Semi-Supervised Semantic Segmentation under Label Noise via Diverse Learning Groups

Peixia Li[1,2], Pulak Purkait[2], Thalaiyasingam Ajanthan [2], Majid Abdolshah [2], Ravi Garg[2,4], Hisham Husain [2], Chenchen Xu[2,3], Stephen Gould[3], Wanli Ouyang [1], Anton van den Hengel [2,4]
[1] The University of Sydney, [2] Amazon,
[3] Australian National University, [4] The University of Adelaide

Abstract

Semi-supervised semantic segmentation methods use a small amount of clean pixel-level annotations to guide the interpretation of a larger quantity of unlabelled image data. The challenges of providing pixel-accurate annotations at scale mean that the labels are typically noisy, and this contaminates the final results. In this work, we propose an approach that is robust to label noise in the annotated data. The method uses two diverse learning groups with different network architectures to effectively handle both label noise and unlabelled images. Each learning group consists of a teacher network, a student network and a novel filter module. The filter module of each learning group utilizes pixel-level features from the teacher network to detect incorrectly labelled pixels. To reduce confirmation bias, we employ the labels cleaned by the filter module from one learning group to train the other learning group. Experimental results on two different benchmarks and settings demonstrate the superiority of our method over state-of-the-art approaches.

1. Introduction

Semantic segmentation is a critical step in many computer vision tasks, and one that, like so many, is now predominantly achieved through deep learning [5, 35]. A key factor in the success of these methods has been access to high-quality annotated datasets. Pixel-accurate annotation of segmentation data is challenging, and becomes impractical as datasets grow. Unlabelled data is abundant and thus semi-supervised semantic segmentation has received increasing attention [15, 20, 8]. The predominant semi-supervised approach employs a teacher-student framework to allocate pseudo labels to unlabelled data [11, 23, 36, 40]. These methods rely on the availability of a small but perfectly annotated dataset, however. This is a challenge in many practical situations where annotations are collected via crowd-sourcing (*e.g.*, Sagemaker MTurk), or automated labelling techniques [31, 3], and are thus inevitably noisy.

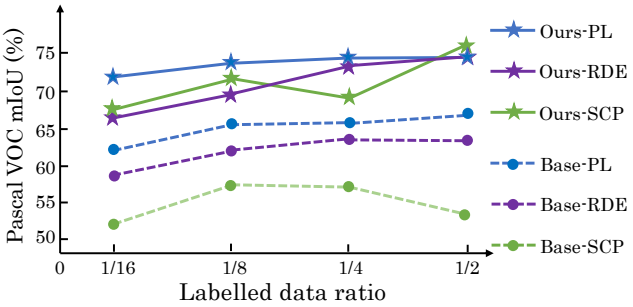


Figure 1: Existing semi-supervised semantic segmentation models are vulnerable to the inevitable label noise in the pixel-level annotations. The proposed method performs significantly better than the current state-of-the-art [20] in the presence label noise. We apply three types of noise (labelled PL, RDE and SCP in Sec. 5.2) to achieve a 9% error rate in the labelled data.

We consider here the practically important case whereby semi-supervised semantic segmentation is to be applied despite the fact that the provided pixel labels contain noise. We propose an approach that is robust to this noise to the extent that it outperforms the baselines by a significant margin (see Fig. 1), despite being relatively simple to implement and incurring no extra computation cost at test time.

It has been shown recently [19] that supervised segmentation models trained on noisy pixel-level labels first fit the clean labels during an "early-learning" phase before eventually memorizing the noisy labels. The above learning dynamics are exploited in [19] for weakly supervised segmentation by adding a multi-scale consistency loss. However, no noise detection module is used in [19] and the effect of unsupervised consistency loss on the label noise is weak.

Recently there has been a surge of interest in data-centric methods [38, 39] that utilize the k-NN of the feature representation of the data-points to detect annotation noise in classification datasets. A straightforward extension of this approach to semantic segmentation would lead to computing the k-NN of the feature representation at pixel-level over all images, which would be infeasible. In this work, we

adapt [39] to semi-supervised semantic segmentation by introducing an explicit label noise detection module that predicts label noise at the pixel level without significant additional memory or computation requirements.

Pseudo labelling approaches [11, 23, 36, 2] for semi-supervised segmentation often use a single model to generate pseudo labels on the unlabelled data as well as train on them. This leads to confirmation bias as the model can overfit to its prediction errors. The same bias is apparent even when a single model is employed to filter label noise and train on those cleaned labels [19]. To mitigate this effect, we introduce two diverse learning groups with different network architectures that encourage mutual knowledge distillation and effective noise filtering. These learning groups also exploit complementary learning paths that boost the competency of each group. Our contributions are thus summarized as follows:

- We propose an approach to semi-supervised semantic segmentation that is robust to label noise.
- We present a general architecture that maintains two diverse learning groups to overcome confirmation bias in label assignment problems.
- We introduce a filter module that extends [39] for semantic segmentation while demonstrating a good trade-off between accuracy and computation cost.
- We conduct extensive experiments showing that our method significantly outperforms semi/weakly-supervised semantic segmentation baselines regardless of the presence or the type of label noise.

2. Related Work

Semi-supervised Semantic Segmentation. Pixel-level annotation for semantic segmentation is time-consuming, costly, and error prone. Semi-supervised learning addresses the first two elements of this problem using unlabelled data. These approaches can be broadly categorized into self-training (a.k.a self-learning) based methods [11, 23, 10, 25, 29] and consistency regularization-based methods [40, 26, 6, 14, 20]. Self-training-based methods use models trained on labelled examples to predict pseudo-segmentation masks over the unlabelled examples, thus iteratively expanding the training set. The effectiveness of these methods depends on accurately identifying high-confidence pseudo labels via crafted auxiliary networks [11, 23, 36] or GANs [10, 25, 29]. Consistency regularization methods enforce consistency among predictions under varying perturbations, where the perturbations can be applied to the input image (French [8], PseudoSeg [40], CAC [16]), the intermediate feature representations (CCT [26]) or the network itself (CPS [6], GCT [14]). Most effort in this vein aims to identify perturbations that are challenging enough to support generalization without changing the underlying true labels. Particularly, recent work [20] on Mean Teacher [30] demonstrates superior performance and flexibility in the choice of perturbations. Our work builds upon [20] and introduces a general framework with two learning groups and a noise filter module to effectively learn from noisy labelled data and unlabelled data.

