

# Learning from Pixel-Level Label Noise: A New Perspective for Semi-Supervised Semantic Segmentation

Rumeng Yi, Yaping Huang, Qingji Guan, Mengyang Pu, and Runsheng Zhang

*Abstract*—This paper addresses semi-supervised semantic segmentation by exploiting a small set of images with pixel-level annotations (strong supervisions) and a large set of images with only image-level annotations (weak supervisions). Most existing approaches aim to generate accurate pixel-level labels from weak supervisions. However, we observe that those generated labels still inevitably contain noisy labels. Motivated by this observation, we present a novel perspective and formulate this task as a problem of learning with pixel-level label noise. Existing noisy label methods, nevertheless, mainly aim at image-level tasks, which can not capture the relationship between neighboring labels in one image. Therefore, we propose a graph based label noise detection and correction framework to deal with pixel-level noisy labels. In particular, for the generated pixel-level noisy labels from weak supervisions by Class Activation Map (CAM), we train a clean segmentation model with strong supervisions to detect the clean labels from these noisy labels according to the cross-entropy loss. Then, we adopt a superpixel-based graph to represent the relations of spatial adjacency and semantic similarity between pixels in one image. Finally we correct the noisy labels using a Graph Attention Network (GAT) supervised by detected clean labels. We comprehensively conduct experiments on PASCAL VOC 2012, PASCAL-Context and MS-COCO datasets. The experimental results show that our proposed semi-supervised method achieves the state-of-the-art performances and even outperforms the fully-supervised models on PASCAL VOC 2012 and MS-COCO datasets in some cases.

*Index Terms*—Semi-supervised semantic segmentation, label noise, graph neural network.

## I. INTRODUCTION

**B**UILDING a large image dataset with high quality annotations for semantic segmentation is costly and time consuming. In order to tackle this problem, semi-supervised approaches have attracted more interests in recent years. Instead of pixel-level annotations, those methods make use of a small set of strong annotations and a large set of weak annotations, such as scribbles [1] [2], bounding boxes [3] or image-level class labels [4] [5], to reduce the data annotation requirements. Among these three types of weak annotations, image-level class labels are the easiest one to obtain and have been widely used to bridge the gap between image-level and pixel-level annotations. In this paper, we address

Rumeng Yi, Yaping Huang, Qingji Guan, Mengyang Pu and Runsheng Zhang are with the Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China (e-mail: 19112038@bjtu.edu.cn; yphuang@bjtu.edu.cn; qingjiguan@gmail.com; mengyangpu@bjtu.edu.cn; rszhang@bjtu.edu.cn).

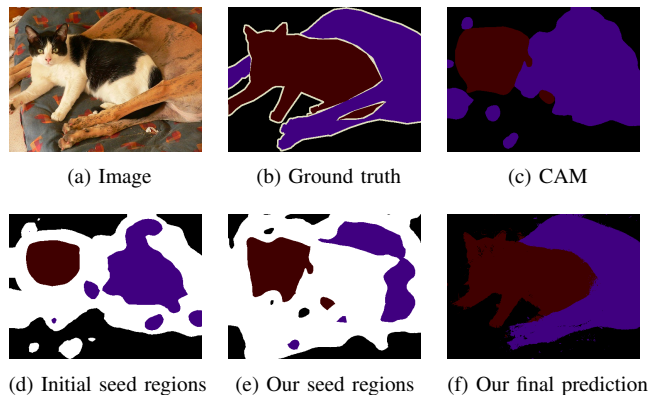


Fig. 1. For (a) an image with class labels, (b) ground truth, and (c) the original CAM, we choose the pixels with activation scores larger than 0.3 as foreground pixels, and the pixels smaller than 0.05 as background pixels to generate (d) initial seed regions (the black represents background and the white represents unlabeled pixels). However, we observe that these regions with high confidence contain many mislabeled pixels. Motivated by this, we propose a new perspective to generate (e) seed regions by detecting noisy labels. Subsequently, we design a graph based method to achieve better (f) pseudo labels by further correcting noisy labels.

semi-supervised semantic segmentation using a small set of pixel-level strong annotations and a large set of image-level weak annotations.

The key step of semi-supervised methods is to infer accurate pixel-level labels for a large number of images with only image-level weak annotations. One popular choice is Class Activation Map (CAM) [6], which highlights local discriminative regions by investigating the contribution of hidden units from a deep classification model. However, the generated object regions are small or sparse, and not sufficient to cover entire object area. In order to generate dense object regions, most previous methods regard these initial discriminative regions with high confidence as segmentation seeds. Then they focus on expanding them with different techniques like learning pixel-level affinities [4], adversarial erasing manner [7] or stochastic regularization [8]. As shown in Figure 1(d), we can see that the regions with high confidence do not mean that their corresponding labels are always correct. We experiment on the PASCAL VOC 2012 *trainaug* set and the accuracy of seed regions is only **86.7%** [9], which largely affect the following expansion process.

To tackle the problem, this paper proposes a novel per-

spective by observing the fact that the pixel-level labels generated by CAM contain a large number of noisy labels (compared with ground truth human annotations). We believe that semi-supervised semantic segmentation can be formulated as a problem of learning with pixel-level label noise. However, existing approaches of learning from noisy labels mainly aim at image classification tasks, such as CIFAR-100 [10], Clothing-1M [11], how to learn with pixel-level noisy labels has not been explored. Compared with image-level noisy labels, the critical issue of pixel-level noisy labels is to model the relations between pixel labels in one image. Therefore, we propose a graph based label noise detection and correction framework to capture the relations between neighboring pixels and further use these relations to correct the noisy labels.

Specifically, we first adopt CAM to generate the pixel-level labels for the images with image-level labels. Then, we train a clean segmentation model with a small set of strong annotations to distinguish the clean labels from the noisy pixel-level labels according to the cross-entropy loss. A common method is to consider samples with smaller loss as clean ones [12] [13] [14]. Subsequently, we construct a superpixel-based graph on one image by considering the dual constraints of spatial adjacency and semantic similarity, and thus we can correct noisy labels by embedding the clean labels into the graph and propagating the clean labels to the noisy labels using a Graph Attention Network (GAT) [15]. Finally, the corrected pixel-level pseudo labels are used to train a semantic segmentation model. Compared with initial seed regions, the accuracy of our seed regions is largely improved (96.7% vs 86.7% in PASCAL VOC 2012 *trainaug* set).

In summary, our contribution of this paper is three-fold:

- We address semi-supervised semantic segmentation from a new perspective and formulate it as a problem of learning with pixel-level label noise.
- We propose a graph based label noise detection and correction framework to deal with pixel-level noisy labels, which can capture the relations between pixel-level labels in one image and correct the noisy labels efficiently.
- We conduct comprehensive experiments on the PASCAL VOC 2012, PASCAL-Context and MS-COCO datasets, and achieve the state-of-the-art performance. Especially, our proposed semi-supervised method even outperforms the fully-supervised models both on the PASCAL VOC 2012 and MS-COCO datasets in some cases.

## II. RELATED WORKS

### A. Semi-supervised semantic segmentation

To reduce annotation effort, most existing approaches rely on semi-supervised training schemes, which use a small set of strong annotations and a large set of weak annotations, such as scribbles [1] [2], bounding boxes [3] or image-level

class labels [4] [5] [16]. Among three types of weak annotations, image-level class labels are the easiest one to obtain. Class Activation Map (CAM) [6] is a good choice for generating pixel-level labels from image-level annotations, which can roughly localize object areas by drawing attentions on discriminative regions of class-specific objects. However, such localization maps are coarse and only activate parts of target objects. To address this issue, several techniques have been proposed to expand these activated regions to the whole target object [17] [18][19] [5] [8]. However, most of those methods need to carefully choose seed regions, which are only based on confidence maps produced by CAM. So they do not make full use of pixel-level strong supervisions when generating pseudo labels from weak annotations. Recently, a few studies provide novel perspectives to solve this problem. [20] designs a strong-weak dual-branch network to exploit strong and weak annotations. [21] proposes a consistency training to enforce an invariance of the model’s predictions over small perturbations applied to the inputs.

