Adaptive Superpixel for Active Learning in Semantic Segmentation

Hoyoung Kim[1]   Minhyeon Oh[2]   Sehyun Hwang[2]   Suha Kwak[1,2]   Jungseul Ok[1,2*]

Graduate School of AI, POSTECH[1],   Dept. of CSE, POSTECH[2]

{cskhy16, minhyeonoh, sehyun03, suha.kwak, jungseul}@postech.ac.kr

Abstract

Learning semantic segmentation requires pixel-wise annotations, which can be time-consuming and expensive. To reduce the annotation cost, we propose a superpixel-based active learning (AL) framework, which collects a dominant label per superpixel instead. To be specific, it consists of adaptive superpixel and sieving mechanisms, fully dedicated to AL. At each round of AL, we adaptively merge neighboring pixels of similar learned features into superpixels. We then query a selected subset of these superpixels using an acquisition function assuming no uniform superpixel size. This approach is more efficient than existing methods, which rely only on innate features such as RGB color and assume uniform superpixel sizes. Obtaining a dominant label per superpixel drastically reduces annotators' burden as it requires fewer clicks. However, it inevitably introduces noisy annotations due to mismatches between superpixel and ground truth segmentation. To address this issue, we further devise a sieving mechanism that identifies and excludes potentially noisy annotations from learning. Our experiments on both Cityscapes and PASCAL VOC datasets demonstrate the efficacy of adaptive superpixel and sieving mechanisms.

1. Introduction

With the advent of deep learning, many computer vision tasks including semantic segmentation have dramatically evolved in recent years. Such advances are thanks to complex deep network models that can learn huge datasets. However, labeling such large datasets is prohibitively time-consuming and labor-intensive, in particular, for semantic segmentation tasks that demand a dense annotation on each pixel [8, 11]. Active learning (AL) offers an approach to alleviate the annotation cost by selectively querying only the most informative samples to annotators.

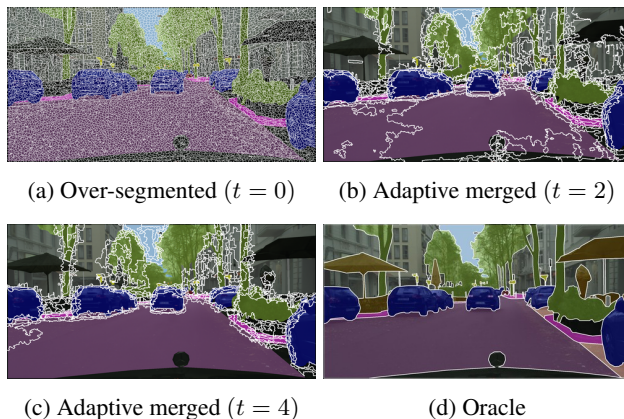Designing an effective form of annotation query is crit-

---

*Corresponding author.

(a) Over-segmented ($t = 0$)   (b) Adaptive merged ($t = 2$)

(c) Adaptive merged ($t = 4$)   (d) Oracle

Figure 1: *Examples of adaptive superpixels.* (a) We begin active learning with over-segmented superpixels. (b, c) In each round t, we merge superpixels in an adaptive manner using the model from the previous round. (d) As the round progresses, adaptive superpixels look similar to oracle ones.

ical in practice as it determines the actual annotation cost such as the number of clicks required and the informativeness per annotation query. For semantic segmentation, an image-wise query can be asked for a complete annotation on the semantic of every pixel in an image [9, 10, 34, 38, 40]. This is a daunting task requiring an enormous amount of clicks to indicate boundaries (using polygons or contours) for each semantic segment or to annotate semantic pixel-wisely, while the diversity of contexts which we can observe in a single image is restricted. Alternatively, one can design a *region-based* query enquiring only about the dominant label of a small region such as rectangle patch [5, 27, 37] or superpixel [4, 33]. This is known to be simple yet effective as it requires only a single click per query while enabling AL to put more focus on significant regions and to avoid annotation wastes.

AL with the region-based query needs a delicate generation of candidate regions to be queried. A small region size dilutes the budget efficiency, whereas the dominant labeling even by a perfect annotator is prone to give noisy labels
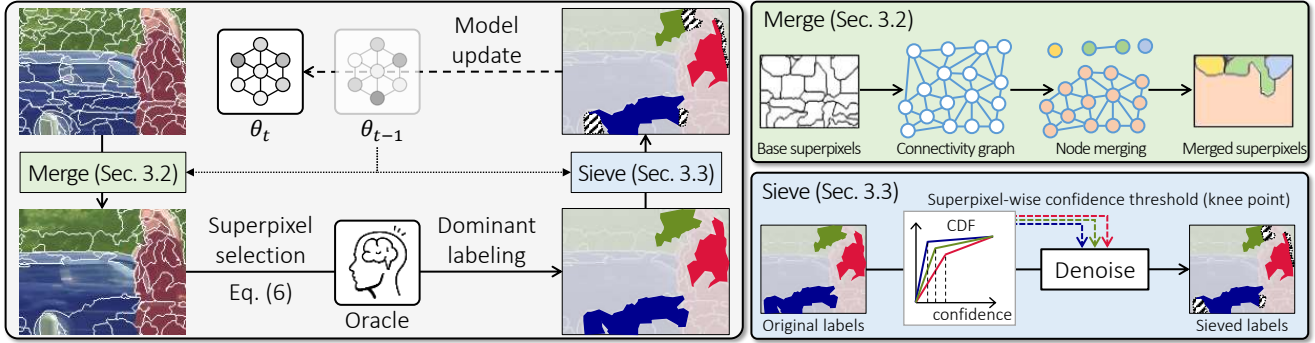
Figure 2: *An overview of the proposed framework.* In each round t, we merge superpixels with a graph using the latest model, and obtain dominant labels for selected superpixels. The dominant labels are selectively propagated to pixels with confidence above the detected knee point, resulting in the creation of a sieved dataset. Finally, we train a model with the sieved one.

when regions are too large to be consisting of pixels with a single class. However, the previous works [5, 27, 33] rely on a fixed candidate set of regions of uniform size, while we could adjust the size and shape of candidate regions as we train the semantic segmentation model over rounds of AL. This limitation remains even in recent work [4] with superpixel candidates providing less risk of noisy labels than rectangle ones since the superpixels are produced, only at the beginning, by a conventional superpixel algorithm, where conventional superpixel algorithms [1, 32, 35] cluster adjacent pixels of similar innate features (*e.g.*, color) with implicit or explicit regularization to make similar sizes or shapes of superpixels, *i.e.*, limited freedom of query region.