Semantic Segmentation with Label Noise. Segmentation is a more challenging task than classification, and the labelled data is more prone to label noise. To achieve robustness some methods explicitly model the human annotation errors [12], or add a regularization term [28, 32]. Interestingly, [22] shows how high-level structure information (about microscopy images) learned by the model can be used as additional signal to remedy wrong annotations. In [24], a two stream mutual-attention network is devised to smooth out the influence of noisy labels during backpropagation. ADELE [19] exploits the learning dynamics demonstrated by noisy examples and proposes an early-learning strategy accordingly. These supervised segmentation methods assume all of the data is labellebd and that the majority of the labels are correct. They thus focus on alleviating the influence a small set of incorrect labels.

The approach most similar to that proposed here is that of DivideMix [18] whereby two divergent networks are applied to separate data partitions. The method there is specifically designed for classification, however, as it is based on an image-level Gaussian Mixture Model. We propose here instead a pixel-level filter module appropriate for semantic segmentation. Additionally, in contrast to [18], we employ multiple learning groups with varying architectures to encourage diversity.

3. Preliminaries

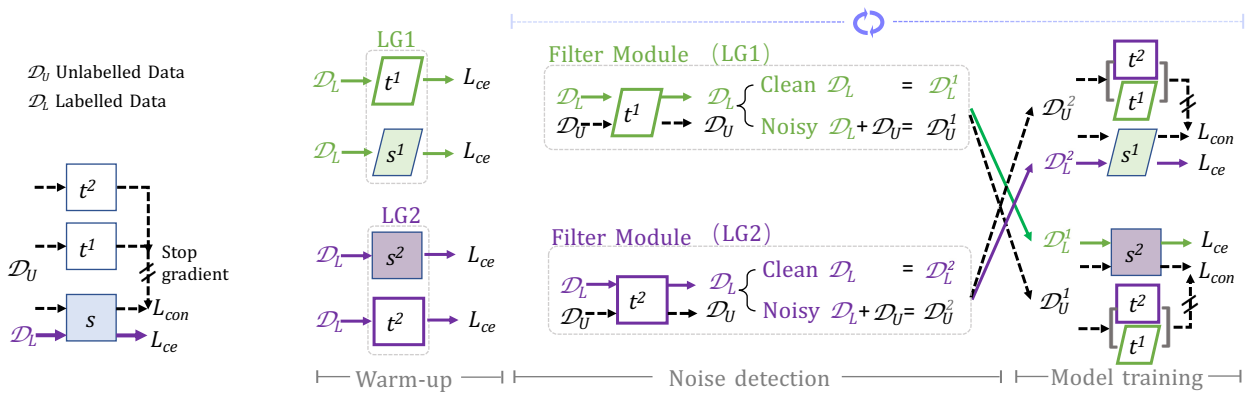
3.1. Problem Setup

We consider semi-supervised segmentation in the case where we have a small set of potentially noisy labelled examples and a large set of unlabelled examples. Let $\mathcal{D}_L = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_L}$ be the labelled set where $\boldsymbol{x}_i \in \mathcal{X} := [0,1]^{h \times w \times 3}$ and $\boldsymbol{y}_i \in \mathcal{Y} := \mathcal{C}^{h \times w}$ denote an image and the corresponding pixel-wise segmentation mask respectively. Here, $\mathcal{C} = \{1, 2, \ldots, \ell\}$ denotes the set of labels. Similarly, let $\mathcal{D}_U = \{\boldsymbol{x}_i\}_{i=1}^{N_U}$ denote the unlabelled set of images.

In contrast to the standard semi-supervised segmentation problem, the segmentation masks $\{\boldsymbol{y}_i\}$ may contain erroneous labels.

3.2. Noisy Label Detection

Here we briefly review the noisy label detection approach SimiFeat [38] developed for robust image classification. Our work extends this framework to the segmentation problem by pixel-level reasoning over noisy labels in



a) Baseline model b) Our pipeline

Figure 2: The pipelines of a) the baseline model [20] and b) the proposed approach. In contrast to the baseline that uses two teachers and a single student network, we propose a framework with two diverse Learning Groups (LGs) (t^1, s^1) and (t^2, s^2) where each group contains a teacher and a student network. The two learning groups propagate complementary information and reduce the confirmation bias. Here, t denotes a teacher model, s is a student model, solid lines show the forward flow of labelled data, and dotted lines show the flow of unlabelled data. L_{ce} denotes the cross-entropy loss and L_{con} is the confidence-weighted cross-entropy loss.

Sec. 4.2. SimiFeat estimates the label noise proportion for each class and then compute a per-class "noisiness" score for all the images in order to rank them and eliminate top R_c percentile of images, where R_c is the noise percentage for class c.