Different from the aforementioned approaches, we address semi-supervised semantic segmentation from a new perspective and formulate it as a pixel-level label noise learning problem. Therefore, we can generate more accurate pseudo labels from image-level annotations by making full use of a small set with pixel-level annotations.

### B. Learning with noisy labels

Most large-scale datasets contain noisy labels. The techniques to alleviate the effect of noisy labels can be divided into two categories: (1) detecting noisy labels and then cleansing potential noisy labels or reduce their impacts [22] [12] [23]; (2) directly training noise-robust models with noisy labels [11] [24] [25].

So far, most existing methods are proposed to deal with image classification tasks. Only few works about noisy labels are applied to saliency detection [26], anomaly detection [27] or instance segmentation [28]. In this paper, we propose the first usage of learning with noisy labels for semi-supervised semantic segmentation task, which can be considered as a pixel-wise classification problem. However, relations between the pixel labels need to be adequately modeled, and very few studies have explicitly addressed this with unreliable and noisy labels. Inspired by the previous researches, we propose a graph based label noise detection and correction framework and attempt to learn with noisy labels in pixel-level segmentation task.

### C. Dealing with graph-structured data

Generalizing CNNs to inputs with graph-structured data is an important topic in the field of deep learning. Advances in this direction are often categorized as spectral-based approaches and spatial-based approaches. Spectral-based approaches work with a spectral representation of the graphs and have been successfully applied in the context of node classification. The first prominent spectral research on spectral Graph Convolutional Network (GCN) is presented

in [29]. Later, subsequent studies aim to designing more efficient and flexible spectral-based solutions [30] [31] [32]. In all of the aforementioned spectral approaches, the learned filters depend on the Laplacian eigenbasis. Spatial-based approaches define graph convolutions based on a node’s spatial relations, where an image can be considered as a special form of a graph with each pixel representing a node [33]. Graph Attention Network (GAT) [15] is a representative approach toward spatial-based methods. It introduces an attention-based architecture to perform node classification of graph-structured data, which computes the hidden representations of each node in the graph by attending over its neighbors.

Recently, many studies attempt to apply these techniques in computer vision tasks, such as scribble and bounding box-based weakly semantic segmentation [34], image-text matching [35], visual question answering [36], semantic object parsing [37] and human-object interaction detection [38]. The most relevant work is [34], which uses GCN to deal with scribble and bounding box-based weakly semantic segmentation. However, our approach is fundamentally different from [34]. Directly extending [34] does not deal with image-level annotations. The reason is that GCN heavily relies on the supervision quality. But the generated labels by CAM contain a large number of noise compared with scribble and bounding box, which leads to poor performance. Therefore, one key contribution of our work is to detect clean labels from the perspective of learning with noisy labels, which is the crucial foundation of seed-and-expansion strategies.

### III. PROPOSED METHOD

In this section, we give the details of our proposed label noise detection and correction framework for semi-supervised semantic segmentation. Figure 2 shows the overall framework. Firstly, we adopt CAM to generate the pixel-level labels from image-level annotations and regard them as initial segmentation labels, which contain a large number of noisy labels. Secondly, in order to ensure that the clean labels are detected from the initial segmentation labels accurately, we train a clean segmentation model by using a small set of strong annotations to guide the detection process. Then, we construct a superpixel-based graph on one image by considering the dual constraints of spatial adjacency and semantic similarity, and adopt GAT to correct noisy labels supervised by clean labels. Finally, the corrected labels are used as pseudo labels to train a segmentation network in a semi-supervised manner.

#### A. Generating pixel-level labels from weak annotations

We follow the approach of [6] to compute CAMs of training images as initial segmentation labels. The architecture is a typical classification network with global average pooling (GAP) followed by a fully connected layer, which is trained with image-level labels. Given an image, the CAM of a ground truth class  $c$  is computed by:

$$M_c(x, y) = \sum_k \omega_k^c f_k(x, y) \quad (1)$$

where  $M_c$  is the class activation map for class  $c$ ,  $f_k(x, y)$  is the activations of unit  $k$  in the last convolutional layer at spatial location  $(x, y)$ , and  $\omega_k^c$  is the weight corresponding to class  $c$  for unit  $k$ . Besides, for any class  $c'$  irrelevant to the ground truths, we disregard  $M_{c'}$  by setting its activation scores to zero. Thus we can assign the corresponding class label to each pixel according to the highest activation score, and the pixels with activation scores smaller than 0.05 are background. These generated pixel-level labels can be used as the initial segmentation labels, which obviously contain inevitable noisy labels.

#### B. Detecting clean and noisy labels

The key issue of our proposed method is to detect clean and noisy labels from the initial segmentation labels computed by CAM. In this section, we propose to use a small set of pixel-level annotations to train a clean segmentation model, and then detect the clean and noisy labels according to the cross-entropy loss of generated pixel-level labels trained on the clean model.

Specifically, our training dataset  $\mathcal{D}$  comprises of two subsets:  $\mathcal{D}^c : \{(x_i^c, y_i^c), 1 \leq i \leq M\}$  with a small set of pixel-level annotations, where  $y_i^c$  is the ground truth pixel-level labels corresponding to  $i$ -th input image  $x_i^c$ ; and  $\mathcal{D}^n : \{(x_j^n, y_j^n), 1 \leq j \leq N, M \ll N\}$  with image-level annotations, where  $x_j^n$  is the  $j$ -th input image and  $y_j^n$  is the corresponding pixel-level labels produced by CAM. Firstly, we train a segmentation network on  $\mathcal{D}^c$  and obtain a clean segmentation model  $C$ . Then we use  $C$  to predict the label for each pixel  $p$  of  $x_j^n$  in  $\mathcal{D}^n$  and calculate their cross-entropy loss  $l_{jp}$  under the supervision of  $y_{jp}^n$ :

$$l_{jp} = -y_{jp}^n \log(F_C(x_{jp}^n)) \quad (2)$$

where  $F_C$  is a segmentation function of clean model  $C$  that projects the image to the prediction. Following many noisy label learning approaches which consider samples with smaller loss as clean ones [12] [39] [22], we set a threshold  $\theta$  to distinguish the labels with small loss as the clean labels and then use these clean labels as the supervisory information for training GAT.

Because the threshold  $\theta$  plays an important role in detecting clean and noisy labels. In this paper, we introduce an efficient strategy to select a suitable threshold. At first, we generate the pixel-level labels by CAM on  $\mathcal{D}^c$ . Then we use the clean model  $C$  to predict the label for each pixel in  $\mathcal{D}^c$  and calculate their cross-entropy loss under the supervision of pixel-level labels computed by CAM. Since we have the ground truth labels on  $\mathcal{D}^c$ , we can easily choose a suitable threshold according to the loss distributions of the clean labels. It should be noted that we need to consider both the quantity and quality of the labels we choose, which means that we need to guarantee that not only the labels we choose are correct as much as possible but also the number of selected labels is not too little. The detailed experiments will be given in ablation study.

