Appendix for "Adaptive Early-Learning Correction for Segmentation from Noisy Annotations"

The appendix is organized as follows.

- In Appendix A, we include additional figures illustrating the early-learning and memorization phenomena in PASCAL VOC 2012 and SegTHOR. We also include additional results for these two datasets.
- In Appendix B, we describe the implementation details of our proposed method ADELE, including a description of the hyperparameters, experiment settings, and other technical details.
- In Appendix C, we report ablation studies on the influence of different components and hyperparameters of ADELE.

License of the assets

Licence for the codes

We reproduce the code for AffinityNet [3], SEAM [52], ICD [14], all of which are under MIT License according to https://opensource.org/licenses/MIT.

Licence for the dataset

As the PASCAL VOC [13] includes images obtained from the "flickr" website, we respect the corresponding terms of use for "flickr" according to https://www.flickr.com/help/terms.

For the SegThor [27] dataset, we follow the data use agreement according to https://pagesperso.litislab.fr/cpetitjean/wp-content/uploads/sites/19/2021/03/DataUseAgreement_CHBDatabase.pdf.

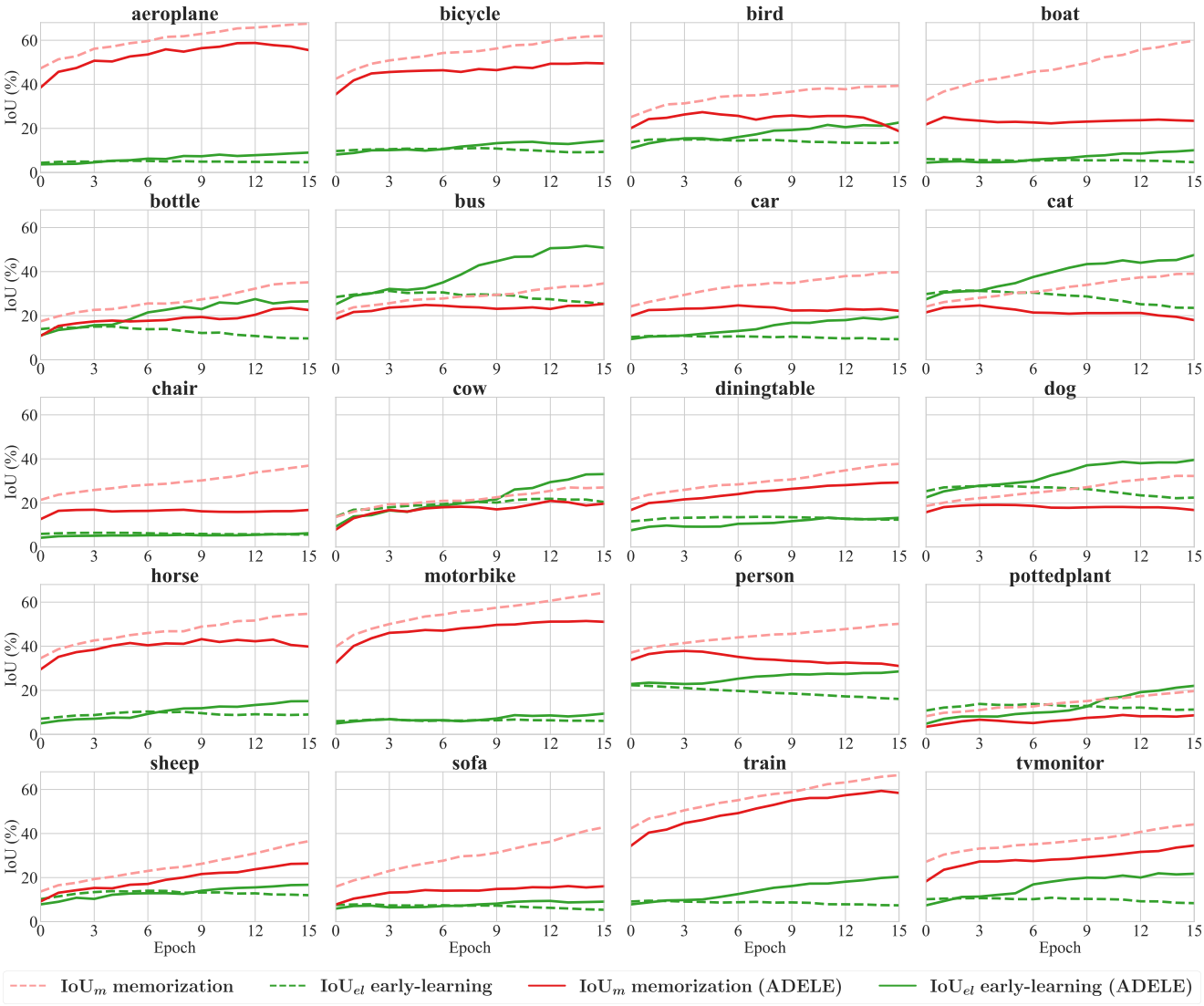


Figure 10. We visualize the effect of early learning (IoU$_{el}$, green curves) and memorization (IoU$_m$, red curves) on segmentation models trained with (solid lines) and without (dashed lines) ADELE for each category of the WSSS dataset VOC 2012 [13] The WSSS model is a standard DeepLab-v1 network trained with annotations obtained from SEAM [52]. IoU$_{el}$ is the IOU between the model output and the ground truth computed over the incorrectly-labeled pixels. IoU$_m$ is the IOU between the model output and the incorrect annotations. For all classes, IoU$_m$ increases substantially as training proceeds because the model gradually memorizes the incorrect annotations. This again occurs at different speeds for different categories. In contrast, IoU$_{el}$ first increases during an early-learning stage where the model learns to correctly segment the incorrectly-labeled pixels, but eventually decreases as memorization occurs (the phenomenon is more evident when we zoom in, as shown in Figure 11 in the Appendix). Like memorization, early-learning also happens at varying speeds for the different semantic categories.