In this paper, to fully enjoy the benefit in terms of annotation cost while suppressing the risk of noisy labels, we devise an AL framework, illustrated in Figure 2, consisting of adaptive merging and sieving methods. The adaptive merging method repeatedly evolves the candidate superpixels for dominant labeling at every round with the latest model and no explicit regularization on the size and shape of superpixels. This indeed enables the continual improvement of the superpixels' ability to accurately capture the boundaries of semantic objects (Figure 1b and 1c), and a proper variation in the sizes and shapes of superpixels, *i.e.*, larger superpixels being attached to larger semantic objects (*e.g.*, road and building) and smaller ones to smaller objects (*e.g.*, human and vehicle) as shown in the ideal ones (Figure 1d).

Given the adaptive superpixels, we establish a corresponding acquisition function being aware of irregular superpixel sizes. It prioritizes uncertain superpixels of rare classes in order to query the most informative superpixels while balancing class distributions in the entire annotations. In addition, to alleviate the inevitable noise in the dominant labeling, we propose a sieving technique that excludes labeled pixels of high potential risks of being different classes than the dominant one. To be specific, we identify such pixels of potentially noisy labels by per-superpixel sieving with

distinct thresholds over superpixels. This provides stabler denoising than uniform sieving with a constant threshold, which might aggravate class imbalance in the sieved annotations.

Through the integration of adaptive merging and sieving into an AL framework, we achieve improved accuracy and budget-efficiency over a baseline method. Notably, the merging demonstrates effectiveness under small-sized superpixels, while the sieving plays a critical role given largesized superpixels. Moreover, we show a consistent improvement over existing methods in various settings. We provide a thorough justification of the proposed method using various quantitative measures, where we introduce a new evaluation metric for superpixel algorithms that assesses both (achievable) accuracy and recall, where the recall is overlooked in the existing one, the achievable segmentation accuracy (ASA) [22] but important in the context of AL. This may give new insights into developing superpixel algorithms.

Our main contributions are summarized as follows:

- We propose an adaptive merging algorithm where superpixels are updated at each round (Section 3.2), and show the effectiveness of adaptive merging rather than only merging once (Section 4.2).

- We alleviate the side effect of noisy labels via a sieving technique (Section 3.3), and demonstrate especially efficient under large superpixels (Section 4.2).

- In various realistic experiments, we demonstrate the consistent improvement of the proposed AL framework, consisting of the adaptive merging and sieving methods with the dedicated acquisition function, over existing ones (Section 4.2).

- We provide an insightful analysis on proper superpixels for AL with the new evaluation metric of superpixel algorithms being aware of usage in AL (Section 5.1).

2. Related work

Active learning for segmentation. To reduce the labeling cost of semantic segmentation, active learning for segmentation selectively collects labels among unlabeled samples, and they utilize different predefined labeling units. Early approaches [34, 40] perform image-wise selection and mask labeling. Patch-based methods [5, 7, 16, 25, 37] divide images into rectangular patches and provide mask label [5, 16, 37] or polygon overlay of an object [7, 25] within the selected patch. Recently, superpixel-based approaches [4, 33] split images to perceptually meaningful regions called superpixel by running an off-the-shelf over-segmentation algorithm [1, 28, 35]. Each superpixel is labeled with a single dominant class, and thus it can be obtained efficiently [4], while a label noise may occur depending on the quality of the superpixel. We present a new efficient labeling unit, that is initialized with the superpixel but its quality continuously improves by the proposed merging algorithm. To the best of our knowledge, the proposed method is the first approach to improve the labeling units during active learning for segmentation.

Learning from noisy labels for segmentation. Considering the difficulty in acquiring high-quality labels [8], semantic segmentation often suffers from noisy annotations. Previous studies address the label noise by using gradient similarity to the clean label [41], structural constraints [2, 21], and noise-aware loss [26, 39]. A recent approach captures the moment when different classes memorize noisy labels [23]. Most of these methods [21, 26, 39] utilize a single confidence threshold to detect label noise within data. Unlike previous approaches, we propose to detect an adaptive confidence threshold per every superpixel queried, using the Kneedle algorithm [30] (Section 3.3). Filtering with the sample-adaptive threshold prevents superpixels with low overall confidence or superpixels containing minor classes from being ignored.

Superpixel mechanisms and their evaluation metric. Numerous studies segment an image into superpixels to reduce the computation burden of pixels. Cut-based approaches [22, 32, 36, 42] create superpixels by adding multiple minimum cuts into a graph with pixel nodes. Other methods evolve homogeneous clusters from the initial set of points [1, 20]. For real-time applications, a simple hill-climbing optimization is utilized to enforce color similarity [35]. Most of methods aim at generating superpixels of predefined size or shape, and the generated superpixels are evaluated by achievable segmentation accuracy and boundary recall compared with ground truth [22, 35] or by examining the regularity in superpixel shape [15, 24, 31]. To save labeling costs in active learning, it is more important to obtain superpixels as close to the ground-truth segments as possible without such constraints on the shape or size of

superpixels. To this end, we propose the merging method (Section 3.2), and a new evaluation metric of superpixel mechanism, that also takes account of the size of ground-truth segments (Section 5.1). The proposed metric not only highlights the difference of the ideal superpixel required in active learning than that in the previous context, but also gives a guideline to develop superpixel algorithms for active learning.

3. Proposed framework

Given an unlabeled image set $\mathcal{I}$, we consider an active learning scenario with dominant labeling, where a query asks an oracle annotator for the dominant class label $\mathrm{D}(s) \in \mathcal{C} := \{1, 2, ..., C\}$ of an associated superpixel s, and we issue a batch $\mathcal{B}_t$ of B queries for each round t. Once we enquire the batch $\mathcal{B}_t$, we train a model θ_t based on the annotations obtained so far. Recalling a superpixel s is a cluster of neighboring pixels, the dominant labeling demands much less annotation effort than the pixel-wise labeling on every individual pixel x in the same superpixel s or manual segmentation to indicate boundaries separating semantics. The benefit becomes greater with larger superpixels. Meanwhile, it is prone to noisy labeling as superpixels can be blunt, *i.e.*, including pixels of different semantics.

In order to fully enjoy the benefit in terms of annotation cost while suppressing the risk of noisy labels, our framework begins with a warm-up round ($t = 0$; Section 3.1; line 1-2 in Algorithm 1) to prepare an initial model from random querying and iterates subsequent rounds ($t = 1, 2, \dots$) with the adaptive merging (Section 3.2; line 4-5 in Algorithm 1) and sieving (Section 3.3; line 6-7 in Algorithm 1) methods to evolve superpixels for dominant labeling round by round and filter out annotations with the high risk of noisy labels given the latest model. The overall procedure is summarized in Figure 2 and Algorithm 1.