Let $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}$ be a dataset of images $\boldsymbol{x}_i \in \mathcal{X}$ with noisy labels $\tilde{y}_i \in \mathcal{C}$, and $g(\cdot) : \mathcal{X} \to \mathbb{R}^d$ be a feature extractor. The noise proportion is computed from the prior probability of clean labels and the noise transition probability, which are estimated via High Order Consensus (HOC) (refer to [38] for more details) as follows:

$$\mathbb{P}(Y), \mathbb{P}(\tilde{Y}|Y) \leftarrow \text{HOC}\left(\{(g(\boldsymbol{x}), \tilde{y}) \mid (\boldsymbol{x}, \tilde{y}) \in \tilde{\mathcal{D}}\}\right), \quad (1)$$

where Y and $\tilde{Y}$ denote the random variables corresponding to clean and noisy labels respectively. The noise proportion is then calculated for a class $c \in \mathcal{C}$ as:

$$R_c = 1 - \mathbb{P}(Y = c|\tilde{Y} = c)\,. \quad (2)$$

Note, $\mathbb{P}(Y|\tilde{Y})$ is obtained via Bayes' rule:

$$\mathbb{P}(Y|\tilde{Y}) = \mathbb{P}(\tilde{Y}|Y)\mathbb{P}(Y)/\mathbb{P}(\tilde{Y})\,, \quad (3)$$

where $\mathbb{P}(\tilde{Y})$ is computed from the observed labels and $\mathbb{P}(Y), \mathbb{P}(\tilde{Y}|Y)$ are computed using HOC (Eq. (1)).

To compute the noisiness score, a k-NN based soft label is employed in the feature space via nearest-neighbour voting. Then, the score for an image is determined as the negative cosine similarity between the soft label of that image and the one-hot encoding of the class label [38]. Finally, the images are ranked based on the score and the noise proportion from Eq. (2) is utilized to determine the percentile of images to be declared as noisy and consequently removed.

4. Our Approach

We introduce a general pipeline with two Learning Groups (LGs), as shown in Fig. 2. Each LG consists of a teacher-student framework and a noisy label detection module (Filter Modules). The teacher network is used to produce pseudo labels for the unlabelled data similar to the semi-supervised learning pipeline in [20]. To stabilize training, we first train both LGs independently in a fully-supervised manner with the labelled dataset $\mathcal{D}_L$ for a few epochs. After this warm-up, at each training epoch, we activate the filter module that detects the pixels with noisy labels in the labelled set $\mathcal{D}_L$ and move them to the unlabelled set $\mathcal{D}_U$. Then, we train the pipeline in the semi-supervised manner. The noise detection and model training are repeated until the end of the training.

Our filter module uses the feature representation of the teacher model from one LG to partition the labelled data into clean and noisy sets. The filter module then ignores the labels of the noisy set and moves the corresponding elements into the unlabelled set to obtain a revised labelled and unlabelled dataset pair. The updated dataset pair generated by one LG is consumed by the other. As training progresses each LG receives a different set of dataset pair due to the decoupled noise filtering, which enables complementary information to propagate in the two learning paths – reducing confirmation bias. This, along with the different network architectures for the two LGs, enhances the performance of our framework in the presence of noise.

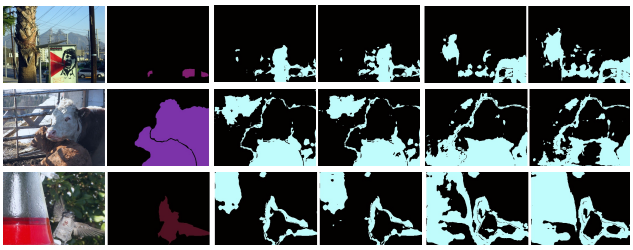


Figure 3: The noise detection results from the two LGs. The images from left to right are: input, clean ground-truth followed by noise detection results from the two LGs when we use the same architecture (two R50s) and different architectures (R50 and PVTv2-B2 [33]). We get more diverse noise detection results when using different architectures.

4.1. Learning Group

The architecture of the LG used in our pipeline is inspired by the recent teacher-student framework [20] as depicted in Fig. 2 (a). We refer the reader to [20] for more details.

At a training epoch k, the first LG receives the revised labelled set $\mathcal{D}^2_{L,k}$ and unlabelled set $\mathcal{D}^2_{U,k}$ after filtering the noisy labels predicted by the second LG (details of $\mathcal{D}^2_{L,k}$ and $\mathcal{D}^2_{U,k}$ can be found in Sec. 4.2). Then the student network of the first LG minimizes the following loss:

$$\mathcal{L}(\mathcal{D}^2_{L,k}, \mathcal{D}^2_{U,k}, \boldsymbol{\theta}_{s^1}) \coloneqq L_{ce}(\mathcal{D}^2_{L,k}, \boldsymbol{\theta}_{s^1}) + \alpha\, L_{con}(\mathcal{D}^2_{U,k}, \boldsymbol{\theta}_{s^1})\,. \tag{4}$$

Here, L_{ce} denotes the cross-entropy loss on the labelled data, L_{con} is a confidence-weighted cross-entropy loss that enforces consistency between the teacher and student model predictions on the unlabelled data, $\boldsymbol{\theta}_{s^1}$ denotes the parameters of the student network in the first LG, and $\alpha > 0$ provides a relative weight between the two losses. Analogously, the parameters of the second student network $\boldsymbol{\theta}_{s^2}$ are optimized utilizing the revised labelled set $\mathcal{D}^1_{L,k}$ and the unlabelled dataset $\mathcal{D}^1_{U,k}$ predicted by the first LG. The teacher network is updated by the moving average of the student network parameters of the same LG.

Note that, at each training epoch, the set of labelled data varies depending on the filter module output. Specifically, the detected pixels with noisy labels will be removed from the label set, which leads to modifying the segmentation mask of the labelled examples with an "ignore-pixel" label that enables the loss Eq. (4) to be defined in terms of L_{con} instead of L_{ce} at those pixels.

To encourage diversity of the LGs, we utilize R50 [9] for the first LG and PVTv2-B2 [33] for the second. In Fig. 3, we show that the noise detection results from two R50 (the third and fourth columns) have less diversity than those from different network architectures (the last two columns).[1] We also show that two different architectures bring better performance in our experiments (Fig. 7).

