amounts of data with accurate pixel-wise annotations have demonstrated outstanding achievements. However, collecting such dense annotations always requires cumbersome manual efforts, which heavily limits the development of semantic segmentation methods. To reduce the cost of manual annotations, many re-

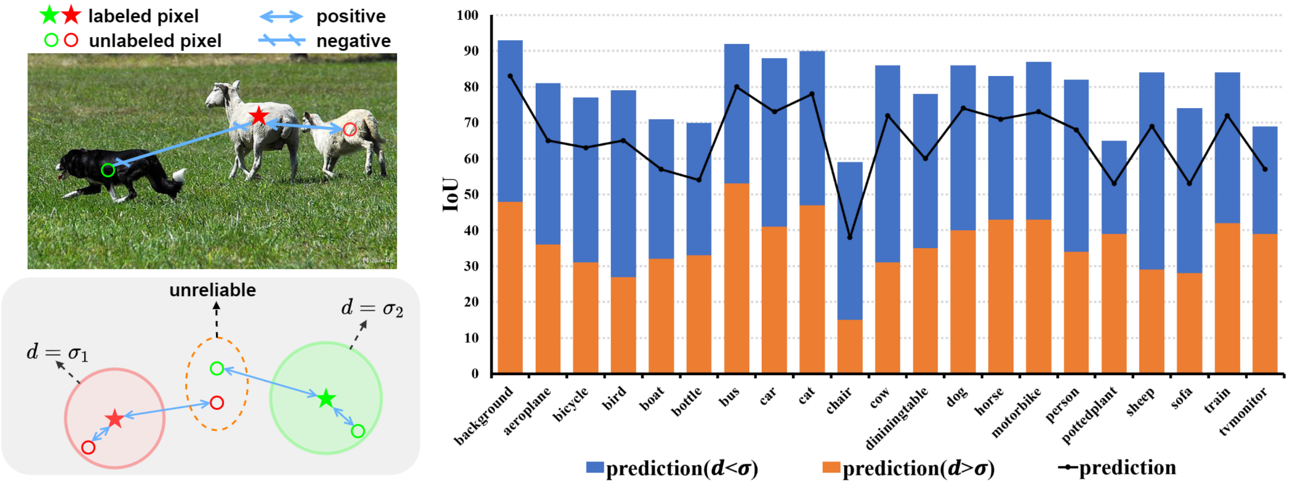


(a) SASS task

(b) Comparison with SASS frameworks

Figure 1. (a) Illustration of SASS task. (b) Different from existing SASS frameworks, our AGMM leverages the reliable information of labeled pixels and generates GMM predictions for dynamic online supervision. f denotes the model, P and G represent segmentation and GMM predictions, respectively. Solid and dashed lines represent model propagation and supervision, respectively.

[*] Corresponding author



(a) Relation between labeled and unlabeled pixels

(b) Category-wise performance

Figure 2. (a) Observation of the inherent relation between the labeled and unlabeled pixels. (b) Category-wise performance on the PASCAL VOC 2012 dataset. The black line, blue bar, and orange bar represent the IoU of all unlabeled pixels, unlabeled pixels that are similar to labeled pixels, and unlabeled pixels that are dissimilar to labeled pixels, respectively. σ is the variance of a class (Eq. 5).

cent works [4,17,27,29,47,48,59] have been made towards sparsely annotated semantic segmentation (SASS), which learns segmentation models via sparse labels, *i.e.*, points or scribbles, as shown in Fig. 1(a). The sparse annotations are cheap to obtain and also contain the least necessary category and location information. Thus, SASS has high research potential in terms of the trade-off between information and costs.

The main challenge of SASS is the lack of information for supervision. Existing SASS methods can be roughly divided into three categories, *i.e.*, low-level regularization [26,30,34,47,48], pseudo supervision [7,27,35,59,60], and consistency learning [19, 38], as shown in Fig. 1(b). Specifically, the low-level regularization methods [26, 30, 34, 47, 48] focus on introducing the low-level affinity of the raw images for supervision. However, the low-level information is not reliable enough to be associated with the high-level semantics. Pseudo supervision [27, 35, 59, 60] aims to generate pseudo labels via training with sparse labels, and then uses these pseudo labels to learn a more robust segmentation model. However, it commonly requires time-consuming multi-stage training and the generated pseudo labels are always coarse and ambiguous, which significantly hinders the learning of unlabeled pixels. Consistency learning [19, 38, 55] further proposes to learn consistent representations in the high-dimension feature space, but it cannot directly supervise the final predictions at the category level.

To solve these problems, we aim to address the SASS task with more reliable supervision. To this end, we argue that the reliable information of labeled pixels should be further exploited. Previous methods only employ the labeled pixels for partial cross-entropy supervision, while largely ignoring the inherent relation between labeled and unlabeled pixels. As illustrated in Fig. 2, we observe that the similarity between labeled and unlabeled pixels is highly associated with the predictions of unlabeled pixels. As shown in Fig. 2(a), if an unlabeled pixel is similar to the labeled pixel in the feature space, its corresponding prediction is more likely to be consistent with the category of the labeled pixel. In Fig. 2(b), we calculate the distance d (see Eq. 6) between labeled and unlabeled pixels to measure the similarity, *i.e.*, $d < \sigma$ as similar and $d > \sigma$ as not similar. It can be seen that the similarity between labeled and unlabeled pixels is highly associated with the accuracy of the predictions. To this end, we propose to explicitly leverage the similarity between the labeled and unlabeled pixels to generate supervision information. The key challenge is how to effectively model the similarity between the labeled and unlabeled pixels.