3.1. Warm-up round

The adaptive merging and sieving methods demand a trained model. To obtain an initial model, we start with a canonical warm-up round, which is identical to the first round of previous work [4]. We first use an off-the-shelf superpixel algorithm, namely SEEDS [35], to partition the pixels in each image $i \in \mathcal{I}$ into a set $S_0(i)$ of superpixels, and to produce a base segmentation $\mathcal{S}_0 := \bigcup_{i \in \mathcal{I}} S_0(i)$. Querying a batch $\mathcal{B}_0$ of B superpixels randomly selected from $\mathcal{S}_0$, we then train a model θ_0 using the dominant labels for $\mathcal{B}_0$. Specifically, to obtain θ_0, we first initialize θ at a model pretrained on ImageNet, and then train it to minimize the following cross-entropy (CE) loss:

$$\hat{\mathbb{E}}_{(x,y) \sim \mathcal{D}_0}[\mathrm{CE}(y, f_\theta(x))] , \tag{1}$$

where $\mathcal{D}_0 := \{(x, y) : \exists s \in \mathcal{B}_0, x \in s, y(c) = \mathbb{1}[c = \mathrm{D}(s)]$

Algorithm 1 Proposed Framework

Require: Image set $\mathcal{I}$, batch size B, and final round T.
1: Produce base superpixels $\mathcal{S}_0 := \bigcup_{i \in \mathcal{I}} \mathcal{S}_0(i)$
2: Obtain model θ_0 training with $\mathcal{D}_0$
3: **for** $t = 1, 2, \ldots, T$ **do**
4:     Adaptively merge the base superpixels and obtain
        $\mathcal{S}_t \leftarrow \bigcup_{i \in \mathcal{I}} \text{AM}(S_0(i), \theta_{t-1})$
5:     Select and query B superpixels $\mathcal{B}_t \subset \mathcal{S}_t$ with (7)
6:     Sieve $s \in \bigcup_{t'=0}^{t} \mathcal{B}_{t'}$ and obtain $\mathcal{D}_t$ in (9)
7:     Obtain model θ_t training with the sieved $\mathcal{D}_t$
8: **return** θ_T

Algorithm 2 Adaptive Merging (AM)

Require: Base superpixels S, model θ, and threshold ϵ.
1: Set $S' \leftarrow \emptyset$ and $\mathcal{G}(S) \leftarrow (S, \mathcal{E}(S))$
2: Mark s as unexplored for each $s \in S$
3: **for** $s \in S$ in descending order of $u_\theta(s)$ **do**
4:     **if** s is unexplored **then**
5:         $S' \leftarrow S' \cup \{\text{MERGE}(s, f_\theta(s); \mathcal{G}, \theta)\}$
6: **return** S'
7: **procedure** MERGE($s, f; \mathcal{G}, \theta$)
8:     Mark s as explored and set $s' \leftarrow s$
9:     **for** each neighbor n of s in $\mathcal{G}$ **do**
10:       **if** n is unexplored and $d_{\text{JS}}(f \parallel f_\theta(n)) < \epsilon$ **then**
11:         $s' \leftarrow s' \cup \text{MERGE}(n, f; \mathcal{G}, \theta)$
12:     **return** s'

$\forall c \in \mathcal{C}\}$ is the training data for round $t = 0$ without sieving, and $f_\theta(x) \in \mathbb{R}^{|\mathcal{C}|}$ is θ's estimate of class probability on pixel x.

We remark that we use the initial model θ_0 for round $t = 1$. In our framework, SEEDS to generate $\mathcal{S}_0$ can be replaced with any other unless $\mathcal{S}_0$ is a fair over-segmentation of semantics with a low risk of noisy labeling while partially enjoying the benefit of low annotation cost. We note that SEEDS clusters neighboring pixels of similar colors while a semantic consists of multiple colors, typically. SEEDS, ready-to-use in OpenCV [3], easily provides the desired over-segmentation [4] and a decent performance of θ_0. In addition, the warm-up round with SEEDS corresponds to that in existing work [4]. Hence, this also enables a fair comparison of our main contributions, *i.e.*, adaptive merging and sieving methods, to existing works.

3.2. Adaptive merging

In advance of dominant labeling in round $t \geq 1$, we first merge the base superpixels in $\mathcal{S}_0$ to obtain $\mathcal{S}_t$ using the model θ_{t-1} from the previous round. We then select a batch $\mathcal{B}_t$ of B superpixels from $\mathcal{S}_t$ to be annotated using an acquisition function that prioritizes uncertain superpixels of rare class labels. For simplicity, we often omit the subscript $t - 1$ and write θ for θ_{t-1}.

Adaptive merging. To obtain $\mathcal{S}_t := \bigcup_{i \in \mathcal{I}} S_t(i)$, the merging process converts base superpixels $S_0(i)$ into merged ones $S_t(i)$ for each image $i \in \mathcal{I}$. We hence focus on how we merge given base superpixels S for an image. To begin with, we convert the superpixels S into a connected graph $\mathcal{G}(S) = (S, \mathcal{E}(S))$ where S is the set of nodes, each of which corresponds to a base superpixel $s \in S$, and $\mathcal{E}(S)$ is the edge set such that $(s, n) \in \mathcal{E}(S)$ if a pair of superpixels $s, n \in S$ are adjacent. Starting from a root node $s \in S$, we then merge neighboring superpixels of similar class predictions with the root s along the breadth-first search tree. To be specific, a neighbor n is amalgamated with root s only if

$$d_{\text{JS}}\big(f_\theta(s) \parallel f_\theta(n)\big) < \epsilon \, , \qquad (2)$$

where $f_\theta(s) := \frac{\sum_{x \in s} f_\theta(x)}{|\{x : x \in s\}|}$ is the averaged class prediction of superpixel $s \in S$, and d_{JS} is a symmetric measure of discrepancy between two distributions, namely the square root of Jensen-Shannon (JS) divergence. More formally,

$$d_{\text{JS}}(p \parallel q) := \sqrt{\frac{d_{\text{KL}}(p \parallel \frac{p+q}{2}) + d_{\text{KL}}(q \parallel \frac{p+q}{2})}{2}} \, , \qquad (3)$$

where d_{KL} is the Kullback-Leibler divergence. Once every node has been either merged to a root or played as a root, we collect the merged superpixels into $S_t(i)$. The merging process is formally described in Algorithm 2.