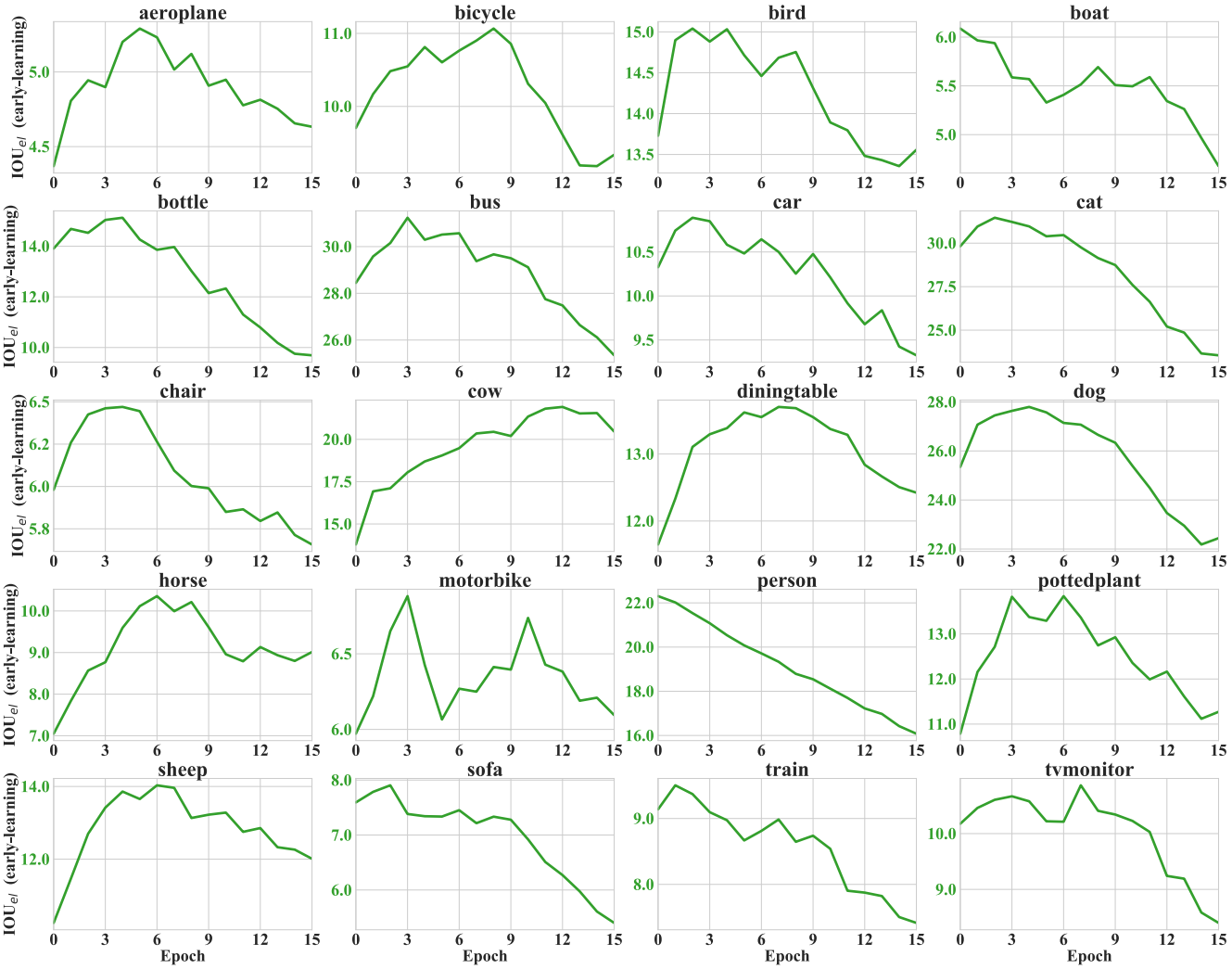


Figure 11. Zoomed-in illustration of early learning for the different semantic categories on PASCAL VOC 2012. IoU$_{el}$ first increases during an early-learning stage where the model learns to correctly segment the incorrectly-labeled pixels, but eventually decreases as memorization occurs. Early learning happens at varying speeds for the different semantic categories. The experimental setting is the same as Figure 3.

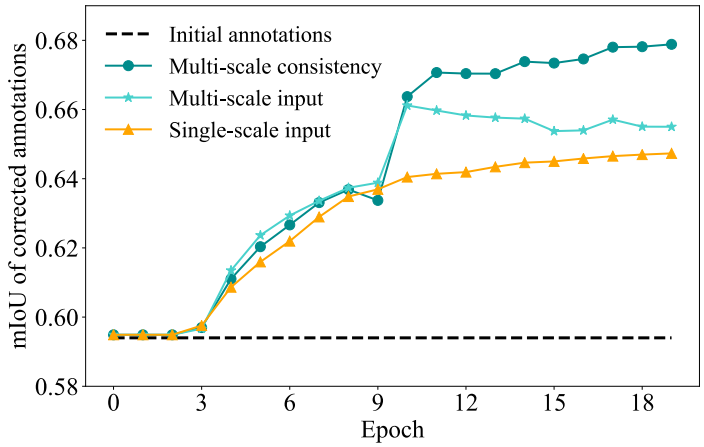


Figure 12. Multi-scaleconsistency regularization leads to more accurate corrected annotations (result on Pascal VOC).

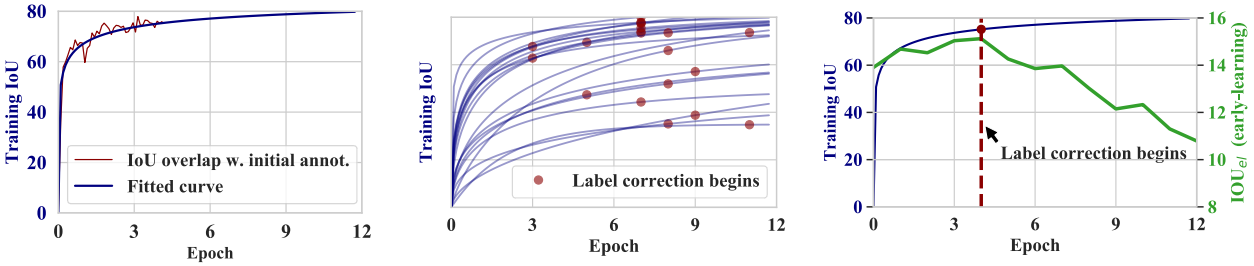


Figure 13. Illustration of the proposed curve fitting method to decide when to begin label correction in ADELE (Results on Pascal VOC). On the left, we plot the IoU between the model predictions and the initial noisy annotations for the same model used in Figures 11 and 4 and the corresponding fit with the parametric model in Equation 1. The label correction beginning iteration is based on the relative slope change of the fitted curve. The center image shows the label correction times for different semantic categories, showing that they are quite different. On the right graph, the green line shows the IoU$_{el}$ for a given category. The IoU$_{el}$ equals the IoU between the model output and the ground truth computed over the incorrectly-labeled pixels, and therefore quantifies early-learning. The label correction begins close to the end of the early-learning phase, as desired.