In this paper, we propose a novel Adaptive Gaussian Mixture Model (AGMM) framework, which is realized by incorporating a GMM branch into the traditional segmentation branch. Specifically, we assign the labeled pixels as the centroids of Gaussian mixtures, enabling us to model the data distribution of each class in the high-dimension feature space. Each Gaussian mixture represents the distribution of a class, which consists of the centered labeled pixels and the relatively similar unlabeled pixels. In this way, we build a GMM to measure the feature similarity between labeled and unlabeled pixels, producing soft GMM predictions to supervise the unlabeled regions from a probabilistic perspective. The process of GMM formulation works in an adaptive manner, where the parameters of GMM are dynamically adapted to the input features, achieving end-to-end online self-supervision. The GMM branch is progressively optimized during training, enabling us to learn more discriminative Gaussian mixtures adaptively.

There are three appealing advantages in our proposed AGMM. First, by capturing category-wise Gaussian mixtures for feature representations, we can learn discriminative decision boundaries between different classes via very limited supervision. Second, AGMM pushes each unlabeled pixel into or away from specific category-wise Gaussian mixtures, which further enables an end-to-end contrastive representation learning. Finally, we leverage the reliable information from labeled pixels to generate GMM predictions for the unlabeled pixels, achieving more reliable supervision.

We conduct experiments under the point- and scribble-supervised settings on two widely used datasets, *i.e.*, PASCAL VOC 2012 [14] and Cityscapes [12]. It is worth noting that compared with existing SASS methods, our AGMM does not require extra information for supervision [19,26,30,34,50], multi-stage training [7,35,37,59,60], and time-consuming post-processing [27,31,50,60]. Extensive experiments demonstrate that our AGMM outperforms the existing state-of-the-art SASS methods.

2. Related Works

Weakly-supervised semantic segmentation: Weakly-supervised semantic segmentation (WSSS) aims to train the semantic segmentation model via coarse weak labels, *e.g.*, image-level labels [1, 18, 20, 54], point-level [4, 29, 47, 48], scribble-level labels [27, 32, 49, 59], and box-level labels [13,60]. WSSS with image-level supervision is widely researched in recent works [1,18,20,54], which usually generates class activation maps (CAM) [61] for training. Although image-level labels require the least effort for manual annotations, they cannot provide the important location information of objects. Thus, these models fail to segment multiple objects with complete constructions and result in limited performance. Although box-level labels [13,60] can provide more information for supervision, they tend to overlap with each other and thus result in confusing supervision during training. In addition, these box-level labels still require time-consuming annotations, which is not efficient for large-scale semantic segmentation.

Compared with image-level and box-level labels, sparse labels such as points and scribbles are more efficient and also provide the least necessary information for supervision. Thus, many recent works propose to use sparse annotations for sparsely annotated semantic segmentation (SASS) [4, 27, 29, 47, 48, 59]. What's the Point [4] first uses point annotations to supervise a semantic segmentation model. Ozan Unal *et al.* [49] proposes to use scribbles to segment LiDAR point clouds. ScribbleSup [27] further proposes to propagate scribble labels via a graphical model for supervision. Most existing SASS methods are based on pseudo supervision [4, 55, 59, 60], which generate pseudo labels and leverage the pseudo labels for multi-stage self-training. However, the quality of coarse pseudo labels may heavily limit the performance. RAWKS [50], BPG [51], and SPML [19] further utilize extra edge information for supervision. However, the edge information also requires additional annotation efforts. To regularize the consistency between labeled and unlabeled pixels, a variety of regularization losses [26, 30, 34, 47, 48, 57] are proposed, which use the low-level affinity from the raw images to supervise the segmentation predictions. However, these regularization losses highly ignore the large gap between the low-level visuals and high-level semantics, which heavily limits the performance of segmentation.

Gaussian Mixture Models: In this paper, we propose a novel SASS framework based on an adaptive Gaussian Mixture Model. GMM is a typical probabilistic model for representing mixture distributions. A GMM consists of K Gaussian mixture components to represent K mixtures distributions, and each component is a Gaussian mixture $g^{'}$ formulated as follows:

$$g^{'}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{1}$$

where x is the input variable, μ and σ represent the mean and variance of the Gaussian distribution $g^{'}$, respectively. Thus, a GMM $G^{'}$ with K components can be formulated as follows:

$$G^{'}(x, \mu, \sigma) = \sum_{i}^{K} g_i^{'}(x, \mu_i, \sigma_i), \tag{2}$$

GMM has been widely applied to model the distributions of hand-crafted features in an unsupervised way [5, 6, 11, 39–41]. Previous methods propose to use expectation–maximization (EM) algorithms [24,33,53] to formulate GMMs, which demand initial prior estimates and iterative parameter updates. However, in SASS, sparsely annotated labels are available, which can be regarded as accurate prior information for GMM formulation. Thus, we can easily formulate a GMM with the help of the annotated information. In this paper, instead of time-consuming EM algorithms, we leverage the reliable information of labeled pixels and employ an effective self-supervision loss function to adaptively optimize the GMM. We will present the details of our designed AGMM in Section 3.