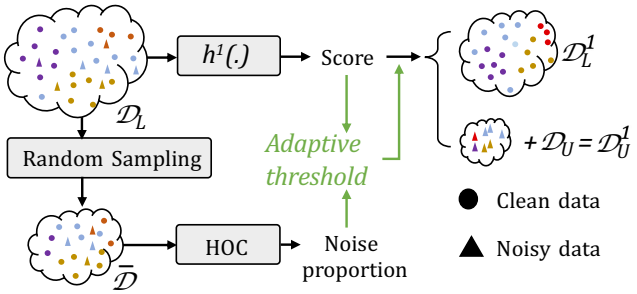


Figure 4: The architecture of our filter module. First, we generate a representative dataset $\bar{\mathcal{D}}$ and calculate the noise proportion based on HOC [39]. Then, we get the adaptive threshold based on the noise proportion and example score. According to the adaptive threshold and example score, the original dataset $\mathcal{D}_L$ is divided into a clean set $\mathcal{D}^1_L$ and a noisy set. The noisy set is added to the unlabelled dataset $\mathcal{D}_U$.

4.2. Filter Module

The task of the filter module is to detect incorrectly labelled pixels efficiently. To this end, rather than relying on label correction approaches [19], we design a filter module that accurately detects pixel-level noise scalably. The filter module is shown in Fig. 4.

Note that, according to Eq. (1), the HOC approach requires a dataset of feature encodings and the corresponding labels. Since semantic segmentation is a classification problem at the pixel-level, the natural extension is to treat each pixel as a datapoint and directly apply HOC. Unfortunately, this becomes intractable as the computational complexity is quadratic in number of pixels, labels, and number of images in the dataset.

To mitigate this, we use a small subset of pixels per class to generate a representative set $\bar{\mathcal{D}} = \{(\boldsymbol{x}_i, j, \tilde{y}_{i,j})\}$, which comprises of a fixed number of pixels per image subsampled from $\mathcal{D}_L$. Here, $\boldsymbol{x}_i \in \mathcal{X}$ is an image, $j \in \mathcal{I}(\boldsymbol{x}_i)$ is the index of a pixel in image $\boldsymbol{x}_i$, and $\tilde{y}_{i,j} \in \mathcal{C}$ is the label of the j-th pixel. In order to generate such a set, we found that random pixel sampling is more effective than elaborate methods such as k-means clustering, and is more computationally efficient also.

Let $g^1_j(\cdot) : \mathcal{X} \to \mathbb{R}^d$ denotes the backbone feature extractor, *i.e.*, the penultimate layer output for the first LG at pixel j (where $j \in \mathcal{I}(\boldsymbol{x}_i) \coloneqq \{1, \ldots, w \times h\}$ if images are resized to a fixed size). We construct the embedded repre-

[1] It should be noted that the two networks with R50 architecture are trained with different initialization independently on the labelled data for a warm start for a fair comparison.

sentative set $\bar{\mathcal{D}}^1_G \coloneqq \left\{(g^1_j(\boldsymbol{x}_i), \tilde{y}_{i,j}) \mid (\boldsymbol{x}_i, j, \tilde{y}_{i,j}) \in \bar{\mathcal{D}}\right\}$ and invoke HOC (Eq. (1)) to estimate the noise proportion $\{R^1_c\}_{c\in\mathcal{C}}$ at pixel-level (Eq. (2)). Analogously, the noise proportion $\{R^2_c\}_{c\in\mathcal{C}}$ for the other LG can be estimated by constructing $\bar{\mathcal{D}}^2_G$.

Let $h^1_j(\cdot) : \mathcal{X} \to [0,1]^\ell$ and $h^2_j(\cdot) : \mathcal{X} \to [0,1]^\ell$ be the teacher model mappings to softmax scores in the first and second LG respectively. We compute a noisiness score based on the teacher model predictions, as opposed to using k-NN in SimiFeat [38]. This eliminates the computation cost of running k-NN for every pixel over all images in $\mathcal{D}_L$. We define the score for the first LG at pixel j in image $\boldsymbol{x}_i$ as:

$$S^1(\boldsymbol{x}_i, j) = \mathrm{CE}(h^1_j(\boldsymbol{x}_i), \boldsymbol{e}_{i,j}) + \lambda\, \mathrm{KL}(h^1_j(\boldsymbol{x}_i) \| h^2_j(\boldsymbol{x}_i)) \quad (5)$$

where CE and KL denote the cross-entropy and Kullback–Leibler (KL) divergence respectively, $\boldsymbol{e}_{i,j} \in \{0,1\}^\ell$ is the one-hot encoded noisy label at pixel j in image $\boldsymbol{x}_i$, and $\lambda > 0$. CE measures the alignment between label noise and model predictions, and KL measures the inconsistency between the two teacher model predictions. Large CE at a pixel implies a high misalignment between the predicted class label and annotated label and hence high model confidence of the label to be noisy unless the current model overfits at that pixel. On the other hand, if one of the models starts to overfit the noisy annotations, then the KL measure will be high. If any of them is high, such pixel corresponds to a high score, and it is marked as noisy. Experimentally, we verify that the above choice of score works better than the other alternatives (shown in Fig. 8 (b)).

Once the noisiness score and the noise proportion have been calculated, we get the adaptive noise threshold as follows:

$$\beta^1 = \frac{\sigma \ell S^1_{avg1}}{\sum_{c\in\mathcal{C}} R^1_c}\ , \text{ where } S^1_{avg1} = n^{-1} \sum_{\substack{(\boldsymbol{x}_i,\cdot)\in\mathcal{D}_L,\\ j\in\mathcal{I}(\boldsymbol{x}_i)}} S^1(\boldsymbol{x}_i, j)\ , \quad (6)$$

where σ is a hyperparameter and n denotes the number of all the pixels in $\mathcal{D}_L$. Experimentally we found that the noise proportion R_c predicted by HOC is inaccurate for some of the classes and a single threshold β^1, computed from the average as in Eq. (6), works better than class-wise thresholds. A higher β means filtering out less noisy annotations. We utilize the adaptive threshold β^1 to determine the pixels to be removed and marked in the segmentation masks to reflect the pixels with noisy labels. Let us denote the set of pixels that are detected as label noise by the first LG as:

$$\mathcal{M}^1_k \coloneqq \left\{(\boldsymbol{x}_i, j) \mid S^1(\boldsymbol{x}_i, j) > \beta^1, (\boldsymbol{x}_i, \cdot) \in \mathcal{D}_L, j \in \mathcal{I}(\boldsymbol{x}_i)\right\}\ . \quad (7)$$