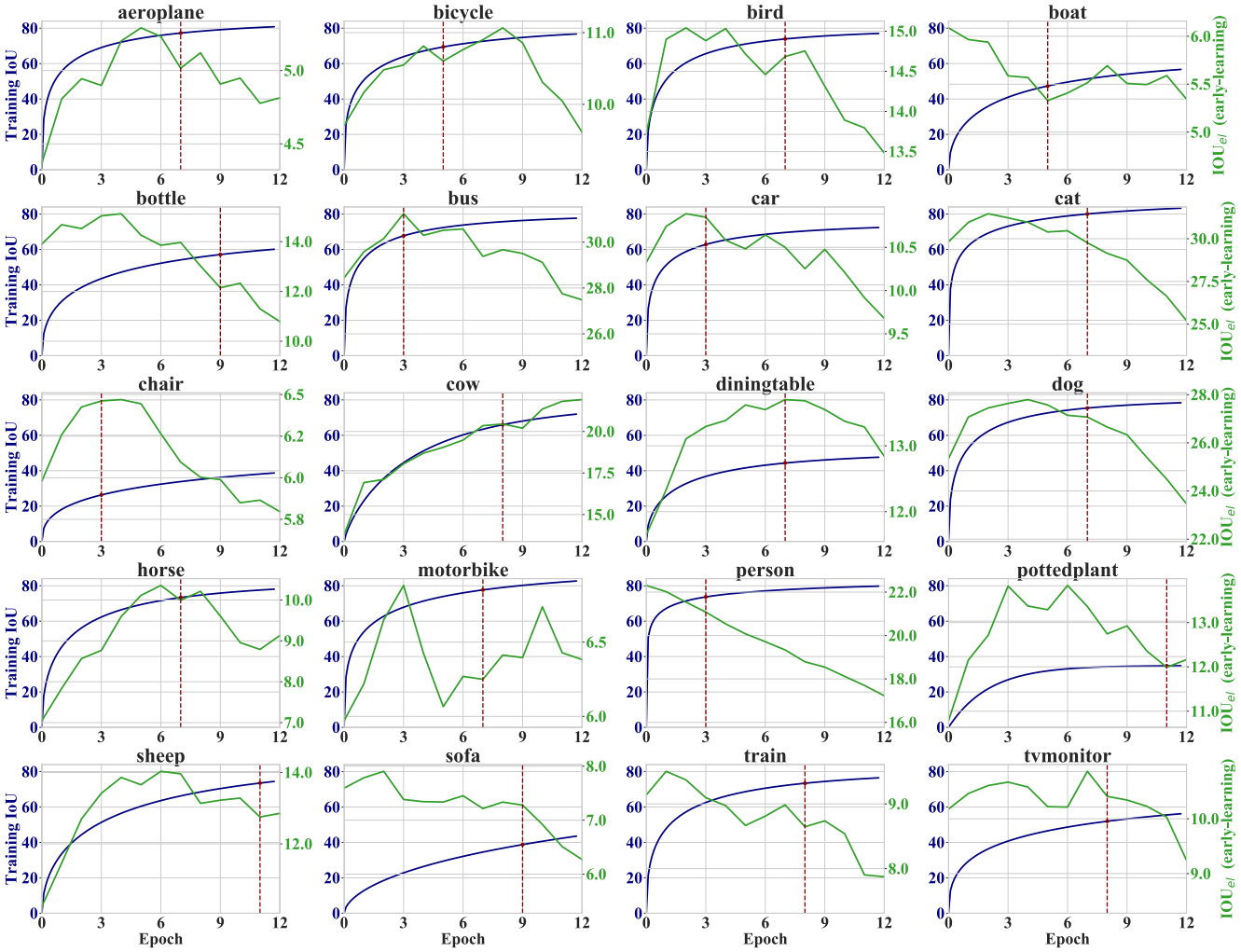


Figure 14. Illustration of the proposed curve fitting method to decide when to begin label correction in ADELE. The blue line shows the parametric model in Equation 1 fit to the IoU between the model predictions and the initial noisy annotations for the same WSSS model used in Figures 3 and 4. The green line shows the IoU$_{el}$ for a given category. The IoU$_{el}$ equals the IOU between the model output and the ground truth computed over the incorrectly-labeled pixels, and therefore quantifies early-learning. The label correction begins close to the end of the early-learning phase for most of the categories with a few exceptions (boat and motorbike).