3. Methodology

In this section, we first describe our motivation in Section 3.1. Second, we introduce the overall framework of our proposed AGMM in Section 3.2. After that, we present the details of GMM formulation in Section 3.3. Then, the training losses in our proposed AGMM framework are described in Section 3.4. Finally, in Section 3.5, we further discuss the

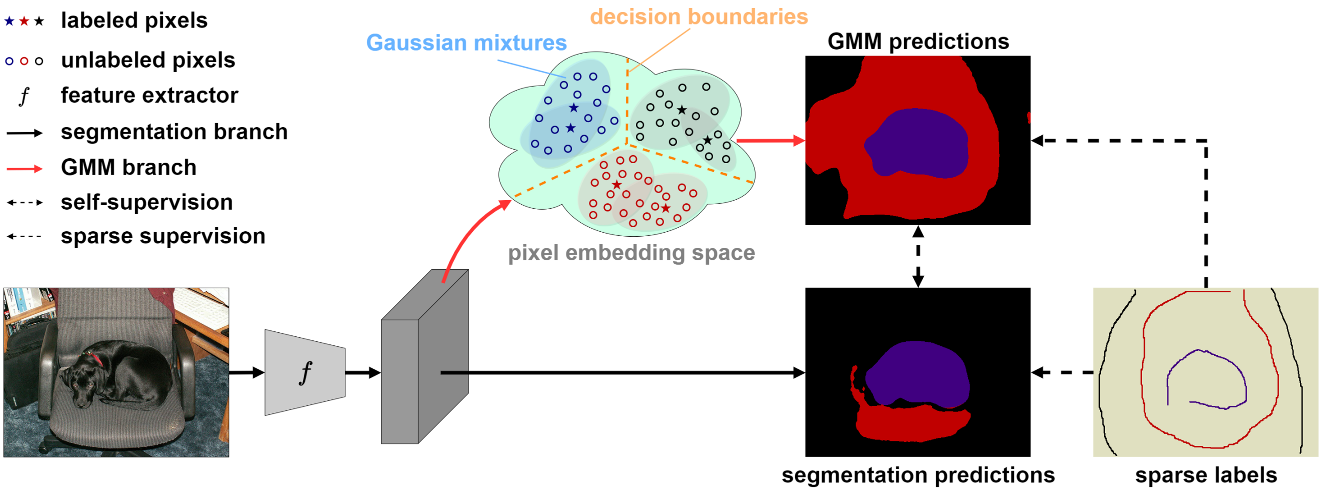


Figure 3. **The overall framework of our proposed AGMM.** AGMM contains a main segmentation branch and a GMM branch. Given an input image x, the segmentation branch directly outputs segmentation predictions P, which is supervised by L_{seg} according to Eq. 3. During training, the extracted deep features $f(x)$ are fed into the GMM branch to generate soft GMM predictions G according to Eq. 7, which is also supervised by the sparse labels Y_l according to Eq. 9. Then we employ segmentation predictions P and GMM predictions G for online self-supervision according to Eq. 8. It is worth noting that during testing, the GMM branch is discarded since the sparse labels are not available in the inference process.

difference between our AGMM and previous SASS methods.

3.1. Motivation

In SASS, the input pixels x can be separated into two parts: labeled pixels x_l and unlabeled pixels x_u. As for the labeled pixels x_l, their corresponding sparse labels y_l can be directly used for supervision with a partial cross-entropy loss L_{seg} as follows:

$$L_{seg} = -\frac{1}{|y_l|} \sum_{\forall y \in y_l} ylog(P_i), \tag{3}$$

where P is the network prediction. However, as for the unlabeled pixels x_u, there is no available label for supervision. One popular solution is to assign pseudo labels to x_u for supervision [7, 26, 27, 59, 60], which requires a time-consuming multi-stage training process. However, these generated pseudo labels are always very coarse and unreliable for supervision, significantly resulting in performance degradation. To address this problem, in this paper, we aim to introduce a more reliable and effective approach for supervising the unlabeled pixels.

In Fig. 2, we observe that two pixels sharing visual similarity tend to belong to the same semantic class. Specifically, if an unlabeled pixel is similar to a labeled pixel, the semantics of these two pixels are more likely to be consistent. Thus, we propose to leverage the similarity between labeled and unlabeled pixels to generate predictions for unlabeled pixels. Then, these predictions can be used to supervise the unlabeled regions, achieving dynamic online self-supervision. However, it is not appropriate to directly set fixed thresholds for the similarity to generate hard one-hot predictions, which will bring a lot of noise and hinder the performance. To solve this problem, we propose to generate the predictions in a soft probabilistic form. In this paper, we propose to use a probabilistic model to measure the similarity and generate soft probabilistic predictions for online self-supervision.

As a typical probabilistic model, GMM can generate multiple Gaussian mixtures to represent the distributions of different categories [5, 6, 11, 25, 40, 41], which can be further introduced into the field of SASS. In SASS, only the labeled pixels can be regarded as completely reliable information. We argue that the learned features of labeled pixels can be seen as the centroids of different Gaussian mixtures. In this way, we can build a GMM to represent the feature distributions, enabling us to model the similarity between labeled and unlabeled pixels. To this end, we propose a simple yet effective AGMM framework for SASS, which benefits both online self-supervision and discriminative representation learning. The details are described as follows.

3.2. Overall Framework

The overall framework of our proposed AGMM is illustrated in Fig. 3, which contains a main segmentation branch and a GMM branch. The segmentation branch directly predicts segmentation results P for L_{seg} supervision according to Eq. 3. In the GMM branch, soft GMM predictions G are generated from the deep features $f(x)$. The GMM predictions G are also supervised by the sparse labels y_l, which is incorporated with a typical cross-entropy loss. Then, we assign these GMM predictions G for online self-supervision with the segmentation predictions P. We illustrate the process of GMM formulation in Section 3.3.