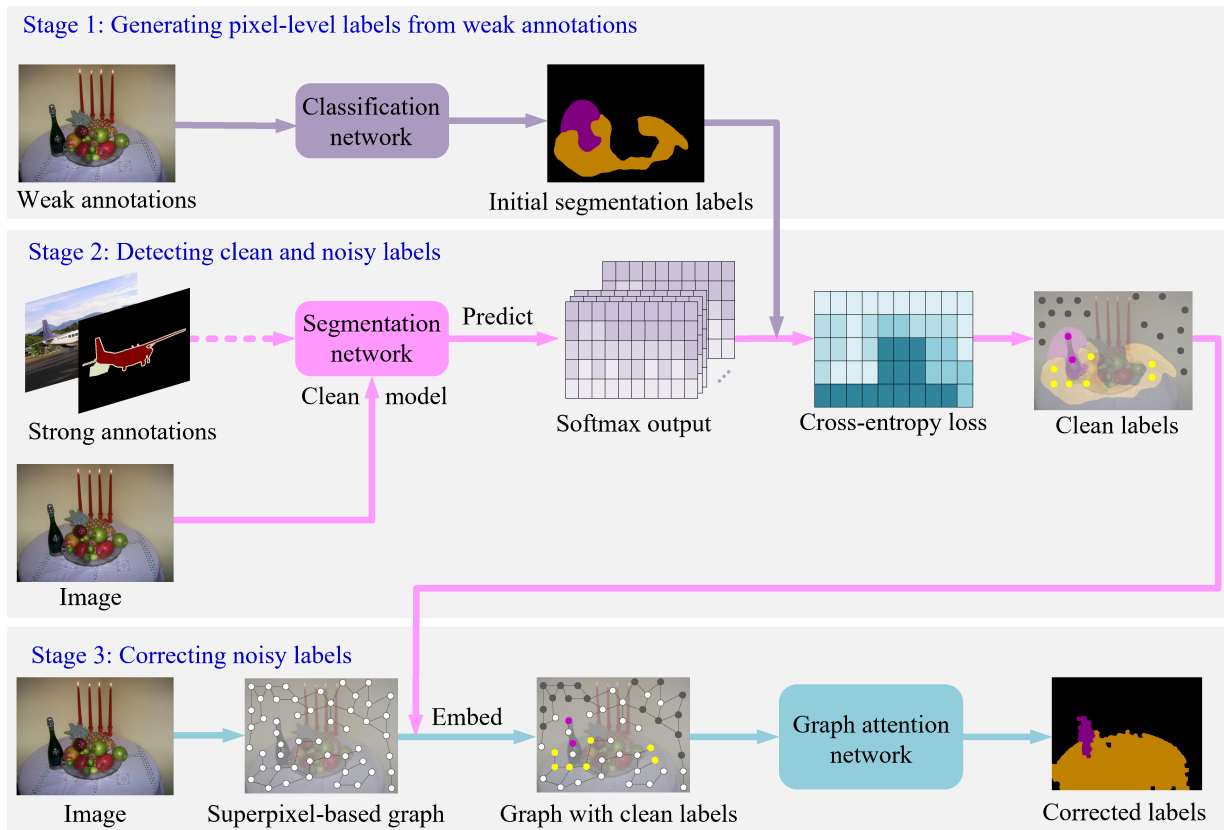


Fig. 2. Overview of the label noise detection and correction framework for semi-supervised semantic segmentation. Firstly, we adopt CAM to generate the pixel-level labels for a large set of images with weak annotations as the initial segmentation labels. Secondly, we train a clean segmentation model with strong annotations to detect clean labels from the initial segmentation labels. Then we construct a superpixel-based graph on one image and embed the clean labels into the graph. Finally, we correct the noisy labels by using a GAT.

### C. Correcting noisy labels

In this section, we aim to correct the pixel-level noisy labels which are selected in detecting clean and noisy labels stage. However, most existing approaches of learning with noisy labels mainly focus on image classification tasks, which do not consider to model the relations of the pixel labels in one image. Therefore, we develop a graph attention network-based method to correct the pixel-level noisy labels. Specifically, we first construct a superpixel-based graph on an image by considering the dual constraints of *spatial adjacency* and *semantic similarity* [34]. Then the clean labels are embedded into the graph. Finally, we employ GAT to propagate the label information from clean labels to noisy labels.

It should be noted that other message-passing neural networks that operate on graph-structured data can be also employed to perform the noisy labels correction in our framework, such as GCN [30]. Considering the better performance, we apply GAT in our paper. The comparison between GCN and GAT is given in ablation study.

1) *Superpixel-based graph construction*: Considering that superpixel can provide a larger, locally homogeneous and coherent regions that preserve most of the structure

necessary for accurate segmentation [40], we transform an image to a superpixel-based graph representation  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ , where  $\mathcal{V}$  is a set of nodes,  $\mathcal{E}$  is a set of edges and  $\mathcal{A}$  is an adjacency matrix. In particular,  $\mathcal{V}$  denotes the superpixel set  $\{sp_i\}_{i=1}^N$ , the edge  $\varepsilon_{ij}$  in  $\mathcal{E}$  connects  $sp_i$  and  $sp_j$  where the two nodes are neighboring in space, and  $\mathcal{A}$  represents the nodes proximity.

**Vertex construction.** Firstly, we use the Simple Linear Iterative Clustering (SLIC) [41] method to over-segment one image and divide it into a superpixel set, denoted as  $\{sp_i\}_{i=1}^N$ , where  $N$  is the number of superpixels contained in one image. In our experiments, one image is over-segment into about 1000 superpixels. Then we use the conv5-3 layer of the clean model  $\mathcal{C}$  to extract the high-level semantic features from the whole image and integrate over superpixels. Since the feature maps are downsampled by the operation of several convolutional layers, we apply a bilinear interpolation on the feature maps to obtain the dense feature maps with the same size as the original images, and then an average pooling is performed on a superpixel along the channels. Finally, a 512-dimensional CNN feature vector for each superpixel  $sp_i$  is obtained.

**Edge construction.** We leverage two characteristics of images to construct edges of the graph on an image, *i.e.*,

*spatial adjacency* and *semantic similarity*. Intuitively, spatial adjacency means neighboring pixels tend to have similar labels, while semantic similarity means the pixels with the same labels probably share similar semantic information. We assume that two spatially adjacent nodes which have similar semantic content commonly tend to belong to the same class. Thus, we consider the dual constraints of spatial adjacency and semantic similarity to construct edges.

Firstly, we model the spatial adjacency on  $\mathcal{G}$ . We construct the *spatial adjacency weight matrix*  $W_l = [w_l^{ij}]_{n \times n} \in \mathbb{R}^{N \times N}$  to measure the spatial adjacent relationship between  $sp_i$  and all other superpixels. If  $sp_i$  and  $sp_j$  are spatially adjacent, then the weight  $w_l^{ij}$  is set to 1, otherwise  $w_l^{ij}$  is set to 0. Then we model the semantic similarity on  $\mathcal{G}$ . Specifically, we construct a *semantic similarity weight matrix*  $W_s = [w_s^{ij}]_{n \times n} \in \mathbb{R}^{N \times N}$  to measure the semantic similarity between  $sp_i$  and its spatially neighboring superpixels. Given the 512-dimensional feature vector  $\{v_i\}_{i=1}^N$  for each superpixel  $sp_i$ , the weight  $w_s^{ij}$  is given by:

$$w_s^{ij} = w_l^{ij} \times \exp\left(-\frac{\|v_i - v_j\|}{2h}\right) \quad (3)$$

where  $h$  is the dimension of the feature vector. If two superpixels  $sp_i$  and  $sp_j$  are not spatially adjacent, *i.e.*,  $w_l^{ij} = 0$ , we will ignore their semantic relationship.