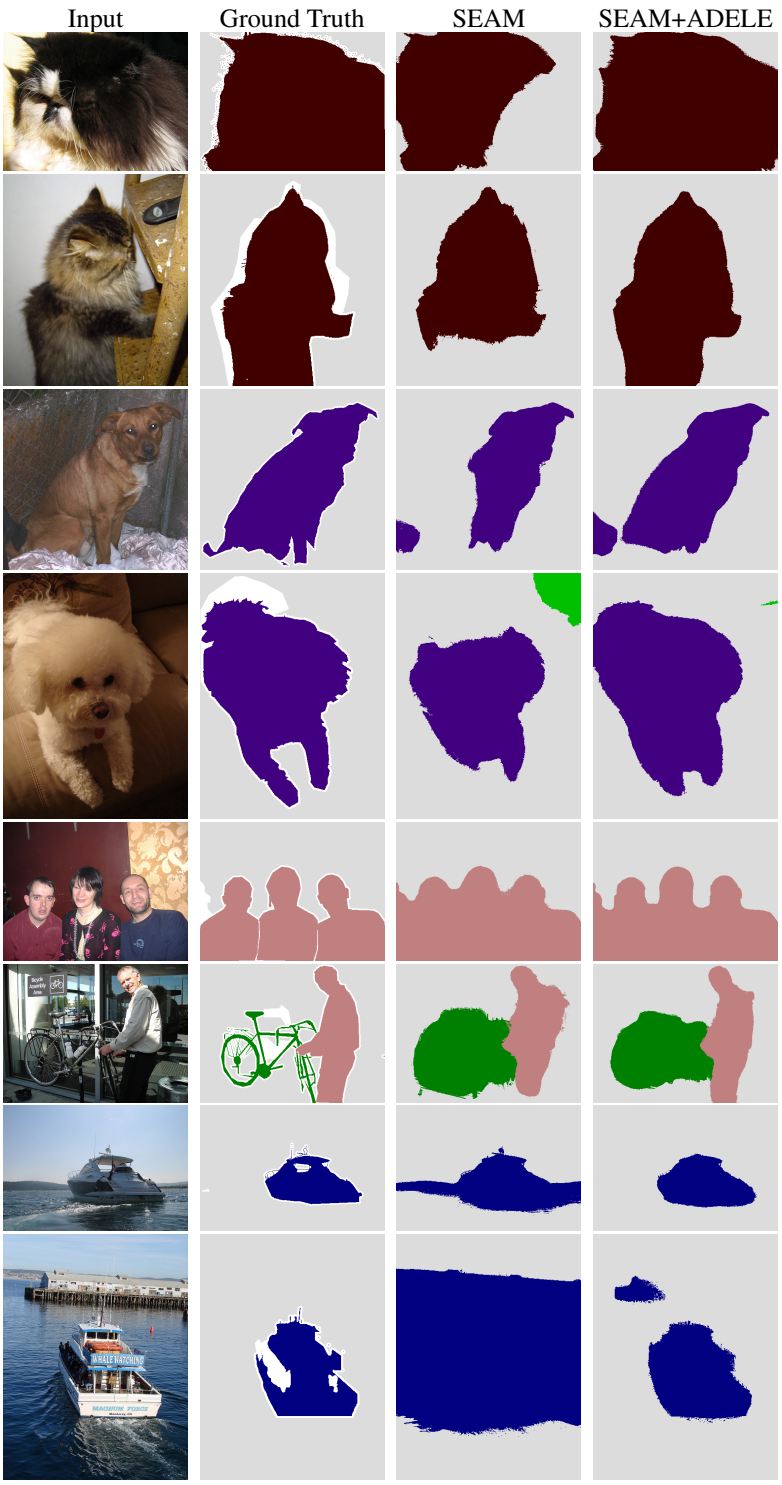


Figure 15. Additional segmentation results of SEAM and SEAM+ADELE for several examples on the validation set of PASCAL VOC 2012. We set the background color to gray for ease of visualization. Supplementary for Figure 1.

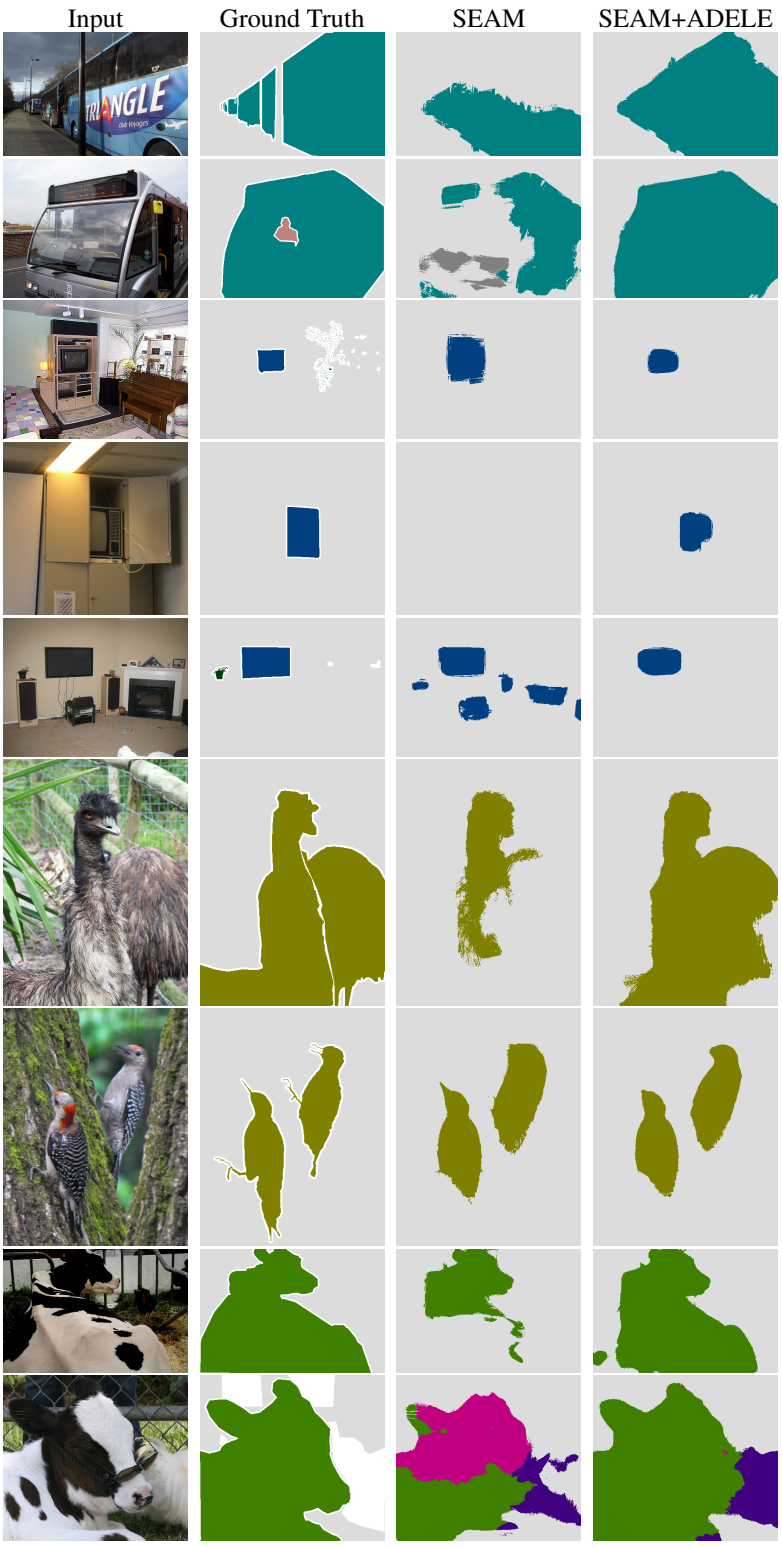


Figure 16. Additional segmentation results of SEAM and SEAM+ADELE for several examples on the validation set of PASCAL VOC 2012. We set the background color to gray for ease of visualization. Supplementary for Figure 1.