By leveraging these supervisions jointly, the segmentation model can be trained progressively with only limited sparse labels. The details of these training losses will be introduced in Section 3.4. It is worth noting that the GMM branch is employed only during training for generating supervision information, which is discarded in the inference process.

3.3. GMM Formulation

Given an input image with K annotated classes, we build a GMM with K Gaussian mixture components. For i_{th} Gaussian mixture component, we first calculate the mean features of labeled pixels x_{li} belonging to i_{th} class as the mean μ_i:

$$\mu_i = \frac{1}{|x_{li}|} \sum_{\forall x \in x_{li}} f(x), \tag{4}$$

where $f(x)$ are the deep features of pixels x, which are produced from the features before the classification layer of the segmentation model. Once obtaining the μ_i, the variance σ_i of i_{th} component can be calculated as:

$$\sigma_i = \sqrt{\frac{1}{|P_i|} \sum_{\forall x \in x_u} P_i d^2}, \tag{5}$$

where P_i means the segmentation prediction scores of the i_{th} category, and d is formulated as:

$$d = f(x) - \mu_i, \tag{6}$$

which measures the distance between labeled and unlabeled pixels. Similar to Eqs. 1 and 2, we then build a GMM to model the feature distributions of labeled and unlabeled pixels. With the GMM, we produce the GMM predictions G as:

$$G = \sum_{i}^{K} g_i(x, \mu_i, \sigma_i) = \sum_{i}^{K} e^{-\frac{d^2}{2\sigma_i^2}}. \tag{7}$$

Compared with the typical GMM introduced in Section 2, we discard the regularization term $\frac{1}{\sqrt{2\pi\sigma^2}}$. In this way, we can guarantee that for each class, the GMM prediction scores g_i range from 0 to 1, enabling us to conduct self-supervision with the segmentation predictions P. These GMM predictions G are in a form of soft scores, denoting each pixel x belongs to which category-wise Gaussian mixture.

Note that, our proposed GMM is implemented in an adaptive manner, where the parameters of the GMM, *i.e.*, number of components K, mean μ, and variance σ, are dynamically adapted to the input images. Thus, our AGMM can dynamically generate reliable GMM predictions for different input images, enabling us to conduct online supervision. The functions of training losses will be presented in the next section.

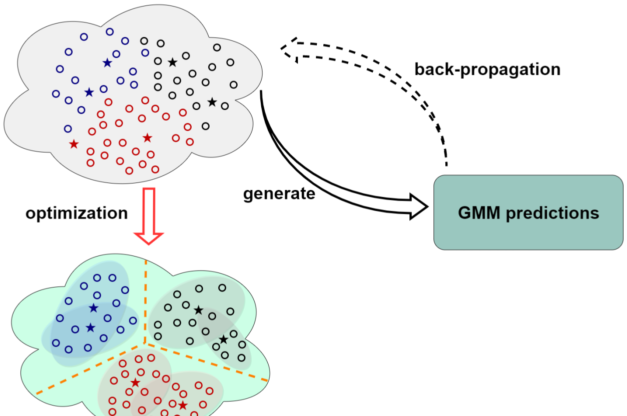


Figure 4. The optimization process of our GMM predictions. With the proposed loss functions according to Eq. 11, we adaptively learn more discriminative category-wise Gaussian mixtures during the optimization process.

3.4. Training with AGMM

Given the GMM predictions G, we assign them for self-supervision with the segmentation predictions P. We adopt a cross-entropy form to formulate the self-supervision loss function L_{self} as follows:

$$L_{self} = -\frac{1}{|x|} \sum [G * log(P) + (1-G) * log(1-P)]. \tag{8}$$

Then, we also assign the sparse labels y_l to supervise G as follows:

$$L_{spar} = -\frac{1}{|y_l|} \sum_{\forall y \in y_l} y log(G). \tag{9}$$

In addition, aiming to learn discriminative Gaussian mixtures, we propose a contrastive loss L_{con} to enlarge the distance between the centroids of different Gaussian mixtures as follows:

$$L_{con} = \frac{2}{K(K+1)} \sum_{\forall i,j \in K, i \neq j} e^{-(\mu_i - \mu_j)^2}. \tag{10}$$

Equipped with these loss functions, we employ the GMM predictions G and the segmentation predictions P to supervise each other mutually. The total loss function L_{GMM} for GMM predictions G can be summarized as follows:

$$L_{GMM} = L_{self} + L_{spar} + L_{con}. \tag{11}$$

Therefore, the overall loss function L in our GMM-SASS framework is formulated as follows:

$$L = L_{seg} + L_{GMM} \tag{12}$$

It is worth noting that we do not stop the gradients of GMM predictions G when calculating L_{GMM}. Since the process of GMM formulation is derivable as described in Section 3.3, our probabilistic GMM predictions G are also optimized progressively during the back-propagation of L_{GMM}. In our AGMM framework, with the mutual self-supervision L_{self}, each unlabeled pixel x_u should be assigned to a specific Gaussian mixture, guiding us to employ strong supervision to the unlabeled regions. As shown in Fig. 4, with the collaborative optimization of L_{seg} and L_{GMM}, we can learn more discriminative class decision boundaries for the generated Gaussian mixtures. In addition, incorporated with the contrastive loss L_{con}, we pull the different Gaussian mixtures of different classes from each other, enabling us to learn more discriminative category-wise representations.