Recalling (2) and the fact that d_{JS} is a distance metric, we can guarantee that any pair of superpixels s and n has the prediction discrepancy at most 2ϵ and thus similar uncertainty and predicted label if they are merged. Hence, the threshold ϵ governs the impurity of predictions in a merged superpixel. We also remark that the merging process is fully dedicated to collecting pixels of similar predictions as a part of saving the annotation budget for querying similar pixels repeatedly. Hence, the merged superpixels can have various sizes differently from existing superpixel algorithms that regularize the superpixel size to be even [15, 24, 31].

Acquisition function. From the merged superpixels $\mathcal{S}_t$, we then select a batch $\mathcal{B}_t \subset \mathcal{S}_t$ of size B to be labeled, according to an acquisition function that estimates the benefit from labeling a merged superpixel, where the benefit would be huge for uncertain superpixels of rare class labels. In what follows, we define an uncertainty measure of superpixel in (5) and a popularity estimate of class in (6), and then introduce an acquisition function in (7).

Recalling $f_\theta(x) \in \mathbb{R}^{|\mathcal{C}|}$ is the probability such that $f_\theta(c; x)$ is the estimated probability that the class c of pixel x, we adapt best-versus-second-best [17] for uncertainty

measures of pixel x and superpixel s as follows:

$$u_\theta(x) := \frac{\max_{c \in \mathcal{C} \setminus \{y_\theta(x)\}} f_\theta(c; x)}{\max_{c \in \mathcal{C}} f_\theta(c; x)} , \qquad (4)$$

$$u_\theta(s) := \frac{\sum_{x \in s} u_\theta(x)}{|\{x : x \in s\}|} , \qquad (5)$$

where $y_\theta(x) := \arg\max_{c \in \mathcal{C}} f_\theta(c; x)$ is the estimated dominant label of pixel x in a given model θ.

We then define a popularity estimate $p(c; \theta)$ of class $c \in \mathcal{C}$ given θ as follows:

$$p(c; \theta) := \frac{|\{x : \exists s \in \mathcal{S}_t, D_\theta(s) = c, x \in s\}|}{|\{x : \exists s \in \mathcal{S}_t, x \in s\}|} , \qquad (6)$$

where $D_\theta(s) := \arg\max_{c \in \mathcal{C}} |\{x \in s : y_\theta(x) = c\}|$ is the majority of predicted labels in superpixel s. We note that low $p(c; \theta)$ implies that class c is rare in the prediction of θ. It is noteworthy that we compute the class popularity in pixel-level due to the various sizes of our merged superpixels, while the previous work [4] proposes a superpixel-wise class popularity, $\frac{|\{s : D_\theta(s) = c, s \in \mathcal{S}_t\}|}{|\{s : s \in \mathcal{S}_t\}|}$, assuming superpixels of uniform size.

Using the uncertainty $u_\theta(s)$ in (5) and the class popularity $p(c; \theta)$ in (6), we define the following acquisition function $a(s; \theta)$ prioritizing uncertain superpixels of rare classes:

$$a(s; \theta) := u_\theta(s) \exp\left(-p(D_\theta(s); \theta)\right) . \qquad (7)$$

We select B superpixels of highest values of $a(s; \theta_{t-1})$ from the merged $\mathcal{S}_t$ for query batch $\mathcal{B}_t$.

Remarks. We note that it is possible to produce $\mathcal{S}_t$ from scratch rather than from base segmentation $\mathcal{S}_0$. To reduce the computational cost for the adaptive merging process, we however compose $\mathcal{S}_t$ by merging base superpixels in $\mathcal{S}_0$ from SEEDS, which is known to generate an over-segmentation of semantics. Moreover, it is computationally expensive to explore all the possible mergers and obtain $\mathcal{S}_t$ followed by the query selection. We hence conduct the merging process only for a certain portion of base superpixels with the highest values of uncertainty (c.f., line 3 in Algorithm 2) and then select $\mathcal{B}_t$ to be queried since the acquisition function would select merged superpixels of high uncertainty in the end. Further details are presented in Appendix B.

3.3. Sieving

Despite the sophisticated design of the adaptive merging, a queried superpixel can inevitably include pixels of classes different from the dominant one, in particular, as we select superpixels of which model predictions are unsure. Hence, the dominant labeling is liable to make noisy annotations. To alleviate such side effects of the dominant labeling, we propose a simple sieving technique that filter out pixels that have high potential risks of being different classes than the dominant one. We observe that for a queried superpixel s and given model θ, the risk of mismatch between the dominant label $D(s)$ and the true label of pixel $x \in s$ would be high when $f_\theta(D(s); x)$ is low. From this observation, we define

$$h(s; \theta) := \{x \in s : f_\theta(D(s); x) \geq \phi(s; \theta)\} , \qquad (8)$$

where $\phi(s; \theta)$ is a knee point of the cumulative distribution function of values of $f_\theta(D(s); x)$ in superpixel s, detected by Kneedle algorithm [30]. In addition, the knee point detection allows us to have a tailored sieving threshold to each superpixel. This is important to avoid the case that the remained pixels are heavily biased to relatively easy labels after sieving. Further details are in Appendix C.

We revisit all the queried superpixel $s \in \bigcup_{t'=0}^{t} \mathcal{B}_{t'}$ and sieve them using (8) with the latest model θ_{t-1} since the model evolves round by round. We finally obtain the following sieved dataset $\mathcal{D}_t$ for round $t \geq 1$:

$$\mathcal{D}_t := \left\{ (x, y) : \begin{array}{l} \exists s \in \cup_{t'=0}^{t} \mathcal{B}_{t'}, \ x \in h(s; \theta_{t-1}), \\ y(c) = \mathbb{1}[c = D(s)] \ \forall c \in \mathcal{C} \end{array} \right\} . \qquad (9)$$

Analogously to the warm-up round, initializing model θ at a model pretrained on ImageNet, we obtain θ_t trained to mainly minimize the following CE loss:

$$\hat{\mathbb{E}}_{(x,y) \sim \mathcal{D}_t}[\text{CE}(y, f_\theta(x))] . \qquad (10)$$

4. Experiments

4.1. Experimental setup

Datasets. We use two semantic segmentation datasets: Cityscapes [8] and PASCAL VOC 2012 (PASCAL) [13]. Cityscapes comprises 2,975 training and 500 validation images with 19 classes, while PASCAL consists of 1,464 training and 1,449 validation images with 20 classes.

Implementation details. We adopt DeepLab-v3+ architecture with Xception-65 [6] as our segmentation backbone. During training, we use the SGD optimizer with a momentum of 0.9 and set a base learning rate to 7e-3. We decay the learning rate by polynomial decay with a power of 0.9. For Cityscapes, we resize training images to 769×769 and train a model for 60k iterations with a mini-batch size of 4. Similarly, for PASCAL, we resize training images to 513×513 and train a model for 30k iterations with a mini-batch size of 12. Unless specified, we set the value of ϵ to 0.1.