The updated datasets at epoch k are created by removing the pixels with noise labels from the labelled data and placing the images with noisy pixels in the unlabelled set, *i.e.*, the clean labelled set $\mathcal{D}^1_{L,k} \coloneqq \mathcal{D}_L \setminus \mathcal{M}^1_k$[2] and the unlabelled dataset $\mathcal{D}^1_{U,k} \coloneqq \mathcal{D}_U \bigcup \left\{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, \cdot) \in \mathcal{M}^1_k\right\}$. The student network of the second LG is then trained on these datasets according to Eq. (4). Analogously, the score for the second LG can be computed by switching h^1_j and h^2_j in Eq. (5), and the datasets $\mathcal{D}^2_{L,k}, \mathcal{D}^2_{U,k}$ are created in the above manner on which the student network of the first LG is trained. Our final algorithm is summarized in Algorithm 1.

Algorithm 1 Our training pipeline

Input: $\mathcal{D}_L$, $\mathcal{D}_U$, total epochs E, warm-up epochs N
Output: Optimized $\{\boldsymbol{\theta}_{s^1}, \boldsymbol{\theta}_{t^1}, \boldsymbol{\theta}_{s^2}, \boldsymbol{\theta}_{t^2}\}$

Initialize parameters $\{\boldsymbol{\theta}_{s^1}, \boldsymbol{\theta}_{t^1}, \boldsymbol{\theta}_{s^2}, \boldsymbol{\theta}_{t^2}\}$,
for $k \in \{0, 1, \ldots, E\}$ **do**
  if $k < N$ **then**
    $\mathcal{D}^1_{L,k}, \mathcal{D}^2_{L,k} \coloneqq \mathcal{D}_L$ and $\mathcal{D}^1_{U,k}, \mathcal{D}^2_{U,k} \coloneqq \emptyset$
  else
    $\{(\boldsymbol{x}_i, j, \tilde{y}_{i,j})\} \leftarrow$ Sample from $\mathcal{D}_L$
    for $l \in \{1, 2\}$ **do**
      $\left\{R^l_c\right\} \leftarrow \mathrm{HOC}\left(\left\{(g^l_j(\boldsymbol{x}_i), \tilde{y}_{i,j})\right\}\right)$
      Scores $\{S^l\} \leftarrow$ Using Eq. (5)
      $\beta^l \leftarrow$ Adaptive $\left(\{S^l\}\right)$ [Eq. (6)]
      $\mathcal{M}^l_k \leftarrow$ Using Eq. (7)
      $\mathcal{D}^l_{L,k} \leftarrow \mathcal{D}_L \setminus \mathcal{M}^l_k$
      $\mathcal{D}^l_{U,k} \leftarrow \mathcal{D}_U \bigcup \left\{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, \cdot) \in \mathcal{M}^l_k\right\}$
  repeat
    $\boldsymbol{\theta}_{s^1} \leftarrow$ Backprop. $\mathcal{L}(\mathcal{D}^2_{L,k}, \mathcal{D}^2_{U,k}, \boldsymbol{\theta}_{s^1})$ [Eq. (4)]
    $\boldsymbol{\theta}_{s^2} \leftarrow$ Backprop. $\mathcal{L}(\mathcal{D}^1_{L,k}, \mathcal{D}^1_{U,k}, \boldsymbol{\theta}_{s^2})$ [Eq. (4)]
  until M times
  Update $\boldsymbol{\theta}_{t^1}, \boldsymbol{\theta}_{t^2} \leftarrow$ Moving Average of $\boldsymbol{\theta}_{s^1}, \boldsymbol{\theta}_{s^2}$.

5. Experiments

Evaluation has been carried out following [19, 20] to allow direct comparison. As in [20], the labelled data is augmented with synthetic label noise. Performance is measured with, and without an unlabelled dataset to illuminate performance differences between the weakly and semi supervised settings.

5.1. Implementation Details

We utilize R50 [9] and PVTv2-B2 [33] as the backbones in the two LGs to encourage diversity. Both the teacher and student models are based on DeeplabV3+ [4]. For inference, we use the average of the predictions from the two teacher networks as in [20]. The λ in Eq. (5) is set to 0.1 to balance the contribution from the CE and KL terms in our loss. We randomly sample 100 pixels per class in each image to get the representative dataset $\bar{\mathcal{D}}$. The hyperparameter σ in Eq. (6) is set to be 10 in our experiments. Other

[2] With a slight abuse of notation, this operation indicates marking all pixels in $\mathcal{M}^1_k$ with "ignore-pixel" label in $\mathcal{D}_L$.

Table 1: Comparison against SotA approaches on Pascal VOC 2012 in a semi-supervised semantic segmentation setting. All baselines are based on the DeeplabV3+ architecture. The * indicates results reported by [6]. When we replace 50% of the ground truth of images with 'PL', the proportion of noisy pixels is 9%. We keep the same pixel level noise proportion while introducing noise using 'RDE' and 'SCP' for parity.