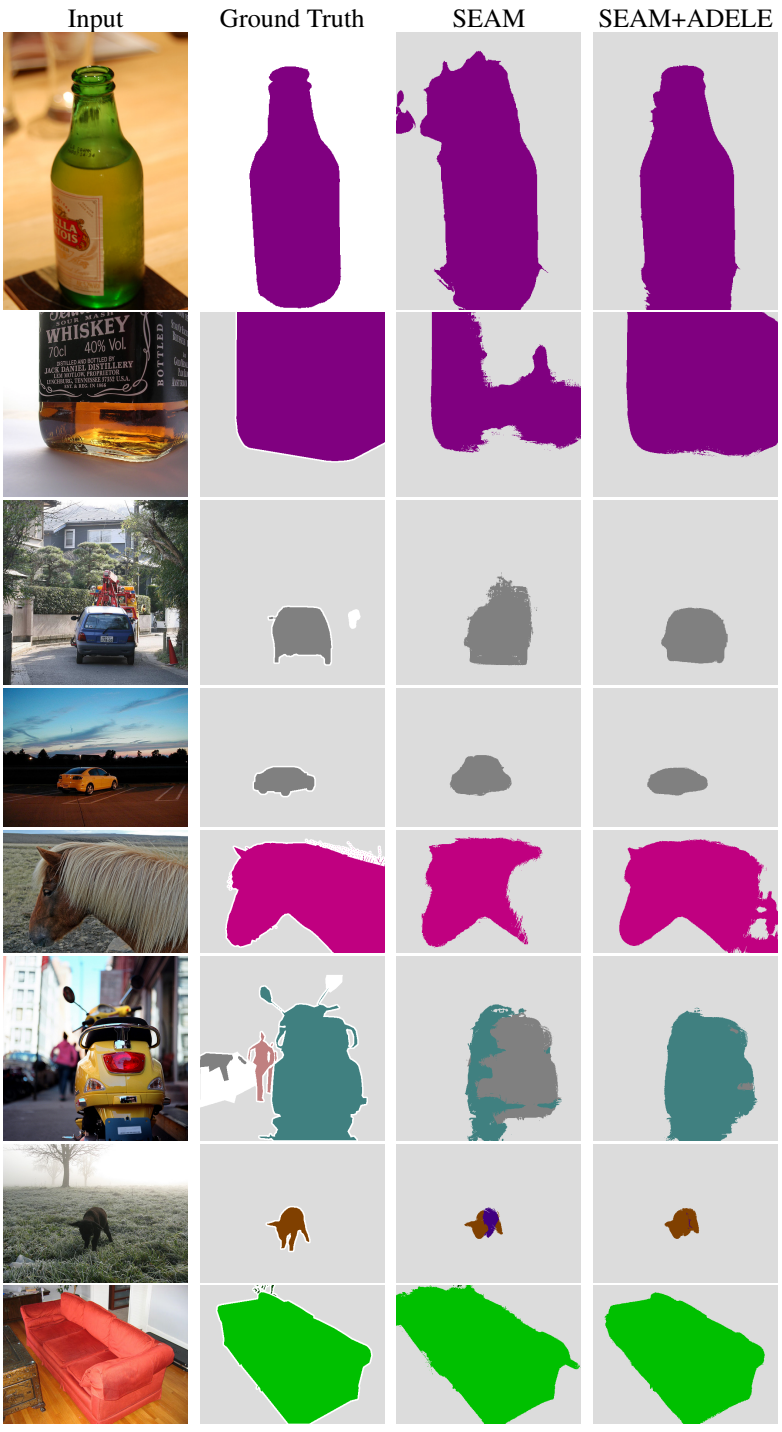


Figure 17. Additional segmentation results of SEAM and SEAM+ADELE for several examples on the validation set of PASCAL VOC 2012. We set the background color to gray for ease of visualization. Supplementary materials for Figure 1.

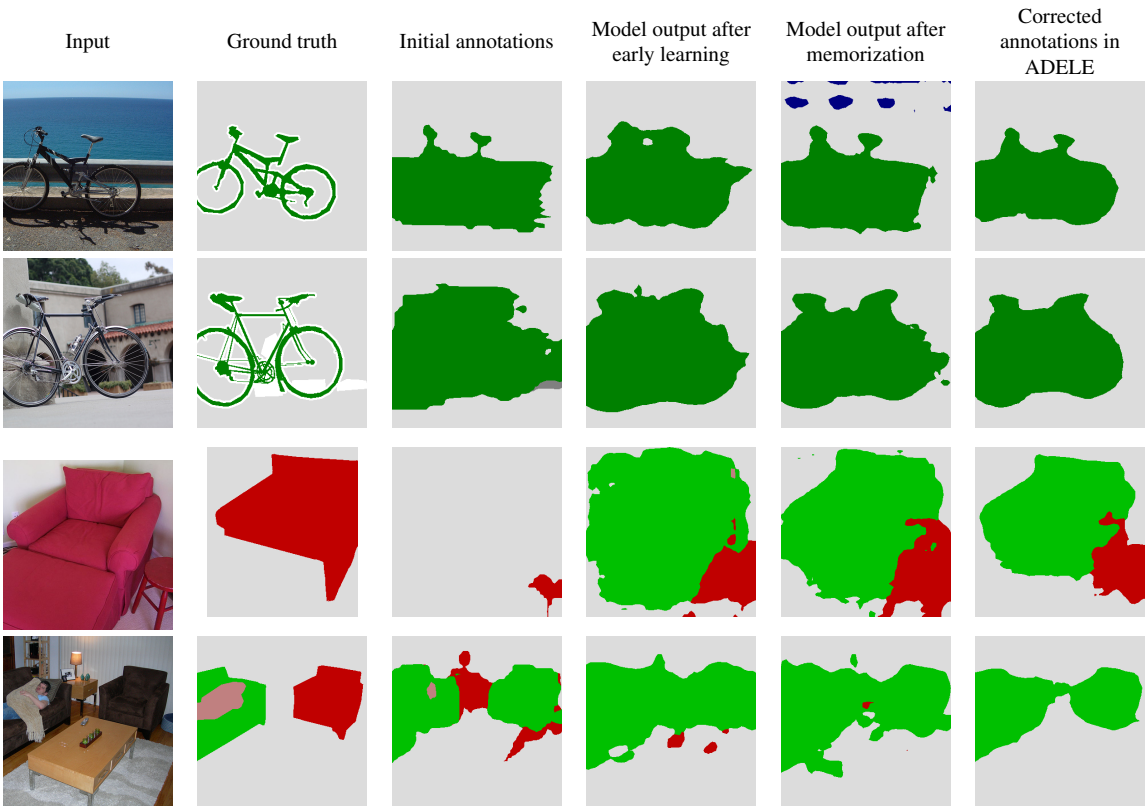


Figure 18. Illustration of a possible limitation of the proposed label-correction approach on PASCAL VOC 2012. The initial annotations coarsely segment the bike and misclassify the chair as a sofa consistently in several training examples. This highly structured annotation noise could potentially prevent early learning from happening, and therefore from being exploited for label correction. We set the background color to light gray for ease of visualization.