Baseline methods. We compare our algorithm to *SP* [4], the state-of-the-art superpixel-based active segmentation method. Our algorithm applies two proposed processes in each round: merging and sieving. We call our complete
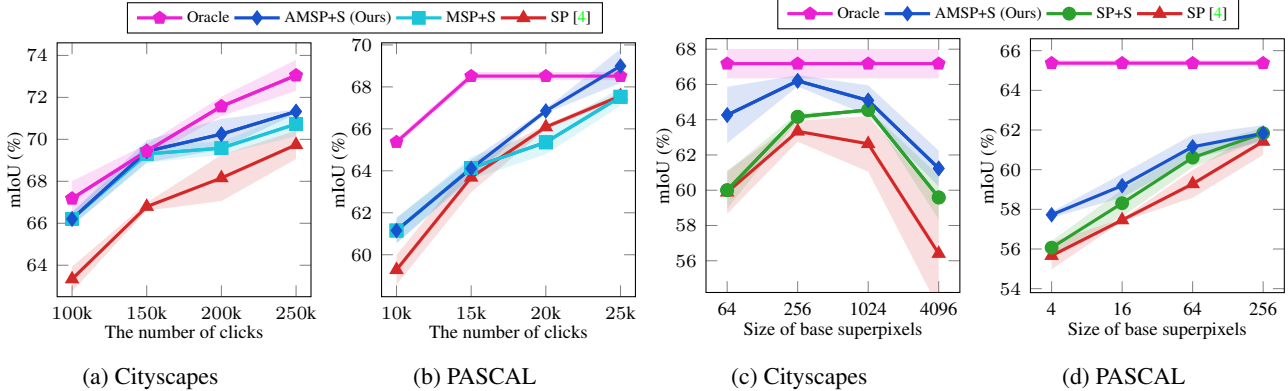
Figure 3: *Effect of adaptive superpixels.* (a, b) mIoU versus the number of clicks as budget. (c, d) mIoU versus the size of base superpixels. Each experiment is conducted with three trials and the shaded region indicates ranges.

method including adaptive merging as *AMSP+S*, while the partial version that only uses the sieving without the merging is called *SP+S*. Additionally, we evaluate the modified version of our method that performs merging only once in the second round, called *MSP+S*. Note that *AMSP+S* and *MSP+S* are identical until the second round.

Oracle baseline. The adaptive superpixel aims to merge every connected region with the same class labels. Thus, the upper bound of it is to consider each region separated by the ground truth mask as a superpixel. We refer to such ideal regions as oracle superpixels in Figure 4c. An active learning model trained using the oracle superpixels is called *Oracle*. Details are in Appendix D. As the number of oracle superpixels is limited, all of them are eventually labeled as the round progresses, and the performance of the trained model becomes equivalent to that of the pixel-wise fully supervised model. We report 100% and 90% of the *Oracle* performance for Cityscapes and PASCAL, respectively.

Evaluation protocol. We set the average size of the superpixels to 256 and 64 pixels on Cityscapes and PASCAL, respectively, for all experiments except for one where we adjust the size. Following *SP* [4], we use the number of clicks as the labeling budget. We conduct 5 rounds of data sampling, where we allocate a budget of 50k and 5k for each round on Cityscapes and PASCAL, respectively. In the first round, we randomly select superpixels to train a model, ensuring that all methods start at the same performance. We evaluate the trained model with mean Intersection-over-Union [11] on the validation images. We emphasize that the average size of superpixels containing 64 pixels is more efficient on Cityscapes, as detailed in Appendix A.

4.2. Effect of adaptive superpixels

Multi-round scenario. In Figures 3a and 3b, we compare the performance of the proposed method to *SP* [4] varying budget for both of Cityscapes and PASCAL. Note that

the performance for round 0, *i.e.*, 50K budget, is omitted as each method has the same performance at the warm-up round. The results show that our adaptive superpixel (*AMSP+S*) clearly outperforms the previous art in every budget setting on both of the datasets. In particular, the *AMSP+S* with only 150k clicks outperforms the previous art with 250k clicks in Cityscapes. In the final round, the proposed method recovers 97% and 92% of the *Oracle* performance for Cityscapes and PASCAL, respectively. To show the effectiveness of our adaptive approach, we compare *AMSP+S* to its one-shot merging version *MSP+S* in Figures 3a and 3b. On both datasets, adaptive feature of *AMSP+S* shows performance gain especially for the last two rounds. The experiments conducted for additional rounds can be found in Appendix A.

Multi-size scenario. The size of superpixels is an essential hyperparameter in superpixel-based AL, affecting both the quantity and quality of labels. In Figures 3c and 3d, we compare the proposed method to *SP* [4] varying the base superpixel size for both of Cityscapes and PASCAL, in the second round. Our adaptive superpixel (*AMSP+S*) outperforms the previous art in various superpixel sizes on both of the datasets. We also evaluate sieving only version (*SP+S*) of our method, which quantifies contribution of each components in our method. The performance improvement between *SP* and *SP+S* shows our sieving is especially helpful for large superpixels, and the performance gap between *SP+S* and *AMSP+S* shows our merging is especially effective for small superpixels. Thanks to the proposed sieving and merging, *AMSP+S* are comparably robust to the change of the superpixel size than *SP*.

Qualitative results. The quality of the proposed adaptive superpixel is illustrated in Figure 4. As shown in Figure 4a, superpixels used in the previous study [4] have uniform sizes for all areas regardless of their content. In contrast, Figure 4b demonstrates that adaptive superpixels accurately