Methods	Year	Noisy pixel	Noise type	Labelled Data Ratio			
				1/16 (662)	1/8(1323)	1/4 (2646)	1/2 (5291)
MT* [30]	2017	0%	None	66.70	70.78	73.22	75.41
French [8]	2019	0%	None	68.90	70.70	72.46	74.49
CCT* [26]	2020	0%	None	65.22	70.87	73.43	74.75
GCT* [14]	2020	0%	None	64.05	70.47	73.45	75.20
ECS* [23]	2020	0%	None	-	67.38	70.70	72.89
CPS* [6]	2021	0%	None	71.98	73.67	74.90	76.15
CAC* [16]	2021	0%	None	70.10	72.40	74.00	-
PS-MT [20]	2022	0%	None	72.83	75.70	76.43	77.88
Ours	-	0%	None	77.75 (+4.92)	79.31 (+3.61)	79.14 (+2.71)	79.54 (+1.66)
PS-MT [20]	-	9%	PL	61.90	65.14	65.53	66.78
Ours	-	9%	PL	71.60 (+9.70)	73.95 (+8.81)	74.53 (+9.00)	74.44 (+7.66)
PS-MT [20]	-	9%	RDE	58.87	62.18	63.81	63.46
Ours	-	9%	RDE	66.82 (+7.95)	69.95 (+7.77)	73.82 (+10.01)	74.47 (+11.01)
PS-MT [20]	-	9%	SCP	52.14	57.19	56.81	53.69
Ours	-	9%	SCP	67.80 (+15.66)	70.83 (+13.64)	69.38 (+12.57)	75.95 (+22.26)

training and inference settings are the same as those of the baseline method [20]. In all of our experiments, the last epoch checkpoint model is used unless otherwise specified. Further details of the implementation can be found in the supplementary materials.

5.2. Semi-supervised Semantic Segmentation

Pascal VOC 2012 [7] is employed for evaluating the performance of our model in a semi-supervised semantic segmentation setting. This dataset contains more than 13k images with 21 different classes. To introduce realistic noise in $\mathcal{D}_L$, the following three approaches are considered:

- **Pseudo Labels (PL).** Following [19], a portion of clean labels are replaced by pseudo-labels from a weakly-supervised segmentation method [34].
- **Random Dilation and Erosion (RDE).** Similar to [19, 21], we apply random degrees of dilation and erosion to the ground-truth masks.
- **Similar Class Perturbation (SCP).** We introduce another type of noise by randomly replacing ground-truth label of a segmentation mask with a different class that resembles the texture of original clean class.[3] This label perturbation simulates a more natural scenario pertaining to annotators confusing semantics.

[3]Note that, SCP does not change the shape of the clean segmentation masks, but only labels may be switched.

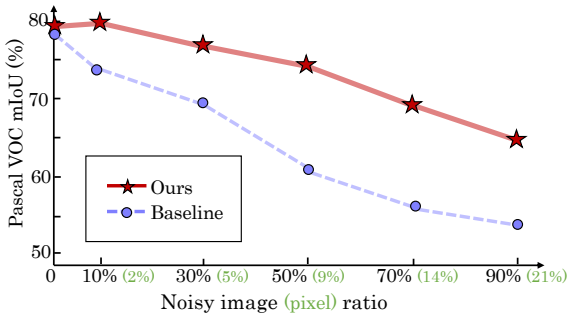


Figure 5: A performance comparison under different noise proportions achieved by replacing (from 0% to 90%) clean segmentation masks with the PL from [34] at image-level. The green font shows the corresponding noisy pixel proportions. All experiments are conducted with $1/8$ labelled data.

Details about the implementation and visualization of noise generation can be found in the supplementary material.

In the context of semi-supervised semantic segmentation, we have compared our approach with the following studies: MT [30], French [8], CCT [26], GCT [14], ECS [23], CPS [6], CAC [16], and PS-MT [20]. Following the previous studies [19, 20], we report the mean Intersection-over-Union (mIoU) on the validation set. Table 1 shows that our approach ('Ours') outperforms all baselines trained with clean labelled data. In the presence of noise, we compare the best performing semi-supervised segmentation method

Table 2: The mIoU of the weakly-supervised semantic segmentation baselines and ours on the test set of SegTHOR. Best Val and Last Epoch correspond to the checkpoint that performs best on the validation set and the last checkpoint.

Model	Best Val	Last Epoch
Base-ADELE [19]	62.6	59.1
ADELE [19]	71.1	70.8
Ours	**73.2**	**73.5**

with no noise, *i.e.*, PS-MT [20], against our method with varying noise types (see Table 1) and noise proportions (see Fig. 5). The results demonstrate significant improvements in mIoU in all cases, *e.g.*, $+22.26\%$ with half of the labelled data on the more practical SCP noise. Overall, the superiority of our method is consistent across all labelled data to unlabelled data ratios, regardless of the presence of noise or the type of applied noise.

5.3. Weakly-supervised Semantic Segmentation

We evaluate our method in a weakly-supervised semantic segmentation setting on the following datasets: (i) SegTHOR [17], a medical dataset, and (ii) pseudo labelled Pascal VOC 2012.

Medical image segmentation on SegTHOR. Following [19], we evaluate our approach on the SegTHOR dataset, which consists of 3D CT scans. Each pixel of these images is assigned to one of these five classes: 'esophagus', 'heart', 'trachea', 'aorta', or 'background'. The slices from the images are randomly split into a training set of 3638 slices, a validation set of 570 slices, and a test set of 580 slices [19, 17]. To introduce noise, we apply 'RDE' to the ground-truth segmentation labels, similar to [19]. Note that all annotations in the training set are corrupted by this noise, whereas the validation and test sets remain uncorrupted.

We compare our approach with ADELE [19], which corrects noisy annotations with model predictions in the early training stage, and its variant, Base-ADELE [19] , which is based on a UNet trained with multi-scale inputs. Detailed implementations of these approaches are in the supplementary material. Following the setting in ADELE [19], we evaluate two checkpoints: specifically, the one with the highest mIoU on the validation set and the last checkpoint. In Table 2, our approach outperforms both baselines.

Weakly-supervised segmentation on Pascal VOC 2012. We demonstrate the effectiveness of our approach in comparison to the baselines in the weakly-supervised setting on the PASCAL VOC 2012 [7]. Similar to ADELE [19], we use only noisy pseudo labels generated by [34] to train the models and evaluate on the original test set. Table 3 shows that our approach outperforms recent baselines, such as ADELE combined with SEAM [19]. In addition, PS-MT increases the mIoU marginally with PVTv2 compared to R50 with 66.7 $vs.$ 64.0. However, after using our framework, the mIoU is further improved from 66.7 to 71.7, clearly demonstrating the effectiveness of our method.