Subsequently, the adjacency matrix  $\mathcal{A} = [a_{ij}]_{n \times n} \in \mathbb{R}^{N \times N}$  is given by:

$$a_{ij} = \begin{cases} 0, & \text{if } w_s^{ij} < \gamma \ \& \ w_s^{ij} < \max_{k \in \mathcal{N}_{\mathcal{G}}(i)} (w_s^{ik}) \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where  $\gamma$  is a threshold to filter out the edges with low similarity from the edge set  $\mathcal{E}$ . In our work,  $\gamma = u(W_s) - \sigma(W_s)$ , where  $u(\cdot)$  and  $\sigma(\cdot)$  are mean and standard deviation, respectively.  $\max_{k \in \mathcal{N}_{\mathcal{G}}(i)} (w_s^{ik})$  is the maximum semantic similarity between  $sp_i$  and its neighboring superpixels  $sp_k$ . Moreover, if  $w_s^{ij}$  is lower than both  $\gamma$  and  $\max_{k \in \mathcal{N}_{\mathcal{G}}(i)} (w_s^{ik})$ , the edge  $\varepsilon_{ij}$  is removed.

2) *Correcting noisy labels*: After constructing the superpixel-based graph for each image, the clean labels which are selected in detecting clean and noisy labels stage are embedded into the graph as the supervision information for training GAT.

**Embedding clean labels.** We denote the clean labels set of each image as  $\mathcal{S} = \{s_k, c_k\}_{k=1}^K$ , where  $s_k$  is the  $k$ -th pixel and  $c_k$  is the category label of  $s_k$ . If  $sp_i$  overlaps with  $s_k$  ( $sp_i \cap s_k \neq \emptyset$ ), we will assign the category label  $c_k$  to  $sp_i$ . However, we find that one superpixel usually contains more than one clean label in our experiments, so we need to select the corresponding label with the largest number of clean labels and assign it to the superpixel.

**Correcting noisy labels by GAT.** We transform an image to a superpixel-based graph representation and leverage GAT [15] to correct noisy labels. Following [15], we perform

*self-attention* on the nodes to compute attention coefficients. The importance of node  $j$ 's features to node  $i$  is given by:

$$e_{ij} = \varphi(v_i)^T \phi(v_j) \quad (5)$$

where  $\varphi(v_i) = W_\varphi v_i$  and  $\phi(v_j) = W_\phi v_j$ .  $W_\varphi$  and  $W_\phi$  are the weight parameters learned by back propagation. We only compute  $e_{ij}$  for nodes  $j \in \mathcal{N}_i$ , where  $\mathcal{N}_i$  is the set of neighboring nodes of node  $i$  in the graph. To make coefficients easily comparable across different nodes, we normalize  $e_{ij}$  across all choices of  $j$  using the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \quad (6)$$

Then we extend the above graph attention mechanism by employing multi-head attention, and the final output features for each node are given by:

$$v'_i = \parallel_{l=1}^L \sigma\left(\sum_j \alpha_{ij}^l W_g^l v_j\right) \quad (7)$$

where  $\parallel$  denotes concatenation,  $\sigma(\cdot)$  is a nonlinear function such as ReLU,  $\alpha_{ij}^l$  are normalized attention coefficients computed by the  $l$ -th attention mechanism, and  $W_g^l$  is the corresponding input linear transformation's weight matrix.

In our experiment, we adopt a two-layer GAT for label correction. The forward model is given by:

$$Z = f(V, A) \quad (8)$$

where  $V$  is the superpixel-based feature matrix computed by Eq.7,  $A$  is the adjacency matrix computed by Eq.4, and the cross-entropy loss over all superpixels with clean labels is defined by:

$$\mathcal{L} = - \sum_{i=1}^P p_i \ln z_i \quad (9)$$

where  $p_i$  is the corresponding label of  $sp_i$ ,  $P$  is the number of the superpixels with clean labels and  $z_i$  is the GAT's prediction of  $sp_i$ .

#### D. Training the segmentation network

The category information of pixels which were originally considered to be noise in each image has been corrected by GAT, thus we can recover the label of each pixel according to their corresponding superpixel and then the corrected segmentation labels are refined by dense CRF [42] to better estimate object shapes.

The segmentation labels obtained by detecting and correcting stages are finally used as supervision to train a segmentation network. Note that any fully supervised semantic segmentation model can be employed in our approach.

## IV. EXPERIMENTS

In this section, we evaluate our approach on PASCAL VOC 2012 [43], PASCAL-Context [44] and MS-COCO [45] datasets, where our framework generates segmentation labels as ground truth labels for training a segmentation model. The

performance is measured using the mean Intersection-over-Union (mIoU) across the available classes.

**Datasets.** We evaluate our approach on PASCAL VOC 2012, PASCAL-Context and MS-COCO datasets. The original PASCAL VOC 2012 dataset consists of 1464 training, 1449 validation, and 1456 test images covering 21 classes (one background class). An auxiliary dataset of 9118 training images is provided by [46]. PASCAL-Context dataset is a whole scene parsing dataset containing 4998 training and 5105 testing images with dense semantic labels, and this dataset involves 60 classes (one background class). MS-COCO dataset contains 81 classes (one background class), 80k and 40k images for training and validation.

**Implementation details.** We use the DeepLab-CRF-LargeFOV model [47] where the parameters are initialized by VGG-16 [48] network pre-trained on the ImageNet dataset [49] (hereinafter referred to as "DeepLab v1-vgg16") and Deeplab v2 model [50] where the parameters are initialized by resnet-101 network [51] pre-trained on the ImageNet dataset (hereinafter referred to as "DeepLab v2-resnet101") as the basic network. And we use the standard settings for all parameters.

#### A. Results on PASCAL VOC 2012 dataset

**Experimental results.** We compare our approach with current state-of-the-art methods [5] [8] [16] [18] [19] [21] [20]. We adopt the same strong/weak split, *i.e.*, 1.4K strongly annotated images and 9K weakly annotated images. Table III summarizes the comparison. To the best of our knowledge, most semi-supervised methods report the results on Deeplab v1-vgg16, only [21] and [20] are performed on resnet50 [21] and Deeplab v2-resnet101 [20]. Therefore, for a fair comparison, we illustrate the backbone used for each approach in Table III, and evaluate our method on different backbones. From the results, we can see that our proposed method outperforms other methods and achieves the best mIoU of 67.5% and 67.9% (v1-vgg16) and 77.1% and 77.2% (v2-resnet101) on *val* and test sets, respectively. It should be noted that the performance on *val* set outperforms the fully-supervised model (76.5%) by 0.6% using v2-resnet101 segmentation network. We explain this surprising result and find a possible explanation. We believe the pseudo labels corrected by GAT can successfully highlight some instances which are improperly labeled in ground truth annotations, thus resulting in an improvement in the performance of the segmentation network, which exceeds the fully-supervised model. Moreover, our proposed method dedicates to generate more accurate pseudo-labels, and thus the performance can be further improved by incorporating the strong-weak dual-branch network [20].

Subsequently, we evaluate our approach with different sizes of fully-labeled set  $\mathcal{D}^c$ . Similar to [52], we use 200, 400, 800 and 1464 images with strong annotations to train the clean model  $C$  respectively, and compare to the different baselines which are trained only on  $\mathcal{D}^c$ . As shown in Table IV, our method achieves an improvement of 5.0% to

16.9% (v1-vgg16) and 2.5% to 11.2% (v2-resnet101) over the baselines for different sizes splits. More surprisingly, our performance on *val* set outperforms the fully-supervised model (76.5%) by 0.6% using v2-resnet101 segmentation network. Notably, the approach works well even with only 1.89% ( $\mathcal{D}^c = 200$ ) of fully-labeled images, which illustrates that our method can provide a good training signal for the weak set so as to largely alleviate the problem of lacking of segmentation labels. The quantitative results on per-class IoU are shown in Table I and II, and the qualitative results are shown in Figure 3.