(a) Base superpixels [35]   (b) Merged superpixels (Ours)   (c) Oracle superpixels
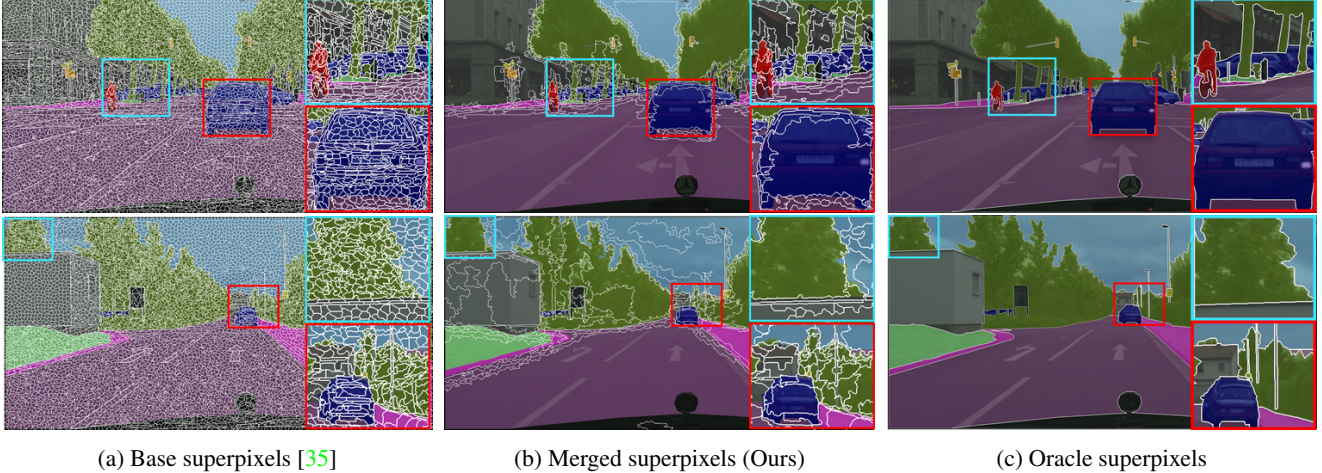
Figure 4: *Qualitative results of adaptive superpixels.* (a) Base superpixel generated by SEEDS [35] with size 256. (b) Superpixels generated with proposed adaptive merging at round 4. (c) Oracle superpixels generated from the ground truth.

reflect the actual size of the content in images, carefully capturing small object classes while efficiently covering large background classes. More examples are in Appendix F.

5. Analyses of adaptive superpixels

We propose new evaluation metrics to measure the quality of superpixel as a labeling unit for active segmentation, and utilize it to analyze our adaptive superpixels (Section 5.1). We also conduct analyses about the effect ϵ to our adaptive superpixels (Section 5.2). All analyses are conducted on Cityscapes with an average superpixel size of 256 pixels.

5.1. Achievable metrics

While various evaluation metrics for superpixel are presented [15, 22, 24, 31, 35], most of them aims to measure the quality of over-segmentation. For instance, achievable segmentation accuracy (ASA) [22] measures the segmentation accuracy when each superpixel $s \in S$ is associated with the oracle superpixel with the largest overlap. The ASA is calculated as follows:

$$\text{ASA}(S; G) := \frac{\sum_{s \in S} \max_{g \in G} |s \cap g|}{\sum_{s \in S} |s|} , \tag{11}$$

where S and G represent the generated and oracle superpixels from the same image, respectively. As an image becomes more over-segmented, *i.e.*, the superpixel size becomes smaller, the ASA value increases. However, active learning (AL) aims to achieve the maximum benefit with the least amount of labeling effort, and therefore, the number of labels should be taken into account. In addition, the ASA is heavily biased towards classes with a large number of pixels.

In order to measure the suitability of superpixels for AL, we introduce precision and recall between generated and oracle superpixels. A generated superpixel can be viewed as positive on the inside and negative on the outside, and its precision and recall with respect to the corresponding oracle one can be calculated. For all generated superpixels, we define the achievable precision (AP) as follows:

$$\text{AP}(S; G) := \frac{1}{|S|} \sum_{s \in S} \frac{\max_{g \in G} |s \cap g|}{|s|} , \tag{12}$$

where the summation is performed in superpixels, unlike in ASA, which implies pixel-wise precision. As we put the same weight on each superpixel, AP is less influenced by large objects than ASA. We note that AP is different to average precision [12, 29], used in object detection, which utilize the precision and recall curve. We also define the achievable recall (AR) and F1-score (AF) as:

$$\text{AR}(S; G) := \frac{1}{|S|} \sum_{s \in S} \frac{\max_{g \in G} |s \cap g|}{|g'(s; G)|} , \tag{13}$$

$$\text{AF}(S; G) := \frac{2}{|S|} \sum_{s \in S} \frac{\max_{g \in G} |s \cap g|}{|s| + |g'(s; G)|} , \tag{14}$$

where $g'(s; G) := \arg\max_{g \in G} |s \cap g|$ refers to the corresponding oracle superpixel. Details are in Appendix E.

All the metrics evaluate generated superpixels in comparison to oracle ones. However, the size of superpixels is also important besides their quality in AL. Therefore, it is necessary to evaluate the oracle superpixels against the generated superpixels, *i.e.*, ASA$(G; S)$, AP$(G; S)$, AR$(G; S)$ and AF$(G; S)$. We hence propose AF$(G; S)$ defined as:

$$\text{AF}(G; S) := \frac{2}{|G|} \sum_{g \in G} \frac{\max_{s \in S} |g \cap s|}{|g| + |s'(g; S)|} , \tag{15}$$

Methods	ASA($S;G$)	ASA($G;S$)	AP($S;G$)	AR($S;G$)	AF($S;G$)	AP($G;S$)	AR($G;S$)	AF($G;S$)	mIoU
SLIC$_{4096}$	0.887	0.082	0.897	0.046	0.066	0.695	0.259	0.185	53.18
SEEDS$_{4096}$	0.909	0.082	0.900	0.050	0.070	0.665	0.309	0.221	57.61
SLIC$_{256}$	0.956	0.013	0.958	0.007	0.012	0.400	0.622	0.278	58.04
SEEDS$_{256}$	0.961	0.014	0.960	0.007	0.012	0.395	0.647	0.297	58.97
Merged$_2$	0.898	0.515	0.883	0.042	0.063	0.553	0.472	0.333	<u>60.00</u>
Merged$_4$	0.898	0.496	0.883	0.042	0.062	0.548	0.484	0.340	**61.36**
Merged*	0.899	0.597	0.880	0.045	0.066	0.547	0.510	0.359	61.85
Oracle	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	70.81

Table 1: *Evaluation metrics of superpixels.* The subscript indicates the average size of the superpixel for SLIC [1] and SEEDS [35], while it indicates the round for Merged. Merged* indicates superpixel merged by a model trained with full supervision. To compute the mIoU, we train a model with 100k randomly selected superpixels.
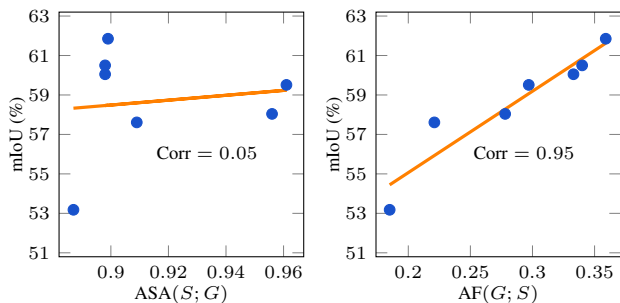

Figure 5: *Relationship between metrics and mIoU.* The correlation between ASA($S;G$) and mIoU is low, while the correlation between AF($G;S$) and mIoU is high. For the correlation calculation, *Oracle* in Table 1 is excluded.

where $s'(g;S) := \arg\max_{s \in S} |g \cap s|$ refers to the generated superpixel with the highest overlap, which is linked to the maximum amount of labeling we receive. Table 1 evaluates various superpixels through eight metrics. Although our merged superpixels have a relatively low ASA($S;G$), they exhibit high ASA($G;S$) and AF($G;S$).