Table 3: Comparison with state-of-the-art weakly supervised semantic segmentation methods on the Pascal VOC 2012 dataset using mIoU (%) of the validation set.

Model	Year	mIoU (%)
AffinityNet [1]	2018	61.7
SSDD [27]	2019	64.9
SEAM [34]	2020	64.5
CONTA [37]	2020	66.1
SPML [13]	2021	69.5
ADELE [19] + SEAM	2022	69.3
PS-MT [20] (R50)	2022	64.0
PS-MT [20] (PVTv2-B2)	2022	66.7
Ours	-	**71.7**

5.4. Ablation Study

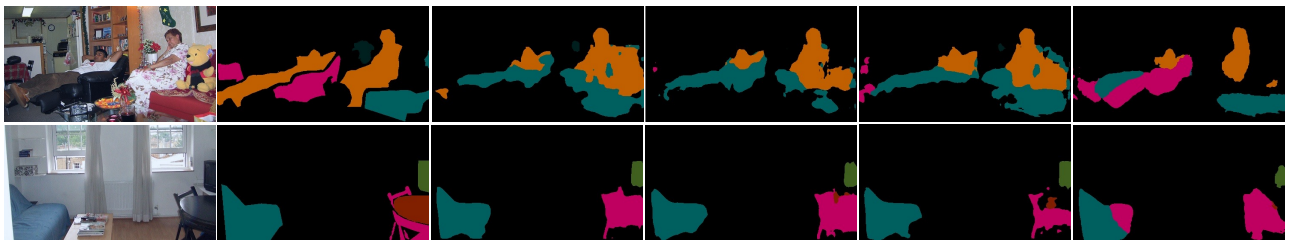
All experiments in this section are conducted with $1/8$ labelled data ratio and 9% PL noisy annotations on Pascal VOC. We verify the effectiveness of the two main components in our approach, the diverse LGs and filter module.

Diversity in the LGs with different architectures. In contrast to previous works [20, 18], we found that using different network architectures in the two LGs provides more complementary information. In Fig. 3, we show the qualitative results of noise detection when we use the same and different architectures in the two LGs. The two LGs provide more diverse noise detection results (lower overlap between predicted noisy labels) when we employ different architectures. In Fig. 6, the predictions from the two LGs with two different architectures ((e)&(f)) are more complementary than those from the two LGs with the same architecture ((c)&(d)). Fig. 7 shows the mIoU of our pipeline when we use different network combinations. The combination of R50 and PVTv2-B2 [33] achieves the best performance.

Noise detection with different LGs. To avoid confirmation bias, we utilize the teacher network from one LG to detect noise for the other LG. If we utilize the teacher network from the same LG for noise detection (*i.e.*, replacing $\boldsymbol{\theta}_{s^1}$ with $\boldsymbol{\theta}_{s^2}$ in Eq. (4)), the mIoU drops from 74.0% to 72.4% on Pascal VOC, demonstrating the importance of mutual noise detection with a different LG in our pipeline.

Significance of the filter module. In Table 4, in addition to the baseline PS-MT and ours, we evaluate the performance of the following models: 1) 'PS-MT + FM' by adding the filter module to PS-MT (using R50)[4]. 2) 'Ours − FM' by

[4]Note, PS-MT can be regarded as single LG (Fig. 2 (a)) and the mean

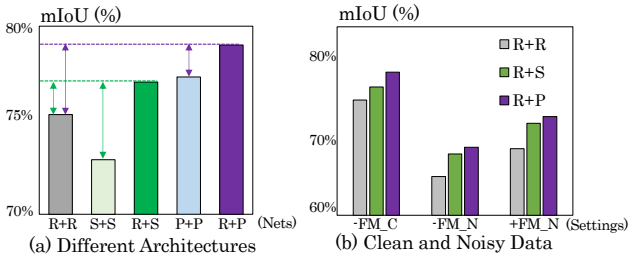


(a) Input Image (b) Ground Truth (c) R50^{LG1} Prediction (d) R50^{LG2} Prediction (e) R50^{LG1} Prediction (f) PVT^{LG2} Prediction

Figure 6: Visualizations from the two LGs when we use the same architecture ((c) and (d)) and different architectures ((e) and (f)). The $1st$ and $2nd$ columns show the inputs and the ground truth. R50 and PVT denote R50 and PVTv2-B2 [33].

Table 4: Ablation study of FM and LG on Pascal VOC.

Method	2 LGs	Filter Module (FM)	mIoU (%)
PS-MT [20]			65.1
PS-MT + FM		✓	67.6 (+2.5)
Ours − FM	✓		69.9 (+4.8)
Ours	✓	✓	74.0 (+8.9)

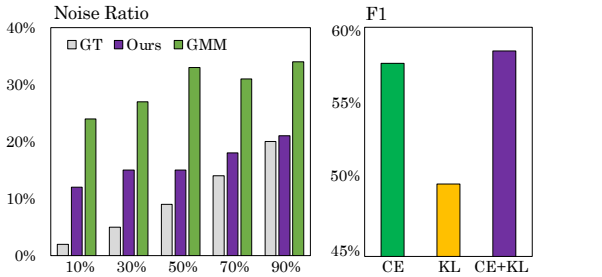


(a) Different Architectures (b) Clean and Noisy Data

Figure 7: (a) shows the mIoU score of different network combinations on clean data. (b) shows the mIoU score of different combinations on different settings. R, S and P denote R50 [9], Segformer [35] and PVTv2-B2 [33] respectively. -FM_C denotes training our model (without the filter module) with clean data. -FM_N denotes training model with noisy data without using filter module. +FM_N is our complete model trained with noisy data.

removing the filter module from our framework. It can be seen that the filter module helps to improve the final performance for both models. The filter module contributes more significantly (+4.1) to our framework due to the diverse learning groups (*i.e.*, using 2 LGs with different architectures). This superiority of our framework is further evidenced by the better performance of 'Ours − FM' compared to PS-MT variants. In Fig. 7 (b), our Filter Module helps to improve the performance under noisy annotations ('+FM_N' $vs.$ '-FM_N') with different architectures.