**Ablation study.** We explore the performance of the generated pseudo labels on 1.4K/9K split in the stages of: 1) Generating pixel-level labels from weak annotations; 2) Detecting clean and noisy labels; 3) Correcting noisy labels. The performance of the generated pseudo labels are measured by the segmentation results using the Deeplab v1-vgg16 model. We train the DeepLab model without and with 1.4K strongly annotated images respectively, and evaluate the mIoU on the validation set.

As shown in Table V, the segmentation result of initial CAM is 57.4%. For the first strategy (w/ detection and w/o correction), the 1.4K strongly annotated images are applied to train a clean model  $C$ , then we use  $C$  to detect the clean labels from initial CAM and train a segmentation network with these clean labels, the performance is boosted by 5.4% (57.4% vs. 62.8%). The result demonstrates that our detecting method can distinguish clean labels from noisy labels. The result also verifies the effect of noisy labels: deep models can learn effective information from the limited clean labels, and however the presence of noisy labels dramatically harms the generalization of deep models.

For the second strategy (w/o detection and w/ correction), we just select clean labels according to the activation scores from initial CAM. Similar to [9], we consider pixels with activation scores larger than 0.3 and smaller than 0.05 as foreground and background clean labels, and use them as the supervision for training GAT. Such strategy is similar with [34], but using GAT instead of GCN. However, the segmentation result is only 59.6%, which is much lower than that of using our proposed detection strategy. The result illustrates that the seed regions with high confidence contain many mislabeled pixels and [34] is not suitable for image-level supervisions. For the third strategy (w/ detection and w/ correction), our approach achieves the best segmentation result of 67.5%. This clearly validates the effectiveness of the proposed detection and correction method.

Besides, it should be interesting to investigate what happens if the selected clean labels are all correct. So we use the ground truth labels to replace the labels of the pixels which are selected by detection strategy and use these ground truth labels to train GAT (hereinafter referred to as the "oracle"). Surprisingly, the segmentation result outperforms the fully-supervised model (68.2% vs. 67.6%). This result also verifies that the key step for seed-and-expansion methods is to generate more accurate seed regions, thus leading to better

TABLE I

PER-CLASS IOU ON PASCAL VOC 2012 VALIDATION AND TEST SET WITH DIFFERENT  $\mathcal{D}^c$  USING DEEPLAB V1-VGG16 SEGMENTATION NETWORK.

$\mathcal{D}^c$	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Results on validation set:																						
200	90.0	76.6	29.2	80.6	55.4	64.2	80.0	73.9	80.5	29.1	70.6	49.5	75.2	68.1	65.8	75.9	41.0	74.9	39.7	61.7	56.1	<b>63.7</b>
400	90.2	76.6	29.6	80.2	55.8	64.8	80.1	74.3	80.2	29.2	70.8	49.4	73.7	68.0	65.7	76.4	46.0	72.9	38.6	63.5	58.6	<b>64.0</b>
800	90.3	76.3	30.4	80.8	55.9	66.8	81.5	73.6	81.4	30.9	74.3	49.3	75.5	69.6	66.2	77.1	45.6	74.4	39.3	66.3	58.3	<b>64.9</b>
1464	92.1	80.9	32.6	80.8	67.3	68.1	83.3	77.2	79.4	33.0	71.5	54.3	74.0	71.5	69.8	79.3	44.7	76.3	45.6	74.6	62.1	<b>67.5</b>
Results on test set:																						
200	90.3	74.4	33.6	81.3	54.9	60.4	77.5	74.0	80.0	26.0	63.7	53.9	75.6	68.6	76.3	74.7	46.6	76.9	43.8	61.6	54.2	<b>64.2</b>
400	90.7	75.7	33.1	80.4	55.8	61.1	77.0	74.2	82.8	25.9	64.5	55.8	76.2	69.3	76.4	75.3	49.4	75.3	45.9	65.0	55.5	<b>65.0</b>
800	90.8	78.1	32.9	78.0	56.1	61.2	78.0	73.8	82.4	26.4	68.6	57.3	76.9	71.9	76.3	76.9	45.6	77.3	48.3	64.6	56.0	<b>65.6</b>
1464	92.4	78.9	38.8	78.5	59.7	63.7	83.1	78.4	80.0	27.1	70.6	59.9	75.3	72.1	79.5	77.9	50.4	82.1	46.2	71.5	59.2	<b>67.9</b>

TABLE II

PER-CLASS IOU ON PASCAL VOC 2012 VALIDATION AND TEST SET WITH DIFFERENT  $\mathcal{D}^c$  USING DEEPLAB V2-RESNET101 SEGMENTATION NETWORK.

$\mathcal{D}^c$	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Results on validation set:																						
200	91.6	83.0	35.5	79.5	62.6	71.8	87.1	80.9	85.1	33.9	75.6	54.6	78.9	72.9	75.9	82.8	50.4	72.5	40.7	76.2	67.3	<b>69.5</b>
400	91.8	83.5	36.1	79.4	62.8	72.9	88.2	81.3	86.6	33.6	78.5	57.2	80.5	74.7	72.5	81.5	52.1	74.4	42.5	74.7	59.7	<b>69.7</b>
800	92.1	84.1	37.3	80.5	62.4	72.8	88.5	80.0	86.3	31.9	79.7	59.7	80.4	79.2	76.1	82.4	56.8	80.1	43.8	78.5	64.8	<b>71.3</b>
1464	94.4	89.0	56.4	84.9	71.9	76.5	93.2	85.6	89.0	35.6	86.6	62.8	84.3	85.9	83.0	86.4	53.9	86.6	49.5	87.9	75.8	<b>77.1</b>
Results on test set:																						
200	91.9	85.0	35.8	84.1	58.1	67.1	88.7	79.2	85.4	34.1	76.1	58.2	82.3	78.1	79.4	78.7	48.7	78.4	56.5	70.8	56.0	<b>70.1</b>
400	92.0	86.4	36.3	85.9	57.6	68.4	87.6	79.4	82.5	34.6	73.7	57.6	81.0	76.6	82.6	78.6	43.2	77.3	58.8	73.8	61.3	<b>70.2</b>
800	92.5	85.7	36.7	79.9	57.5	67.6	88.1	79.2	88.3	35.8	78.7	62.3	83.0	80.6	83.1	80.0	54.1	80.4	59.8	73.2	61.4	<b>71.8</b>
1464	94.6	90.4	52.2	87.7	61.8	74.8	93.3	84.9	90.0	35.3	85.1	66.4	87.9	88.1	84.4	85.3	56.8	88.1	58.7	84.0	70.7	<b>77.2</b>

TABLE III

COMPARISONS ON PASCAL VOC 2012 *val* AND TEST SETS.

Methods	Backbone	<i>val</i>	test
Supervision: 10.6 k pixel-level			
Deeplab	v1-vgg16	67.6	68.6
Deeplab	v2-resnet101	76.5	79.7
Supervision: 1.4k pixel-level + 9k image-level			
GAIN [18]	v1-vgg16	60.5	62.1
DSRG [16]	v1-vgg16	64.3	-
WSSL [19]	v1-vgg16	64.6	66.2
MDC [5]	v1-vgg16	65.7	67.6
FickleNet [8]	v1-vgg16	65.8	-
Ouali <i>et al.</i> [21]	resnet50	73.2	-
Luo <i>et al.</i> [20]	v2-resnet101	76.6	77.1
Ours	v1-vgg16	67.5	67.9
Ours	v2-resnet101	<b>77.1</b>	<b>77.2</b>

TABLE IV

SEGMENTATION RESULTS ON PASCAL VOC 2012 *val* SET WITH DIFFERENT SIZES OF  $\mathcal{D}^c$ .