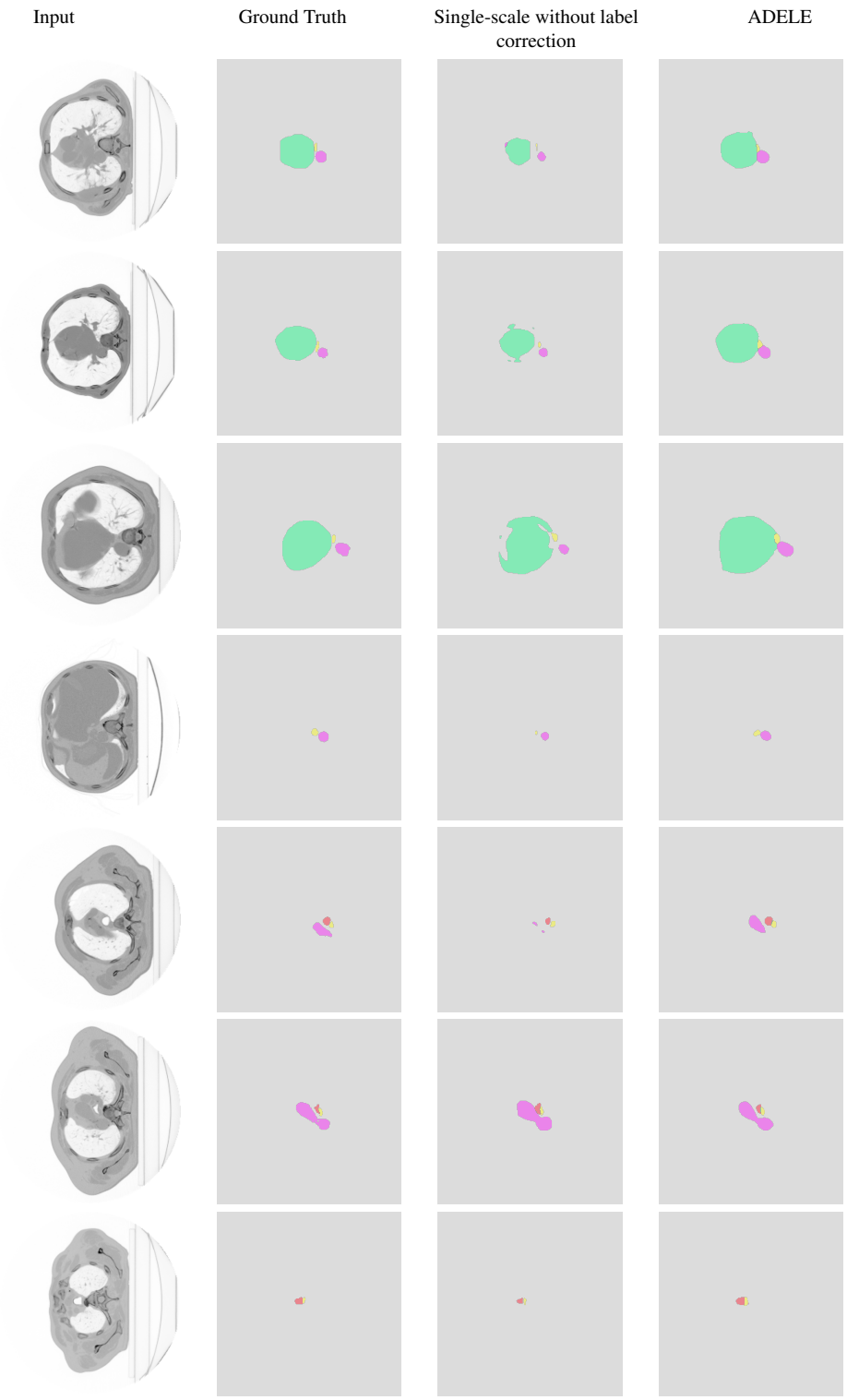


Figure 19. Visualization of the segmentation results of baseline and baseline+ADELE for several examples on the *validation* set of SegTHOR.

A. Additional Experimental Results

In this section, we include additional examples (A.1) and results (A.2).

A.1. Additional Examples

WSSS dataset (PASCAL VOC 2012)

- *Early-learning.* (Supplementary material for Figure 3 and 5 in Section 2) Figure 10 demonstrates that early-learning and memorization on a segmentation CNN trained with noisy annotations generated by a classification model for the standard WSSS dataset (PASCAL VOC2012). In Figure 11 we show the zoomed-in version of the early-learning IoU_{el} curves. Early-learning happens for most of the classes at different speeds. Figure 13 show the curve fitting illustration for PASCAL VOC. Figure 14 shows that the label correction begins close to the end of the early-learning phase for most of the semantic categories for PASCAL VOC.

- *Multiscale consistency.* Figure 12 shows that multi-scaleconsistency regularization leads to more accurate corrected annotations for PASCAL VOC.

- *Superior performance.* (Supplementary material for Figure 1 in Section 5) We provide more visualization examples on the validation set of PASCAL VOC in Figure 15, 16 and 17 comparing ADELE to the baseline method SEAM. ADELE reduces false positives for some examples (*e.g.* boat, person, sofa, bottle, tv *etc.*) and produces a more complete segmentation of other examples (*e.g.* cat, dog, bird, bus, bottle, tv, horse *etc.*).

- *Highly structured annotation noise.* (Supplementary material for Figure 1) Figure 18 shows training examples with highly structured noise, which may prevent early learning from happening, and therefore from being used for label correction.

More visual examples for SegTHOR

- *Superior performance*. (Supplementary material for Section 4). We show some visual examples that indicate ADELE improves segmentation performance on the validation set of SegTHOR dataset in Figure 19.

A.2. Additional Results

ADELE outperforms methods using external information. (Supplementary material for Section 5) ADELE using initial labels generated from SEAM [52] and ICD [14] achieves state-of-the-art performance without using external saliency models. This performance is on par with or even slightly better than methods that rely on external saliency models [20, 45, 59, 64]. To show that our method is complementary with other more advanced WSSS methods, we have conducted an experiment with a recent WSSS method NSROM [59]. ADELE+NSROM achieves mIoU of 71.6 and 72.0 on the validation and test set respectively, which is the SoTA for WSSS with ResNet segmentation backbone.

Method	Supervision	Backbone	val	test
DSRG [18]	I.+ S.	ResNet-101	61.4	63.2
SPLIT MERGE [64]	I.+S.	ResNet-50	66.6	66.7
MCIS [45]	I.+S.	ResNet-101	66.2	66.9
NSROM [59]	I.+S.	ResNet-101	68.2	68.5
NSROM* [59]	I.+S.	ResNet-101	**70.4**	**70.2**
SEAM + Ours	I.	ResNet-38	69.3	68.8
ICD + Ours	I.	ResNet-38	68.6	68.9
NSROM + Ours	I.+S.	ResNet-38	**71.6**	**72.0**

Table 4. Comparison with previous works that use external saliency models [19] on the PASCAL VOC 2012 dataset [13]. I. stands for image-level labels, S. stands for external saliency models. * denotes model is pre-trained on MS-COCO

ADELE improves performance for most categories. (Supplementary material for Figure 8 in Section 5) Table 5 shows that ADELE performs substantially better than the baseline method SEAM on most of the semantic categories, as well as on average.

ADELE improves performance across different noise levels. We provided the full result of Figure 7 in Table 6 for SegTHOR dataset, which shows that ADELE can improve the model performance across different noise level.