3.5. Discussion

Compared with existing SASS methods, our proposed AGMM is more effective and efficient. First, AGMM does not require the unreliable low-level information [26, 30, 34, 47, 48] or extra edge information [19, 50, 51] for supervision. Second, instead of adopting the time-consuming multi-stage training for pseudo labels generation [7, 35, 37, 60], we leverage the GMM predictions for online self-supervision, which is more efficient. Finally, compared with the consistency learning methods [19, 38], our AGMM not only supervises the features in the high-dimension space but also supervises the final predictions at the category-level. Comprehensive experimental results will be presented in Section 4 to demonstrate the effectiveness of our method.

4. Experiments

In this section, we first describe the datasets and implementation details. Then, we perform detailed extensive ablation experiments for our proposed AGMM. Finally, we report the results of our proposed method compared with other state-of-the-art SASS methods.

4.1. Datasets

To verify the effectiveness of our proposed method, we conduct extensive experiments on two widely-used semantic segmentation datasets: PASCAL VOC 2012 [14] and Cityscapes [12].

PASCAL VOC 2012 [14] originally consists of 1,464 images for training and 1,449 images for validation. Following previous SASS settings, we introduce additional data from the SBD [15] and augment the training set to 10,582 images. It contains 20 foreground classes and a background class for semantic segmentation. Point-level annotations [4] and scribble-level annotations [27] are provided for PASCAL VOC 2012 dataset. To conduct fair comparisons with existing SASS methods, we also report our results on the validation dataset.

L_{seg}	L_{self}	L_{spar}	L_{con}	MT	point sup.	scrib. sup.
✓	-	-	-	-	59.2	67.3
✓	-	-	-	✓	66.3	72.4
✓	✓	-	-	-	68.5	75.2
✓	✓	✓	-	-	69.3	76.1
✓	✓	✓	✓	-	69.6	76.4
✓	✓	✓	✓	✓	**74.7**	**77.2**

Table 1. Ablation study for AGMM on the PASCAL VOC 2012 dataset. MT means multi-stage training.

Cityscapes dataset [12] is created for urban scene-understanding. It contains 19 classes with 2,975 training images and 500 validation images. The images of the dataset are all with 2048 × 1024 pixels. Block-wise annotations are provided in [26] for the Cityscapes dataset. However, these block-wise annotations still cover relatively high ratios of the area (10%, 20%, and 50%), which are not efficient for SASS. Thus, to conduct SASS experiments on the Cityscapes dataset, we randomly select points on the ground truth to create sparse point labels, which include 20, 50, and 100 clicks per image (2048 × 1024) for experiments.

4.2. Implementation Details

To conduct fair comparisons, we employ ResNet [16] pre-trained on ImageNet [21] as the backbone and DeeplabV3+ [10] as the segmentation head to build the network structure in our experiments. Following previous settings [26, 59], multiple data augmentation methods, *i.e.*, random resize, random crop, and random horizontal flip are adopted. The randomly crop size is set to 321 × 321 for PASCAL VOC 2012 dataset and 769 × 769 for Cityscapes dataset. Specifically, we employ the stochastic gradient descent (SGD) optimizer for training, where the initial base learning rate of the backbones is set as 0.001 on PASCAL VOC 2012 and 0.004 on Cityscapes, respectively. For the randomly initialized segmentation head, the learning rate is 10 times larger than that of the backbone. In addition, a polynomial learning rate policy [28] is used to decay the learning rate, where the initial learning rate is multiplied by $(1 - \frac{epoch}{total\_epoch})^{power}$ with a power of 0.9. Momentum and weight decay are set to 0.9 and 0.0001, respectively. The total training epochs are 80 and 240 for PASCAL VOC 2012 and Cityscapes, respectively. We conduct the experiments on Pytorch [36] with 4 NVIDIA 3090 GPUs.

4.3. Ablation Study

Ablation study for AGMM. We first conduct thorough ablation studies for AGMM on the PASCAL VOC 2012

method	hard	soft	online	point sup.	scrib. sup.
Baseline	-	-	-	59.2	67.3
Baseline(+MT)	✓	-	-	66.3	72.4
Label Assignment	✓	-	✓	66.5	73.4
AGMM (SG)	-	✓	✓	67.4	74.6
AGMM	-	✓	✓	**69.6**	**76.4**

Table 2. Effectiveness evaluation of AGMM. We report the mIoU results on the PASCAL VOC 2012 dataset. Hard and soft represent the kind of pseudo labels for supervision. MT and SG denote multi-stage training and stop gradient, respectively.

σ	0.1	0.5	0.8	1.0	1.5	Eq. (5)
point sup.	69.2	69.5	69.5	**69.6**	69.4	**69.6**
scrib. sup.	76.0	76.3	76.2	76.3	76.1	**76.4**

Table 3. Effectiveness evaluation of σ in Eq. 5. We report the mIoU results on the PASCAL VOC 2012 dataset.

dataset, as shown in Table 1. Compared with the baseline method using L_{seg} only, the self-supervision L_{self} between GMM and segmentation predictions can achieve 9.3% and 7.9% mIoU improvements for point- and scribble-supervised SASS, respectively. Adopting L_{spar} and L_{con}, the performance can be further improved. We further evaluate the effectiveness of multi-stage training (MT) strategies. Specifically, we adopt the simplest MT strategy, which generates pseudo labels by AGMM for second-round training. It can be seen that the MT can also improve the performances, especially for the point-supervised setting.