Correlation of metrics and mIoU. To show the proposed metric can accurately evaluate the superpixel quality for active segmentation, we measure the correlation between various evaluation metric and the performance of actively learned model in Table 1 and Figure 5. We observe that the proposed AF($G;S$) shows the highest correlation to the performance of the actively learned model. We except AF($G;S$) can select suitable superpixel algorithm for active learning, where the details are provided in Appendix E.

5.2. Ablation studies on epsilon

Epsilon sensitivity. In Figures 6a and 6b, we evaluate the sensitivity of our method to ϵ, which determines the amount of the merging. Proposed method show robustness to the change of ϵ, where the change of mIoU is less than 2%

for both Citycapes and PASCAL when ϵ is between 0.05 and 0.2. We observe that for every investigated ϵ values, *AMSP+S* surpasses the performance of the previous art [4].

Adaptive epsilon. We fix ϵ to 0.10 in all quantitative experiments, but there may exist an optimal ϵ for each round. In Table 2, we analyze ϵ that maximizes AF($G;S$) metric for each round by assuming the existence of 10 validation images with ground truth. As the round increases, the optimal ϵ increases as well, which implies that the improvement of the model enables us to merge aggressively.

Effect of epsilon. Table 3 presents the quality of the merging algorithm under various criteria, defining correctness based on the agreement of dominant labels in paired superpixels. We merge a pair of superpixels when their ground-truth label is identical (Ground Truth), when their dominant top-1 model prediction is identical (Pseudo Label), and when the Euclidean Distance (ED) or Jensen-Shannon Divergence (JSD) of their averaged predictive probability is smaller than ϵ. Using pseudo labels leads to lower-quality merging as it ignores other minor classes. Since we utilize the predicted class probability as a feature space, JSD proves to be more effective than ED. As ϵ increases, the correct ratio decreases due to the aggressive merging of superpixels. We emphasize that ϵ can determine the trade-off between the quantity and quality of labels.

5.3. Implementation remarks for practitioners

Fast merging. The completion of the merging process for an image essentially requires a linear time complexity in the number of the base superpixels. However, to reduce this, the complete merging can be replaced with a *partial merging* that scans only a subset of base superpixels with high uncertainties as roots. This is considerable since we will eventually query only a subset of the merged superpixels according to the acquisition function (7), which prioritizes those with high uncertainties. In Table 4, we compare the complete and partial mergings in terms of the mIoU at 100k
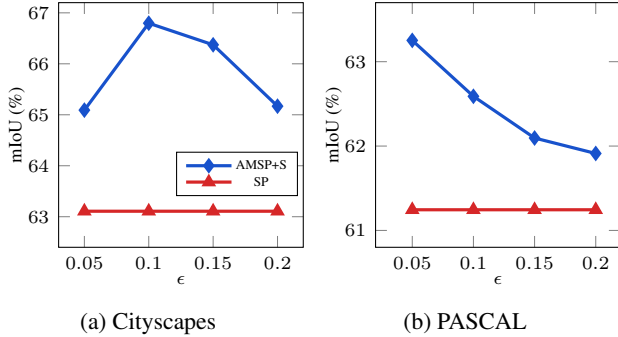
Figure 6: *Epsilon sensitivity.* We experiment on superpixels with varying ϵ and demonstrate the robustness of our *AMSP+S*, while *SP* is independent of ϵ. Each experiment is conducted on the second round.

Epsilon	0.04	0.05	0.06	0.07	0.08
Merged$_1$	0.344	**0.346**	0.344	0.340	0.336
Merged$_2$	0.347	0.346	**0.348**	0.345	0.344
Merged$_3$	0.346	0.349	0.350	**0.351**	0.349
Merged$_4$	0.347	0.347	0.347	**0.348**	0.346

Table 2: *Adaptive epsilon.* $AF(G; S)$ for adaptive superpixels generated by varying ϵ is reported. The subscript indicates the round.

Method	Epsilon	Correct	Incorrect
Ground Truth	-	1.000	0.000
Pseudo Label	-	0.832	0.168
ED	0.05	0.915	0.085
	0.10	0.901	0.099
	0.15	0.891	0.109
JSD	0.05	0.934	0.066
	0.10	0.911	0.089
	0.15	0.896	0.104

Table 3: *Various merging criteria.* Using JSD performs more accurate merging than using ED. As ϵ increases, the rate of incorrect merging increases.

Methods	mIoU
SP [4]	63.77
AMSP+S (top 10%)	65.99
AMSP+S (complete 100%)	**66.53**

Table 4: *Various levels of partial merging.* Experiments are conducted under the same setting of Figure 3a with 100k clicks (Cityscapes, superpixel size of 256).

clicks. As expected, the partial merging on base superpixels with top-10% uncertainty has only a small gap to the complete merging. In addition, we find that by employing the partial merging, the time complexity is reduced significantly by a factor of 25.98 [1]. The partial merging is a useful suggestion to save computation resource for practitioners. A further investigation and discussion on the partial merging are presented in Appendix B.

Compatibility with other base superpixels. For a fair comparison to the previous study [4], we have employed the same superpixel algorithm called SEEDS [35] to generate base superpixels at the beginning. However, our merging and sieving processes can be applied on top of any other base superpixels. Indeed, in Figure 3c (and Appendix A), our method consistently shows gains over SP [4] when using base superpixels of different sizes. Furthermore, we compare SP and AMSP+S (ours) when using SLIC instead of SEEDS in the same setting of Figure 3a, where the mIoU's at 100k clicks of SP and ours have 65.97% and 67.56%, respectively, *i.e.*, the merging and sieving are also effective with SLIC as they were with SEEDS. We believe that our proposed method can work with any base superpix-

---

[1]The per-image runtime of the merging process (CPU-intensive) for Cityscapes is reduced from 12.42s to 0.48s on a server with two AMD EPYC 7513 32-core processors.

els even when they are from an unsupervised segmentation method [18] or a foundation model [19].