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIoU
SEAM	88.8	68.5	**33.3**	85.7	40.4	67.3	78.9	76.3	81.9	**29.1**	75.5	**48.1**	79.9	73.8	71.4	75.2	**48.9**	79.8	40.9	**58.2**	53.0	64.5
SEAM + Ours	**91.1**	**77.6**	33.0	**88.9**	**67.1**	**71.7**	**88.8**	**82.5**	**89.0**	26.6	**83.8**	44.6	**84.4**	**77.8**	**74.8**	**78.5**	43.8	**84.8**	**44.6**	56.1	**65.3**	**69.3**

Table 5. Category-wise comparison of the IoU (%) of SEAM [52] and SEAM combined with the proposed method ADELE on the validation set of PASCAL VOC 2012.

Iteration of erosion/dilation	Annotation mIoU	Multiscale input augmentation (baseline)	Multiscale label correction	Multiscale consistency regularization	ADELE
0	1.00	0.745	0.743	0.762	**0.766**
1	0.91	0.743	0.726	**0.759**	0.757
2	0.73	0.702	0.710	0.714	**0.734**
3	0.61	0.646	0.658	0.647	**0.711**
4	0.52	0.556	0.651	0.606	**0.666**
6	0.39	0.446	0.556	0.514	**0.564**
8	0.28	0.416	0.407	0.423	**0.481**

Table 6. The mIoU (%) comparison of the baseline and ADELE on the test set of SegTHOR [27]. We report the test mIoU of the model that achieves best mIoU on the validation set. It can be seen that ADELE stably improve the model performance across different noise levels.

B. Implementation details

B.1. Hyperparameters

Table 7 provides a complete list of hyperparameter values for our experiments. We use almost identical hyperparameters of ADELE on PASCAL VOC 2012 and SegTHOR. λ (*consistency strength*) controls the strength of the consistency regularization added to the cross entropy loss. ρ (*consistency confidence threshold*) is the threshold that determines when multiscale consistency regularization is applied (when the maximum prediction probability for any category in any pixel of the average q is above ρ). r (*curve fitting threshold*) is the threshold that controls label correction for each semantic category (see Equation equation 2). τ (*label correction confidence threshold*) is the threshold that determines which pixels are corrected by the model prediction. Only the pixels with confidence (maximum prediction probability for any category) above τ are corrected. Note that we report ablation studies for most of these hyperparameters in Section C.

Dataset	λ	ρ	r	τ
PASCAL VOC 2012	1	0.8	0.9	0.8
SegThor	1	0.8	0.9	0.7

Table 7. Complete list of ADELE hyperparameters for PASCAL VOC 2012 [13] and SegTHOR [27].

B.2. Medical segmentation with simulated noise: SegTHOR

Here we provide implementation details for our experiments on SegTHOR (Supplementary material for Section 4 in the main paper).

Details of Noise Synthesis. We apply dilation and erosion on the ground-truth segmentation masks to simulate "over-annotation" and "under-annotation" labels respectively. In particular, "over-annotation" labels tend to assign background pixels surrounding a target organ to the class of this organ. On the contrary, "under-annotation" labels tend to assign a foreground pixel on the edge of a target organ to the background class. These two noise patterns have been previously utilized to simulate

Algorithm 1: Pseudocode for Proposed Annotation Correction.

Require: $\boldsymbol{y}_i$ $1 \leq i \leq N$ // training noisy annotations
Require: $\boldsymbol{x}_i$, $1 \leq i \leq N$ // training images
Require: NN($\boldsymbol{x}$) // segmentation network
Require: C = number of categories
Require: t the training epoch
Require: $f_c(t)$, $1 \leq c \leq C$ // curve function fitted on training IoU for each category c
Require: τ = threshold, $0 < \tau < 1$
Require: r = threshold for when to correct annotation
Require: S = set of rescaling transforms
for each minibatch B **do**
  for b in B **do**
    for s in S **do**
      $p_{bs} = \text{NN}(S(\boldsymbol{x}_i))$ // evaluate the network to obtain model's outputs corresponding to each scaled version of inputs
    end for
    $q_b \leftarrow \frac{1}{S}\sum_{s=1}^{S} p_{bs}$ // obtain the averaged outputs across different scales
  end for
  for each class c in $[1, C]$ **do**
    If $|f_c'(1) - f_c'(t)|/|f_c'(1)| > r$ **do** // when growth of IoU slowdown
      for image x_b containing object of category c **do**
        $\boldsymbol{y}_b[q_{bc} \geq \tau] = \arg\max q_b[q_{bc} \geq \tau]$; // convert outputs to hard label and conduct pixel-wise annotations correction
      end for
  end for
return $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$ // corrected annotations

common errors that human annotators make during manual segmentation [63]. In our experiments, we randomly choose the type and degree of synthetic noise that will be applied to each example.

Training. For SegTHOR [27], we use one NVIDIA V100 GPU to train the model. We train the UNet [38] using SGD optimizer. To optimize the hyper-parameters, we search for the learning rate in $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ and set to 0.01. λ, ρ, r and τ are set according to Table 7. We trained our model for 100 epochs with a batch size of 5.

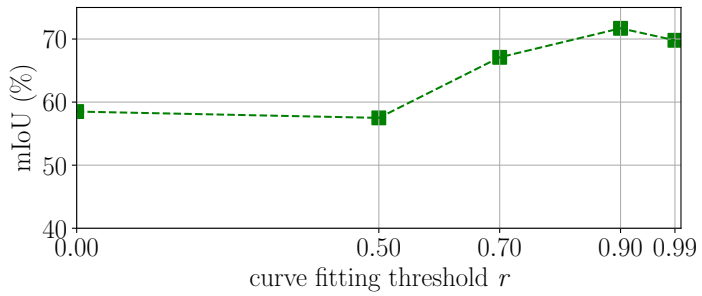
Inference during testing. We conduct the single-scale evaluation using the input without augmentations.

Generating model prediction to conduct label correction. For the medical dataset, since we do not apply any augmentation to the input images, we directly use the outputs during training at every iteration to correct the labels (the model output are processed with $\arg\max$ to produce hard labels for label correction). This is in contrast to PASCAL VOC where we compute the outputs at the end of each epoch, as explained in Section C.1.

B.3. Weakly Supervised Semantic Segmentation: PASCAL VOC 2012

We provide the implementation details of the model for PASCAL VOC 2012. (Supplementary material for Section 5).