Comparisons with non-adaptive baselines. To prove our core insight, *i.e.*, model the inherent relation between labeled and unlabeled pixels with adaptive GMM, we further compare our method with non-adaptive baselines. First, instead of GMM formulation, we use a similarity-based label assignment method to generate hard one-hot pseudo labels for supervision. Specifically, we simply set a fixed threshold $d < \sigma$ to assign the unlabeled pixels to specific categories, *i.e.*, an unlabeled pixel is assigned to i_{th} category when:

$$d < \sigma_i, d > \sigma_j, \forall i, j \in K, j \neq i. \tag{13}$$

If an unlabeled pixel is not satisfied with Eq. 13, we ignore this pixel during training. In addition, to further evaluate the optimization of GMM formulation as shown in Fig. 4, we stop the gradients of GMM optimization for comparison. In this case, L_{spar} and L_{con} are discarded, and only the segmentation branch is updated during training.

The detailed results are shown in Table 2. It can be seen that it is not appropriate to roughly assign hard one-hot pseudo labels to the unlabeled pixels according to Eq. 13, since we cannot set accurate thresholds for the similarity. In addition, the optimization of GMM branch also plays an important role in our GMM-SASS framework, which indicates that we should not stop the gradients of GMM predictions during training.

Evaluation of variance. We further evaluate the settings of σ in Eq. 5, as shown in Table 3. We compare the effectiveness of adaptive σ obtained by Eq. 5 and fixed σ. It can be seen that the segmentation accuracy is not sensitive to the value of σ. Thus, we argue that the distance between labeled and unlabeled pixels matters more to the GMM formulation. The change of intra-class variance has a low impact to the performance of our proposed AGMM framework.

4.4. Comparison with State-of-the-art Methods

Results on PASCAL VOC 2012. We first conduct point-supervised SASS experiments on PASCAL VOC 2012 dataset. The detailed results are shown in Table 4. Equipped with DeepLabV3+ [10] and ResNet-101 [16], the baseline method incorporated with only partial cross-entropy loss L_{seg} achieves a mIoU of 59.2%. Compared with the baseline method, our proposed AGMM achieves 69.6% mIoU with an improvement of 10.4% mIoU, which demonstrates the effectiveness of our proposed method. Among all the existing SASS methods, TEL [26] produces the best performance with 64.9% mIoU. Specifically, TEL [26] uses the tree filter methods [23, 45, 46] to model both low-level and high-level pair-wise affinity for regularization. To conduct fair comparisons with TEL [26], we also report our results without multi-stage training. Under the same settings, our AGMM outperforms TEL [26] by 4.7% mIoU. The results of point-supervised SASS show that our AGMM can achieve state-of-the-art performance, outperforming existing SASS methods by a large margin.

We further conduct scribble-supervised SASS experiments on PASCAL VOC 2012 dataset. The results are also shown in Table 4. It can be seen that most existing methods employ DenseCRF [8] during testing, which can bring about 3% mIoU improvements. However, this post-processing strategy will significantly increase the cost of computation. Multi-stage training strategy is also widely employed, which requires time-consuming training. It is worth noting that RAWKS [50], BPG [51], and SPML [19] create extra edge information [58] for supervision, but it is unfair for comparisons with other SASS methods. In our experiments, we discard these settings to evaluate the pure effectiveness of our proposed AGMM. Our AGMM achieves 76.4% mIoU and outperforms the baseline by 9.1% mIoU. As shown in Table 4, our proposed AGMM method achieves the state-of-the-art performance without extra edge annotations, multi-stage training, and time-consuming DenceCRF [8].

Results on Cityscapes. To evaluate our method on the Cityscapes dataset, we randomly select point labels from the groud-truth for point-supervised SASS, which include 20, 50, and 100 clicks per image (2048 × 1024) for training. We employ ResNet-50 [16] and DeeplabV3+ [10] for experiments. The results are reported in Table 5. For fair

Method	Network	Publication	Supervision	Extra Data	Multi-stage Training	DenseCRF	mIoU
(1) DeeplabV2 [8]	VGG16 [44]	TPAMI'17	F	-	-	✓	71.6
(2) DeeplabV2 [8]	ResNet101 [16]	TPAMI'17	F	-	-	✓	77.3
* (3) DeeplabV3+ [10]	ResNet101 [16]	ECCV'18	F	-	-	-	78.6
* (3) DeeplabV3+ [10]	ResNet101 [16]	ECCV'18	P	-	-	-	59.2
* (3) DeeplabV3+ [10]	ResNet101 [16]	ECCV'18	S	-	-	-	67.3
What's the Point [4]	(1)	ECCV'16	P	-	-	-	43.4
KernelCut Loss [48]	(2)	ECCV'18	P	-	✓	✓	57.0
A2GNN [60]	(2)	TPAMI'21	P	-	✓	✓	66.8
DBFNet [56]	(3)	TIP'22	P	-	-	-	66.8
TEL [26]	(3)	CVPR'22	P	-	-	-	63.3
AGMM	(3)	-	P	-	-	-	**69.6**
ScribbleSup [27]	(1)	CVPR'16	S	-	✓	✓	63.1
RAWKS [50]	(1)	CVPR'17	S	✓	✓	✓	61.4
GraphNet [37]	(2)	ACM MM'18	S	-	✓	-	70.3
NormCut Loss [47]	(2)	CVPR'18	S	-	✓	-	72.8
DenseCRF Loss [48]	(2)	ECCV'18	S	-	✓	-	73.0
GridCRF Loss [31]	(2)	CVPR'19	S	-	✓	✓	72.8
BPG [51]	(2)	IJCAL'19	S	✓	-	-	73.2
SPML [19]	(2)	ICLR'21	S	✓	✓	-	74.2
URSS [35]	(2)	ICCV'21	S	-	✓	-	74.6
PSI [59]	(3)	ICCV'21	S	-	-	-	74.9
A2GNN [60]	(2)	TPAMI'21	S	-	✓	✓	74.3
DBFNet [56]	(3)	TIP'22	S	-	-	-	72.5
PCE [22]	(3)	NPL'22	S	-	-	-	72.6
CCL [52]	(3)	ACM HCMA'22	S	-	✓	-	74.4
* TEL [26]	(3)	CVPR'22	S	-	-	-	75.8
AGMM	(3)	-	S	-	-	-	**76.4**

Table 4. Experimental results of the point- and scribble-supervised SASS methods on the Pascal VOC 2012 validation set. F, P, and S denote fully-, point-, and scribble-supervised, respectively. Experimental settings with extra data, multi-stage training, and DenseCRF post-processing (DenseCRF) [8] are also considered. * represents we reproduce the approach.