Comparison of the filter module with GMM. Previous work [18] utilizes Gaussian Mixture Model (GMM) to detect noise for image classification. We compare our filter module with GMM (including two components) on the representative dataset. Fig. 8 (a) shows the detected noise proportions. We can see that the detected noise proportion of our filter module is closer to the groundtruth than GMM.



(a) Detected Noise Ratio (b) F1 Score of Noise Detection

Figure 8: (a) shows detected noise proportion of GMM [18] and ours, where x-axis shows the noisy image ratio and y-axis shows the detected noisy pixel ratio. The detected noise proportion of ours is closer to the GT than GMM. (b) shows the F1 score of noise detection with different criterion. CE is cross entropy loss; KL is KL divergence between the predictions from the two LGs. Our CE+KL performs the best.

Criterion to compute score. We utilize cross entropy loss and KL divergence (CE+KL) to compute score in Eq. (5). Here, we compare this criterion with others, including only cross entropy loss (CE), and only KL divergence (KL). Fig. 8 (b) shows that ours achieves the best F1 score.

Memory and computation costs. PVTv2-B2 has similar GFLOPs (45.8) with R50 (45.6). Ours only have one more student network than PS-MT, **which is not used during inference**. Thus, our model needs similar memory and computation costs compared with PS-MT during inference.

6. Conclusion

In this work, we propose a learning framework for semi-supervised semantic segmentation involving two diverse learning groups that can handle various labelling errors. Our framework employs a carefully designed filter module to divide the labelled dataset into pixels with and without label noise in a scalable manner. The potentially misla-

teacher output is used for filter module.

belled pixels filtered by the first learning group are utilised for training the second learning group and vice versa. We show that our proposed framework provides state-of-the-art results for semi-supervised learning, by addressing confirmation bias while training, that can handle various amounts of annotation noise to accommodate practical inaccuracies in annotations.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

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

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

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

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

[8] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, high-dimensional perturbations. 2019.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

[11] Mostafa S Ibrahim, Arash Vahdat, Mani Ranjbar, and William G Macready. Semi-supervised semantic image segmentation with self-correcting networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12715–12725, 2020.

[12] Joseph Jacob, Olga Ciccarelli, Frederik Barkhof, and Daniel C Alexander. Disentangling human error from the ground truth in segmentation of medical images.

[13] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. *arXiv preprint arXiv:2105.00957*, 2021.

[14] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *European conference on computer vision*, pages 429–445. Springer, 2020.

[15] Donghyeon Kwon and Suha Kwak. Semi-supervised semantic segmentation with error localization network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9957–9967, 2022.

[16] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1205–1214, 2021.

[17] Zoé Lambert, Caroline Petitjean, Bernard Dubray, and Su Kuan. Segthor: segmentation of thoracic organs at risk in ct images. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2020.

[18] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[19] Sheng Liu, Kangning Liu, Weicheng Zhu, Yiqiu Shen, and Carlos Fernandez-Granda. Adaptive early-learning correction for segmentation from noisy annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2606–2616, 2022.

[20] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and Gustavo Carneiro. Perturbed and strict mean teachers for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4258–4267, 2022.

[21] Yaoru Luo, Guole Liu, Yuanhao Guo, and Ge Yang. Elucidating meta-structures of noisy labels in semantic segmentation by deep neural networks. *arXiv preprint arXiv:2205.00160*, 2022.

[22] Yaoru Luo, Guole Liu, Wenjing Li, Yuanhao Guo, and Ge Yang. Deep neural networks learn meta-structures to segment fluorescence microscopy images. *arXiv preprint arXiv:2103.11594*, 2021.

[23] Robert Mendel, Luis Antonio de Souza, David Rauber, Joao Paulo Papa, and Christoph Palm. Semi-supervised segmentation based on error-correcting supervision. In *European Conference on Computer Vision*, pages 141–157. Springer, 2020.

[24] Shaobo Min, Xuejin Chen, Zheng-Jun Zha, Feng Wu, and Yongdong Zhang. A two-stream mutual attention network for semi-supervised biomedical segmentation with noisy labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4578–4585, 2019.

[25] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1369–1379, 2019.

[26] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[27] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5208–5217, 2019.

[28] Yucheng Shu, Xiao Wu, and Weisheng Li. Lvc-net: Medical image segmentation with noisy label based on local visual cues. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 558–566. Springer, 2019.

[29] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE international conference on computer vision*, pages 5688–5696, 2017.

[30] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[31] Luis Garcia Ugarriza, Eli Saber, Sreenath Rao Vantaram, Vincent Amuso, Mark Shaw, and Ranjit Bhaskar. Automatic image segmentation by dynamic region growth and multiresolution merging. *IEEE transactions on image processing*, 18(10):2275–2288, 2009.

[32] Guotai Wang, Xinglong Liu, Chaoping Li, Zhiyong Xu, Jiugen Ruan, Haifeng Zhu, Tao Meng, Kang Li, Ning Huang, and Shaoting Zhang. A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images. *IEEE Transactions on Medical Imaging*, 39(8):2653–2663, 2020.

[33] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

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

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[36] Rumeng Yi, Yaping Huang, Qingji Guan, Mengyang Pu, and Runsheng Zhang. Learning from pixel-level label noise: A new perspective for semi-supervised semantic segmentation. *IEEE Transactions on Image Processing*, 31:623–635, 2021.

[37] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 33:655–666, 2020.

[38] Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning*, pages 27412–27427. PMLR, 2022.

[39] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *International Conference on Machine Learning*, pages 12912–12923. PMLR, 2021.

[40] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020.