Training. For PASCAL VOC 2012 [13], we use two NVIDIA Quadro RTX 8000 GPUs to train the model. We use the official code of AffinityNet [3], SEAM [52] and ICD [14] to generate initial pixel-level annotations that are used for training the segmentation network. The experimental settings to train the segmentation network follow [3, 41, 52, 61] in which DeepLabv1 [8] is adopted. The model uses ResNet38 [56] as a backbone, with the initial weight load from ImageNet [12] pretrained classfication model. Following the same settings as SEAM [52], we use a SGD optimizer with momentum 0.9 and weight decay $5e^{-4}$. The initial learning rate lr_{init} is set to 0.001, and reduced following a polynomial function of the



r	Best val	Last epoch	Max test
0.0	58.5	55.5	58.5
0.5	57.5	55.4	57.9
0.7	67.1	66.1	67.1
0.9	**71.7**	**71.0**	**71.7**
0.99	69.8	64.7	69.8

Figure 20. Ablation study for r on SegTHOR. We fixed the other hyperparameters to the default settings in Table 7. We report the test mIoU (%) at the best validation epoch on the **Left**, the full result is shown on the **Right**.

iteration number itr: $\text{lr}_{itr} = \text{lr}_{init}\left(1 - \frac{itr}{max\_itr}\right)^{\gamma}$ with $\gamma = 0.9$. We train our segmentation network for 20000 iterations ($max\_itr = 20000$) with a batch size of 10. The input images are randomly scaled and then randomly cropped to 448×448.

Inference during testing. During testing, we use the same inference pipeline as SEAM [52], which includes multi-scale inference [3, 14, 52, 64] and CRF [25].

Generating model predictions to conduct label correction. Updating the model using the model predictions after processing each batch would be difficult for the PASCAL VOC 2012 dataset. The reason is that random data augmentations (*e.g.* rescaling, cropping, random changing contrast in the image, *etc.*) are often applied [3, 14, 52, 64] and these augmentations would need to be inverted in order to use the output for label correction (otherwise the predictions would be inconsistent across training epochs). In fact, some augmentations are not invertible at all, *e.g.* cropping cuts the object out thus is not invertible, rescaling might result in loss of information thus is not invertible as well. In order to avoid this issue on the PASCAL VOC dataset, we evaluate the model at the end of each training epoch on inputs without any random augmentation, and then use the model outputs to perform label correction (the model output are processed with $\arg\max$ to produce hard labels for label correction).

C. Additional ablation studies

C.1. Medical segmentation with simulated noise: SegTHOR

Here we report additional ablation studies for different components in the proposed label correction method on SegTHOR dataset (Supplementary material for Section 4).

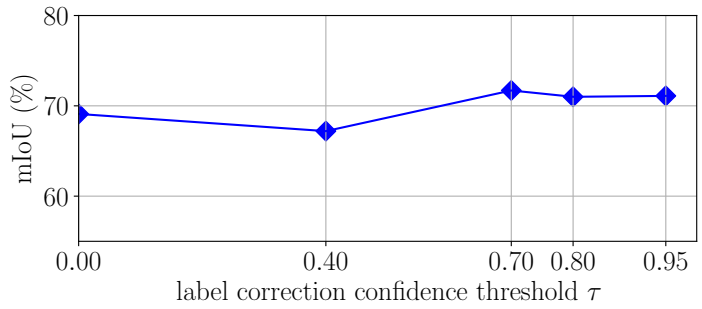
Different options for label correction As described in Section 2.2, ADELE uses the model outputs to correct labels. In this section, we compare two options for computing these outputs.

- *Iteration*: the outputs are computed after each training iteration.
- *Epoch*: the outputs are computed at the end of each training epoch.

Table 8 shows that on SegTHOR *Iteration* performs slightly better on the best test mIoU than *Epoch*, and outperforms it substantially on the mIoU at the last epoch, suggesting that *Iteration* is more effective in preventing memorization. Both two options are improving results with respect to the baseline.

Method	Best val	Last epoch	Max test
ADELE (*Iteration*)	**71.1 ± 0.7**	**70.8 ± 0.7**	**71.2 ± 0.6**
ADELE (*Epoch*)	69.6 ± 0.8	64.1 ± 1.3	70.2 ± 0.5

Table 8. mIoU(%) of ADELE on SegTHOR when label correction is based on model output at the end of each iteration or each epoch. The former achieves better results.



τ	Best val	Last epoch	Max test
0	69.1	68.8	69.1
0.4	67.2	67.1	68.0
0.7	**71.7**	**71.0**	**71.7**
0.8	71.0	70.3	71.0
0.95	71.1	71.2	71.3

Figure 21. Ablation study for τ on SegTHOR. We fixed the other hyperparameters to the default settings in Table 7. We report the test mIoU (%) at the best validation epoch on the **Left**, the full result is shown on the **Right**. The result shows that ADELE is not very sensitive to the value of τ.

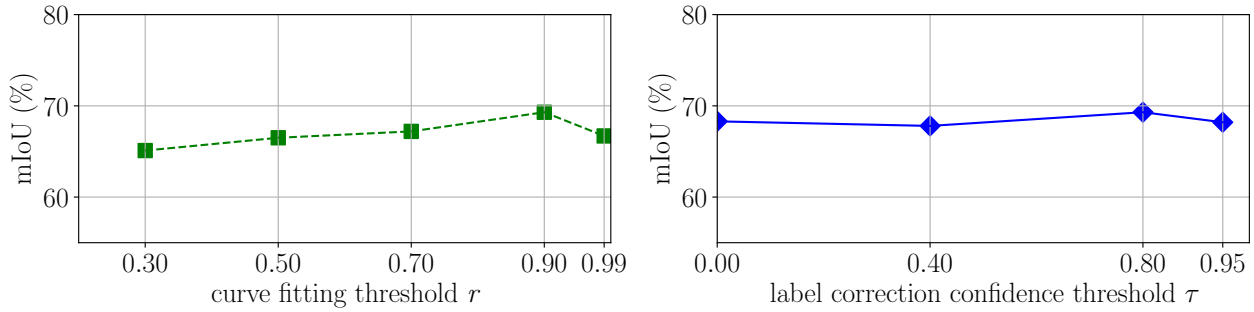


Figure 22. Ablation study for r and τ on PASCAL VOC. We fixed the other hyperparameters to the default settings in Table 7. We report the validation mIoU (%) at the last training epoch.

r and τ for label correction. We report the ablation for r and τ for ADELE on SegTHOR dataset in Figure 20 and 21 respectively. For the r value, we observe similar results as in the PASCAL VOC dataset. If the r value is too small (*e.g.* 0, 0.3) and label correction is conducted too early, the network has not been trained well. This degrades the label correction quality, which hurts the generalization of the model. If the r value is too large (*e.g.* 0.99) then the network barely conduct label correction for any class before stopping training. Results are again very robust to the choice of τ value.

C.2. Weakly Supervised Semantic Segmentation: PASCAL VOC

r and τ for label correction. We report the ablation result for r and τ of ADELE on PASCAL VOC dataset in Figure 22. Small r values encourage the model to conduct label correction earlier, larger r values delay correction. If the r value is too small (*e.g.* 0.3) and label correction is conducted too early, the network has not been trained well. This degrades the label correction quality, which hurts the generalization of the model. If the r value is too large (*e.g.* 0.99), then the method barely conducts label correction for any class before the end of training, which results in a performance similar to the case without label correction. We observe that our model is very robust to the choice of τ.