Method	Cityscapes			
	20 clicks	**50 clicks**	**100 clicks**	**full**
Baseline	53.5	60.3	64.2	78.6
DenseCRF Loss [48]	54.2	61.6	65.5	-
TEL [26]	56.3	62.8	67.6	-
AGMM	62.1	68.3	71.6	-
AGMM (+MT)	**66.5**	**71.7**	**73.4**	-

Table 5. Experimental results of the point-supervised SASS methods on the Cityscapes validation set. MT means multi-stage training.

comparisons, we further conduct the experiments with two existing SASS methods [26, 48] based on low-level regularization. However, the improvements of these two methods are very limited. Since the Cityscapes dataset contains more complex scenes with diverse objects and cluttered backgrounds, the low-level affinity is not obvious in the Cityscapes dataset. Thus, the low-level regularization methods [26,48] cannot achieve obvious improvements. It can be seen that compared with existing methods, our method can also achieve the best performance on the Cityscapes dataset. Specifically, AGMM outperforms the existing state-of-art method TEL [26] with a large margin, *i.e.*, by 5.8% improvements with 20 clicks, 5.5% improvements with 50 clicks, and 4.0% improvements with 100 clicks, respectively. Incorporated with the multi-stage training process, the performance of our method can be further improved.

5. Conclusion

In this paper, we proposed a simple yet effective framework AGMM for SASS. Specifically, we assigned the labeled pixels as the centroids of category-wise Gaussian mixtures, enabling us to formulate a GMM to model the similarity between labeled and unlabeled pixels. Then, we can leverage the reliable information from labeled pixels to generate GMM predictions for dynamic online self-supervision. AGMM is progressively optimized during training, enabling us to capture category-wise Gaussian mixtures. In this way, AGMM learns discriminative decision boundaries between different classes and achieves an end-to-end contrastive representation learning. Extensive experiments demonstrate our method achieves state-of-the-art SASS performance.

Acknowledgments

This work was supported in part by the National Natural Science Fund of China (Grant U22B2014), in part by the Science and Technology Plan Project Fund of Hunan Province (Grant 2022RSC3064), in part by the Key Research and Development Program of Hunan (Grant 2021SK2039), in Part by Natural Science Fund of Hunan Province (Grant 2021JJ30003).

References

[1] J. Ahn and S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, Dec. 2017. 1

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, Dec. 2017. 1

[4] A. Bearman, O. Russakovsky, V. Ferrari, and F. F. Li. What's the Point: Semantic segmentation with point supervision. In *Eur. Conf. Comput. Vis.*, 2016. 2, 3, 6, 8

[5] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *Int. J. Comput. Vis.*, 70(2):109–131, 2006. 3, 4

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 3, 4

[7] Hongjun Chen, Jinbao Wang, Hong Cai Chen, Xiantong Zhen, Feng Zheng, Rongrong Ji, and Ling Shao. Seminar learning for click-level weakly supervised semantic segmentation. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pages 6920–6929, 2021. 2, 3, 4, 6

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, Apr. 2017. 1, 7, 8

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, Apr. 2017. 1

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Euro. Conf. Comput. Vis. (ECCV)*, pages 801–818, 2018. 6, 7, 8

[11] Yung-Yu Chuang, Brian Curless, David H Salesin, and Richard Szeliski. A bayesian approach to digital matting. In *IEEE Conf. Comput. Vis. Pattern Recog. CVPR 2001*, volume 2, pages II–II. IEEE, 2001. 3, 4

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3213–3223, 2016. 3, 6

[13] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Int. Conf. Comput. Vis.*, pages 1635–1643, 2015. 3

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. Jour. Comput. Vision*, 88(2):303–338, Jun. 2010. 3, 6

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Int. Conf. Comput. Vis.*, pages 991–998, 2011. 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput Vis. Pattern Recognit.*, pages 770–778, 2016. 6, 7, 8

[17] Xingxin He, Leyuan Fang, Mingkui Tan, and Xiangdong Chen. Intra-and inter-slice contrastive learning for point supervised oct fluid segmentation. *IEEE Trans. Image Process.*, 31:1870–1881, 2022. 2

[18] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[19] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In *ICLR*, 2021. 2, 3, 6, 7, 8

[20] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Eur. Conf. Comput. Vis.*, pages 695–711, 2016. 3

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.*, 25, 2012. 6

[22] Mingchun Li, Dali Chen, and Shixin Liu. Weakly supervised segmentation loss based on graph cuts and superpixel algorithm. *Neural Process. Letters*, pages 1–24, 2022. 8

[23] Stan Z Li. Markov random field models in computer vision. In *Eur. Conf. Comput. Vis.*, pages 361–370. Springer, 1994. 7

[24] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. *Adv. Neural Inform. Process. Syst.*, 2022. 3