$\mathcal{D}^c$	200	400	800	1464	Fully-supervised
v1-vgg16	46.8	52.9	56.6	62.5	67.6
Ours	63.7	64.0	64.9	67.5	
v2-resnet101	58.3	63.0	68.1	74.6	76.5
Ours	69.5	69.7	71.3	77.1	

segmentation performance. Moreover, our proposed graph-based correction method can further correct pixel-level label noise efficiently. Some visual results are given in Figure 4.

**Impact of superpixels and CRF.** To investigate the impact of the number of superpixels and CRF, we conduct the experiments by over-segmenting one image into

TABLE V

PERFORMANCE OF THE PSEUDO LABELS DURING DIFFERENT STAGES.

Methods	w/ detection	w/ correction	9K	10.6K
initial CAM	-	-	49.7	57.4
	✓	-	59.8	62.8
	-	✓	56.9	59.6
	✓	✓	65.6	67.5
Oracle	✓	✓	66.3	68.2

TABLE VI

PERFORMANCE OF THE PSEUDO LABELS USING DIFFERENT NUMBER OF SUPERPIXELS AND CRF.

Superpixel	Pseudo label		Segmentation	
	w/o CRF	w/ CRF	w/o CRF	w/ CRF
500	66.1	67.5	67.1	67.2
1000	67.2	68.3	67.5	67.5
5000	68.7	69.2	67.6	67.1

500, 1000 and 5000 superpixels with and without CRF respectively. We observe the quality of pseudo labels in mIoU on the *trainaug* set and the segmentation results on the *val* set using Deeplab v1-vgg16 network. The results are shown in Table VI. From the results we can see that the mIoU of pseudo labels increases with the increased number of superpixels, and using dense CRF can further refine them. Meanwhile, the segmentation result is optimal when the images are over-segment into 5000 superpixels and the pseudo labels without using CRF, which is equal to the performance of fully-supervised model. Considering the efficiency and performance, one image is over-segment

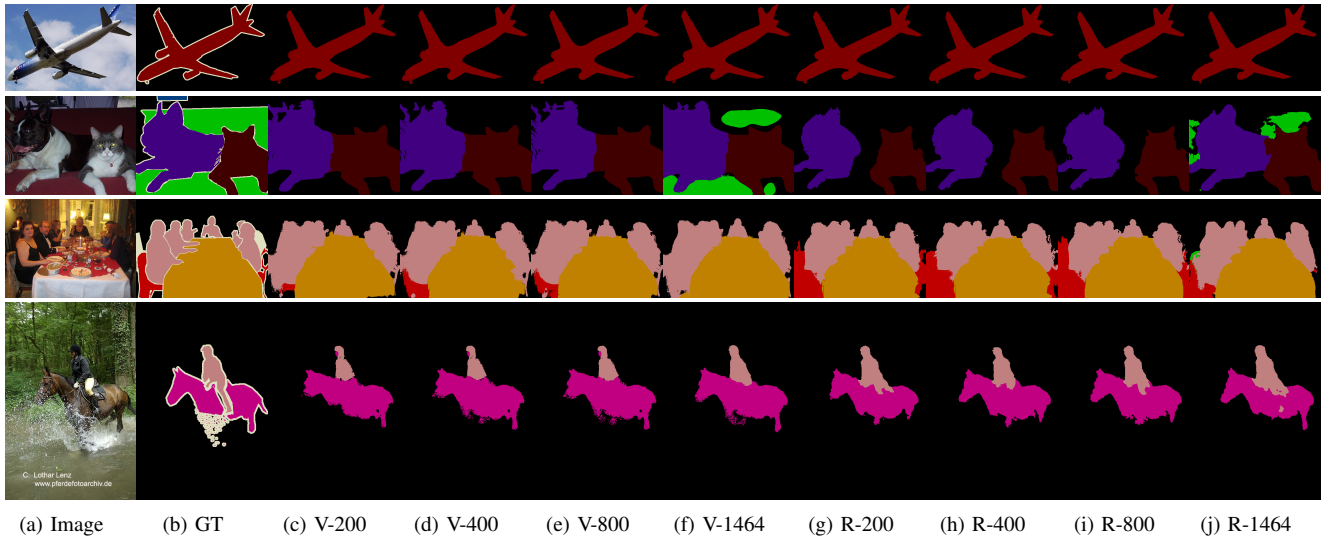


Fig. 3. Qualitative results on the PASCAL voc 2012 *val* set using Deeplab v1-vgg16 and Deeplab v2-resnet101 segmentation network. (a) Input images. (b) Ground truth. (c)-(f) are the segmentation results on  $\mathcal{D}^c=200, 400, 800$  and 1464 using Deeplab v1-vgg16 network. (g)-(j) are the segmentation results on  $\mathcal{D}^c=200, 400, 800$  and 1464 using Deeplab v2-resnet101 network.

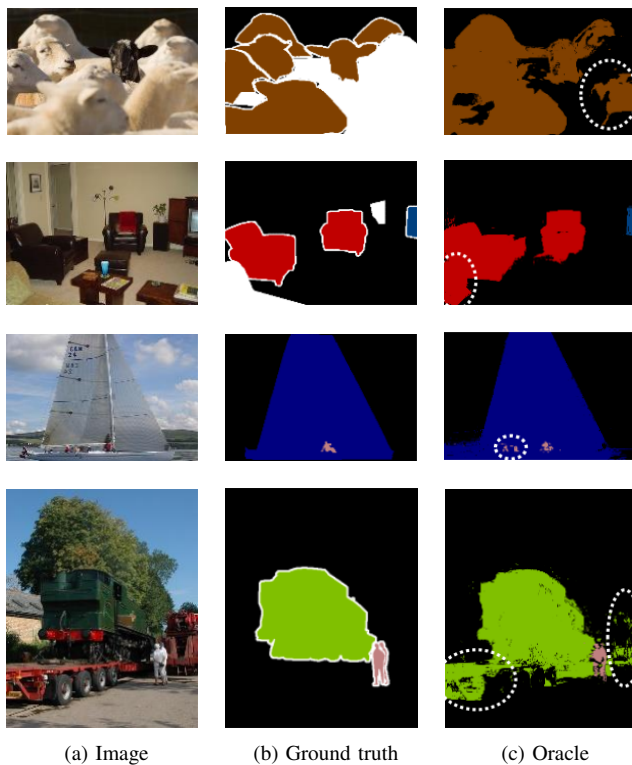


Fig. 4. Qualitative results of pseudo labels on the PASCAL VOC 2012 *trainaug* set using oracle strategy, which can successfully correct the label for missing objects in these images (marked by ellipses).

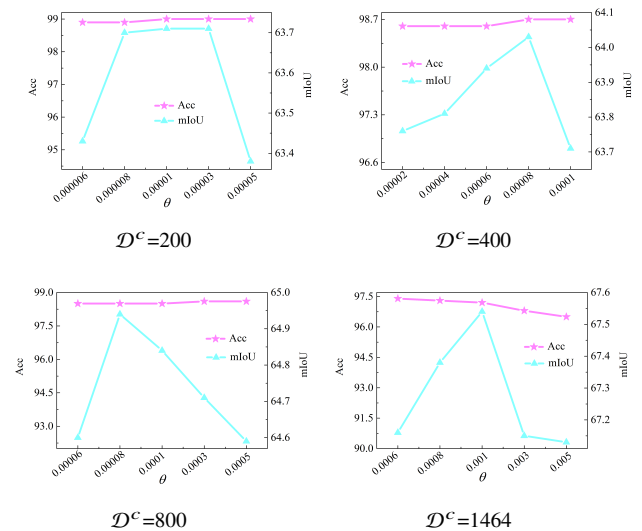


Fig. 5. The accuracy of the selected clean labels and the mIoU of the segmentation results under different  $\theta$ .

into about 1000 superpixels in our experiments.