[25] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. *Adv. Neural Inform. Process. Syst.*, 2022. 4

[26] Zhiyuan Liang, Tiancai Wang, Xiangyu Zhang, Jian Sun, and Jianbing Shen. Tree energy loss: Towards sparsely annotated semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16907–16916, 2022. 2, 3, 4, 6, 7, 8

[27] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3159–3167, 2016. 2, 3, 4, 6, 8

[28] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 6

[29] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 616–625, 2018. 2, 3

[30] Dmitrii Marin, Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Beyond gradient descent for regularized segmentation losses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10187–10196, 2019. 2, 3, 6

[31] Dmitrii Marin, Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Beyond gradient descent for regularized segmentation losses. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10187–10196, 2019. 3, 8

[32] Christoph Mayer, Radu Timofte, and Grégory Paul. Towards closing the gap in weakly supervised semantic segmentation with dcnns: Combining local and global models. *Comput. Vis. Image Under.*, 208:103209, 2021. 3

[33] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal Process. magazine*, 13(6):47–60, 1996. 3

[34] Anton Obukhov, Stamatios Georgoulis, Dengxin Dai, and Luc Van Gool. Gated crf loss for weakly supervised semantic image segmentation. *arXiv preprint arXiv:1906.04651*, 2019. 2, 3, 6

[35] Zhiyi Pan, Peng Jiang, Yunhai Wang, Changhe Tu, and Anthony G Cohn. Scribble-supervised semantic segmentation by uncertainty reduction on neural representation and self-supervision on neural eigenspace. In *Int. Conf. Comput. Vis.*, pages 7416–7425, 2021. 2, 3, 6, 8

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 6

[37] Mengyang Pu, Yaping Huang, Qingji Guan, and Qi Zou. GraphNet: Learning image pseudo annotations for weakly-supervised semantic segmentation. In *ACM Int. Conf. Multimedia*, pages 483–491, 2018. 3, 6, 8

[38] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *AAAI*, volume 33, pages 8843–8850, 2019. 2, 6

[39] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics (TOG)*, 23(3):309–314, 2004. 3

[40] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23:3, 2012. 3, 4

[41] Mark A Ruzon and Carlo Tomasi. Alpha estimation in natural images. In *IEEE Conf. Comput. Vis. Pattern Recog. CVPR 2000*, volume 1, pages 18–25. IEEE, 2000. 3, 4

[42] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, Apr. 2016. 1

[43] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, Apr. 2016. 1

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 8

[45] Lin Song, Yanwei Li, Zhengkai Jiang, Zeming Li, Xiangyu Zhang, Hongbin Sun, Jian Sun, and Nanning Zheng. Rethinking learnable tree filter for generic feature transform. *Adv. Neural Inform. Process. Syst.*, 33:3991–4002, 2020. 7

[46] Lin Song, Yanwei Li, Zeming Li, Gang Yu, Hongbin Sun, Jian Sun, and Nanning Zheng. Learnable tree filter for structure-preserving feature transform. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 7

[47] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1818–1827, 2018. 2, 3, 6, 8

[48] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *Proc. Euro. Conf. Comput. Vis. (ECCV)*, pages 507–522, 2018. 2, 3, 6, 8

[49] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2697–2707, 2022. 3

[50] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7158–7166, 2017. 3, 6, 7, 8

[51] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *IJCAI Int. Joint Conf. Artifi. Intell.*, 2019. 3, 6, 7, 8

[52] Bin Wang, Yu Qiao, Dahua Lin, Stephen DH Yang, and Weijia Li. Cycle-consistent learning for weakly supervised semantic segmentation. In *3rd Inter. Workshop Human-Centric Multi. Anal.*, pages 7–13, 2022. 8

[53] Zhihui Wang, Shijie Wang, Shuhui Yang, Haojie Li, Jianjun Li, and Zezhou Li. Weakly supervised fine-grained image classification via gaussian mixture model oriented discriminative learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9749–9758, 2020. 3

[54] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T.S. Huang. Revisiting Dilated Convolution: A Simple Approach for Weakly- and Semi- Supervised Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[55] Linshan Wu, Leyuan Fang, Xingxin He, Min He, Jiayi Ma, and Zhun Zhong. Querying labeled for unlabeled: Cross-image semantic consistency guided semi-supervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–18, 2022. 2, 3

[56] Linshan Wu, Leyuan Fang, Jun Yue, Bob Zhang, Pedram Ghamisi, and Min He. Deep bilateral filtering network for point-supervised semantic segmentation in remote sensing images. *IEEE Trans. Image Process.*, 31:7419–7434, 2022. 8

[57] Linshan Wu, Ming Lu, and Leyuan Fang. Deep covariance alignment for domain adaptive remote sensing image

segmentation. *IEEE Trans. Geosci. Remote Sens.*, 60:1–11, 2022. 3

[58] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Int. Conf. Comput. Vis.*, pages 1395–1403, 2015. 7

[59] Jingshan Xu, Chuanwei Zhou, Zhen Cui, Chunyan Xu, Yuge Huang, Pengcheng Shen, Shaoxin Li, and Jian Yang. Scribble-supervised semantic segmentation inference. In *Int. Conf. Comput. Vis.*, pages 15354–15363, 2021. 2, 3, 4, 6, 8

[60] Bingfeng Zhang, Jimin Xiao, Jianbo Jiao, Yunchao Wei, and Yao Zhao. Affinity attention graph neural network for weakly supervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 2, 3, 4, 6, 8

[61] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2921–2929, 2016. 3