**Impact of parameter  $\theta$ .** The parameter  $\theta$  is the key to select clean labels from initial pixel-level noisy labels, so we investigate the impact of  $\theta$  on different splits using Deeplab v1-vgg16 segmentation network. Since we have the ground truth labels on  $\mathcal{D}^c$ , the accuracy of the selected clean labels are measured on  $\mathcal{D}^c$ , and the segmentation results are also reported on the validation set. As shown in Figure 5, we select  $\theta$  from 0.0006 to 0.005 on strong/weak split of 1.4K/9K dataset, the accuracy of the selected clean labels evaluated on 1.4K subset drops from 97.4% to 96.5%, and the segmentation result first rises from 67.16% to 67.54%, and then drops to 67.13%. When  $\theta = 0.001$ , the pixel



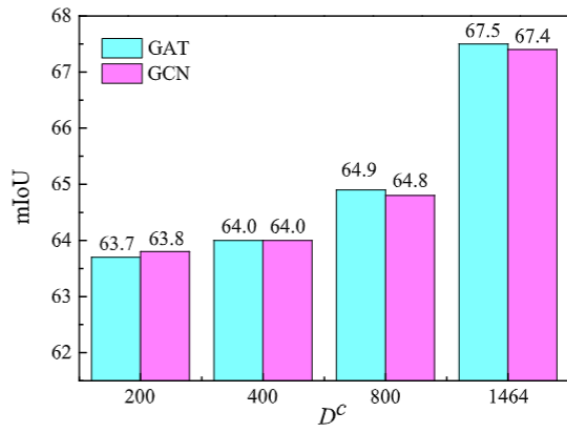


Fig. 6. The comparison of segmentation results using GCN or GAT to perform noisy labels correction.

TABLE VII  
SEGMENTATION RESULTS ON PASCAL-CONTEXT TEST SET WITH DIFFERENT SIZES OF  $\mathcal{D}^c$  USING DEEPLAB V1-VGG16 SEGMENTATION NETWORK.

$D_c$	625	1250	Fully-supervised
DeepLab	30.8	35.0	38.6
Ours	35.8	37.3	

accuracy achieves 97.2% and the segmentation result also reaches 67.54%. Therefore,  $\theta$  is set to 0.001 on 1.4k/9k split. Note that we can use this strategy to select a suitable threshold on different settings. In our experiments,  $\theta$  takes 0.00003, 0.00008 and 0.00008 under the setting of  $\mathcal{D}^c=200, 400$  and 800 respectively.

**Correcting noisy labels by GAT vs. GCN.** To investigate the performance of correcting noisy labels using different graph neural networks, we apply GCN instead of GAT to perform noisy label correction. Figure 6 shows the comparison results. The segmentation results which using GCN are 63.8%, 64.0%, 64.8% and 67.4% under the setting of  $\mathcal{D}^c=200, 400, 800$  and 1464. It also achieves a performance improvement of 4.9% to 17% over the baseline for different sizes splits. Overall, GAT slightly achieves better performance than that of GCN in most settings.

### B. Results on PASCAL-Context dataset

Our approach successfully generalizes to the whole scene parsing PASCAL-Context dataset using Deeplab v1-vgg16 segmentation network. Table VII shows the performance on two splits (1/8 and 1/4 strongly annotated images) on PASCAL-Context dataset. Although this dataset is smaller and more difficult than PASCAL VOC 2012, there is still an improvement over the baseline of 5.0% and 2.3% for the 1/8 and 1/4 splits, respectively. The qualitative results are shown in Figure 7.

### C. Results on MS-COCO dataset

To further evaluate our proposed method, we conduct experiments on MS-COCO dataset using Deeplab v1-vgg16

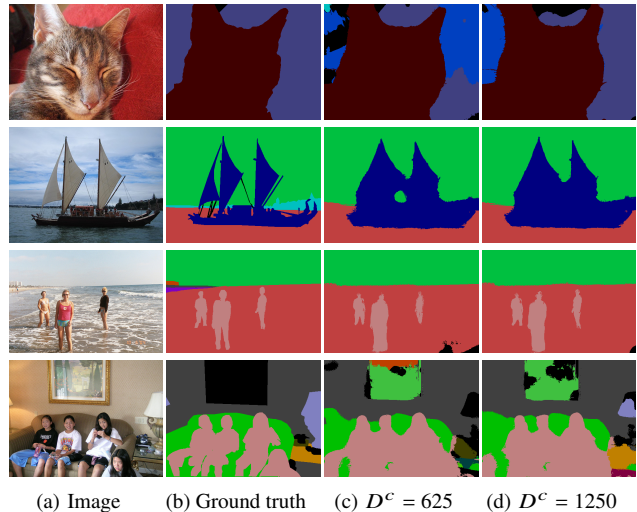


Fig. 7. Qualitative results on the PASCAL-Context test set using Deeplab v1-vgg16 segmentation network. (a) Input images. (b) Ground truth. (c) and (d) are the segmentation results on  $\mathcal{D}^c = 625$  and 1250.

TABLE VIII  
SEGMENTATION RESULTS ON MS-COCO *val* SET WITH DIFFERENT SIZES OF  $\mathcal{D}^c$  USING DEEPLAB V1-VGG16 SEGMENTATION NETWORK.

$D_c$	1293	10348	20696	Fully-supervised
DeepLab	23.5	29.1	29.3	29.5
Ours	29.6	31.5	31.6	

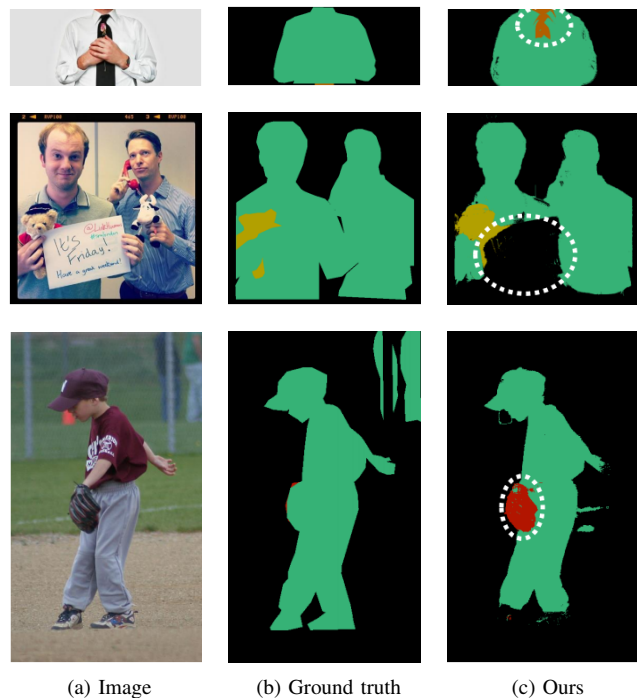


Fig. 8. Qualitative results on the MS-COCO training set, our method can successfully correct the labels for missing objects in ground truth images (marked by ellipses).

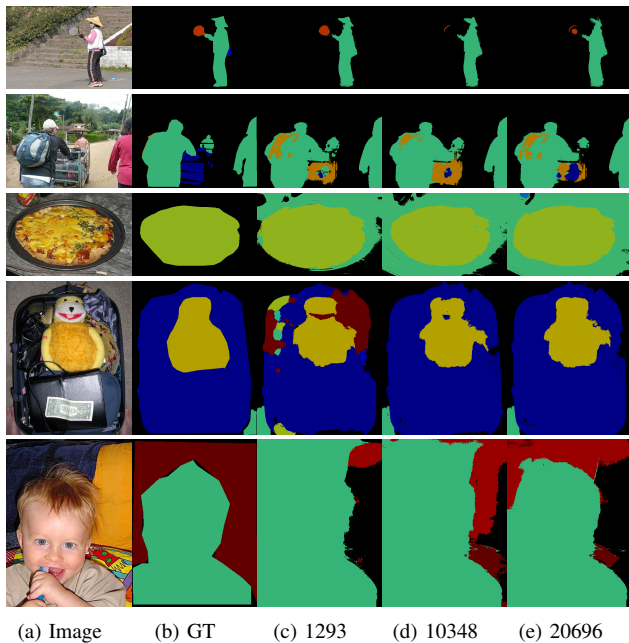


Fig. 9. Qualitative results on the MS-COCO *val* set using Deeplab v1-vgg16 segmentation network. (a) Input images. (b) Ground truth. (c)-(e) are the segmentation results on  $D^c = 1293, 10348$  and 20696.

segmentation network. Table VIII shows the performance on three splits (1/64, 1/8 and 1/4 strongly annotated images). Although this dataset is collected from a complex natural context, there is still an improvement over the baseline of 6.1%, 2.4% and 2.3% for the 1/64, 1/8 and 1/4 splits, respectively. More surprisingly, our semi-supervised setting outperforms the fully-supervised model only using 1/64 of fully-labeled images. The qualitative results of the pseudo labels generated by our method are shown in Figure 8. Due to the rough labeling of the MS-COCO dataset, the pseudo labels corrected by our method successfully highlight some instances which are improperly labeled in ground truth annotations, thus resulting in better performance. This result clearly demonstrates the effectiveness of the proposed noise label learning framework. More qualitative segmentation results are shown in Figure 9. For example, in the second row, the segmentation network trained with our generated pseudo labels can partially mark the regions belonging to class of “backpack” (marked as orange), which are improperly labeled in ground truth annotations. Similarly, in the third row, our proposed method can also successfully mark the class of “dining table” (marked as green) mislabeled by ground truth. These results clearly demonstrate the effectiveness of our method.

## V. CONCLUSION

In this paper, we present a novel perspective for semi-supervised semantic segmentation and formulate the task as a problem of learning with pixel-level label noise. We design a label noise detection and correction framework, which transforms an image to a graph-structured representation and

leverage GAT to address the noisy labels in pixel-level task. The experiments show that our method outperforms previous methods and achieves the state-of-the-art performance on PASCAL VOC 2012, PASCAL-Context and MS-COCO datasets in the semi-supervised setting.

## REFERENCES

- [1] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, “Scribblesup: Scribble-supervised convolutional networks for semantic segmentation,” in *CVPR*, pp. 3159–3167, 2016.
- [2] P. Vernaza and M. Chandraker, “Learning random-walk label propagation for weakly-supervised semantic segmentation,” in *CVPR*, pp. 7158–7166, 2017.
- [3] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple does it: Weakly supervised instance and semantic segmentation,” in *CVPR*, pp. 876–885, 2017.
- [4] J. Ahn and S. Kwak, “Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation,” in *CVPR*, pp. 4981–4990, 2018.
- [5] 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 *CVPR*, pp. 7268–7277, 2018.
- [6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *CVPR*, pp. 2921–2929, 2016.
- [7] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *CVPR*, pp. 1568–1576, 2017.
- [8] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, “Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference,” in *CVPR*, pp. 5267–5276, 2019.
- [9] J. Ahn, S. Cho, and S. Kwak, “Weakly supervised learning of instance segmentation with inter-pixel relations,” in *CVPR*, pp. 2209–2218, 2019.
- [10] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” Citeseer, 2009.
- [11] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in *CVPR*, pp. 2691–2699, 2015.
- [12] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in *NeurIPS*, pp. 8535–8545, 2018.
- [13] J. Huang, L. Qu, R. Jia, and B. Zhao, “O2u-net: A simple noisy label detection approach for deep neural networks,” in *ICCV*, pp. 3326–3334, 2019.
- [14] J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning,” in *ICLR*, 2020.
- [15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” in *ICLR*, 2018.
- [16] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, “Weakly-supervised semantic segmentation network with deep seeded region growing,” in *CVPR*, pp. 7014–7023, 2018.
- [17] A. Kolesnikov and C. H. Lampert, “Seed, expand and constrain: Three principles for weakly-supervised image segmentation,” in *ECCV*, pp. 695–711, 2016.
- [18] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, “Tell me where to look: Guided attention inference network,” in *CVPR*, pp. 9215–9223, 2018.
- [19] G. Papandreou, L.-C. Chen, K. Murphy, and A. Yuille, “Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. arxiv, 2015,” *arXiv preprint arXiv:1502.02734*.
- [20] W. Luo and M. Yang, “Semi-supervised semantic segmentation via strong-weak dual-branch network,” in *ECCV*, 2020.
- [21] Y. Ouali, C. Hudelot, and M. Tami, “Semi-supervised semantic segmentation with cross-consistency training,” in *CVPR*, June 2020.
- [22] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *ICML*, pp. 4334–4343, 2018.
- [23] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” in *NeurIPS*, pp. 5049–5059, 2019.
- [24] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, “Curriculumnet: Weakly supervised learning from large-scale web images,” in *ECCV*, pp. 135–150, 2018.

- [25] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *CVPR*, pp. 5051–5059, 2019.
- [26] J. Zhang, J. Xie, and N. Barnes, "Learning noise-aware encoder-decoder from noisy labels by alternating back-propagation for saliency detection," in *ECCV*, pp. 349–366, 2020.
- [27] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *CVPR*, pp. 1237–1246, 2019.
- [28] Y. Longrong, M. Fanman, L. Hongliang, W. Qingbo, and C. Qishang, "Learning with noisy class labels for instance segmentation," in *ECCV*, 2020.
- [29] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.
- [30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
- [31] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, pp. 3844–3852, 2016.
- [32] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.
- [33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," in *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [34] M. Pu, Y. Huang, Q. Guan, and Q. Zou, "Graphnet: Learning image pseudo annotations for weakly-supervised semantic segmentation," in *ACM MM*, pp. 483–491, 2018.
- [35] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *ICCV*, pp. 4654–4662, 2019.
- [36] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *ICCV*, pp. 10313–10322, 2019.
- [37] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with local-global long short-term memory," in *CVPR*, pp. 3185–3193, 2016.
- [38] X. Lin, Q. Zou, and X. Xu, "Action-guided attention mining and relation reasoning network for human-object interaction detection," in *IJCAI*, pp. 1104–1110, 2020.
- [39] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *International Conference on Machine Learning*, pp. 312–321, PMLR, 2019.
- [40] R. Vieux, J. Benois-Pineau, J.-P. Domenger, and A. Braquelaire, "Segmentation-based multi-class semantic object detection," *Multimedia Tools and Applications*, vol. 60, no. 2, pp. 305–326, 2012.
- [41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [42] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NeurIPS*, pp. 109–117, 2011.
- [43] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [44] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtaasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *CVPR*, pp. 891–898, 2014.
- [45] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, pp. 740–755, 2014.
- [46] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, pp. 991–998, IEEE, 2011.
- [47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.
- [48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [50] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016.
- [52] M. S. Ibrahim, A. Vahdat, M. Ranjbar, and W. G. Macready, "Semi-supervised semantic image segmentation with self-correcting networks," in *CVPR*, 2020.