6. Conclusion

In this work, we propose an adaptive active learning framework with adaptive superpixels. Our merging and sieving methods operate adaptively every round, and the experimental results demonstrate the performance improvement of adaptive merging in various realistic situations. Furthermore, we suggest novel achievable metrics for evaluating superpixels in advance that are suitable for active learning.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 2, 3, 8

[2] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11075–11083, 2019. 3

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4, 15

[4] Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10988–10997, 2021. 1, 2, 3, 4, 5, 6, 8, 9, 12, 13

[5] Arantxa Casanova, Pedro O Pinheiro, Negar Rostamzadeh, and Christopher J Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations*, 2019. 1, 2, 3

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 5

[7] Pascal Colling, Lutz Roese-Koerner, Hanno Gottschalk, and Matthias Rottmann. Metabox+: A new region based active learning method for semantic segmentation using priority maps. *arXiv preprint arXiv:2010.01884*, 2020. 3

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3, 5

[9] Chengliang Dai, Shuo Wang, Yuanhan Mo, Kaichen Zhou, Elsa Angelini, Yike Guo, and Wenjia Bai. Suggestive annotation of brain tumour images with gradient-guided sampling. *International Conference on Medical Image Computing and Computer Assisted Intervention, 2020*, 2020. 1

[10] Shasvat Desai and Debasmita Ghose. Active learning for improved semi-supervised semantic segmentation in satellite images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 553–563, 2022. 1

[11] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 1, 6

[12] 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*, 88:303–308, 2009. 7

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 5

[14] Sean Gillies et al. Shapely: manipulation and analysis of geometric objects, 2007. 15

[15] Rémi Giraud, Vinh-Thong Ta, and Nicolas Papadakis. Robust shape regularity criteria for superpixel evaluation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3455–3459. IEEE, 2017. 3, 4, 7

[16] S. Alireza Golestaneh and Kris M. Kitani. Importance of self-consistency in active learning for semantic segmentation. *British Machine Vision Conference (BMVC)*, 2020. 3

[17] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE, 2009. 4

[18] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2571–2581, 2022. 9

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 9

[20] Alex Levinshtein, Adrian Stere, Kiriakos N. Kutulakos, David J. Fleet, Sven J. Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2290–2297, 2009. 3

[21] Shuailin Li, Zhitong Gao, and Xuming He. Superpixel-guided iterative learning from noisy labels for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, pages 525–535. Springer, 2021. 3

[22] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097–2104, 2011. 2, 3, 7

[23] 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. 3

[24] Vaïa Machairas, Etienne Decencière, and Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4343–4347. IEEE, 2014. 3, 4, 7

[25] Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. Cereals-cost-effective region-based active learning for semantic segmentation. In *BMVC*, 2018. 3

[26] Youngmin Oh, Beomjun Kim, and Bumsub Ham. Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6913–6922, 2021. 3

[27] Yu Qiao, Jincheng Zhu, Chengjiang Long, Zeyao Zhang, Yuxin Wang, Zhenjun Du, and Xin Yang. Cpral: Collaborative panoptic-regional active learning for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2108–2116, 2022. 1, 2

[28] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, IEEE International Conference on*, volume 2, pages 10–10. IEEE Computer Society, 2003. 3

[29] Gerard Salton. Introduction to modern information retrieval. *McGraw-Hill*, 1983. 7

[30] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011. 3, 5, 13, 14

[31] Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. Measuring and evaluating the compactness of superpixels. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 930–934. IEEE, 2012. 3, 4, 7

[32] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 888–905, 2000. 2, 3

[33] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. Viewal: Active learning with viewpoint entropy for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9433–9443, 2020. 1, 2, 3

[34] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019. 1, 3

[35] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 13–26. Springer, 2012. 2, 3, 7, 8, 9, 17

[36] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, page 211–224, 2010. 3

[37] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, and Xinjing Cheng. Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8068–8078, 2022. 1, 3

[38] Shuai Xie, Zunlei Feng, Ying Chen, Songtao Sun, Chao Ma, and Mingli Song. Deal: Difficulty-aware active learning for semantic segmentation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2020. 1

[39] Longrong Yang, Fanman Meng, Hongliang Li, Qingbo Wu, and Qishang Cheng. Learning with noisy class labels for instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 38–53. Springer, 2020. 3

[40] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017. 1, 3

[41] Jiachen Yao, Yikai Zhang, Songzhu Zheng, Mayank Goswami, Prateek Prasanna, and Chao Chen. Learning to segment from noisy annotations: A spatial correction approach. In *International Conference on Learning Representations*, 2023. 3

[42] Yuhang Zhang, Richard Hartley, John Mashford, and Stewart Burn. Superpixels via pseudo-boolean optimization. In *2011 International Conference on Computer Vision*, pages 1387–1394, 2011. 3

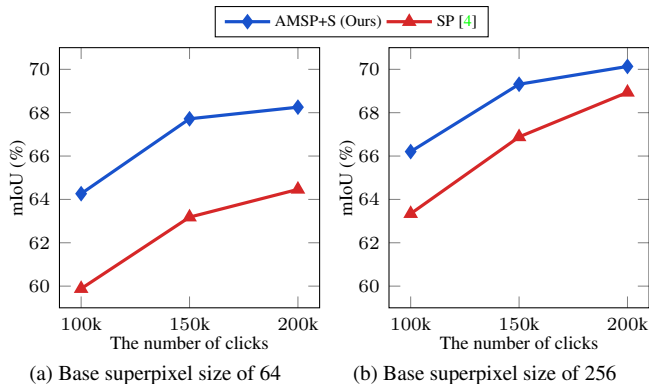# Appendix

A. More gain with other base superpixel sizes

(a) Base superpixel size of 64

(b) Base superpixel size of 256

Figure 7: *Effect of base superpixel size on Cityscapes.* The performance difference is greater when the superpixel size is smaller.
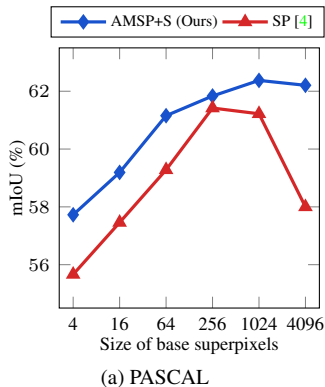

(a) PASCAL

Figure 8: *Effect of base superpixel size on PASCAL.* Our method exhibits robustness to large superpixels, while the baseline is sensitive.

For ease of exposition, Figure 3 presents the gain of our method (compared to *SP* [4]) for a limited set of base superpixel sizes. In this section, we report an additional investigation suggesting further gain with different base superpixels.

Further gain on Cityscapes. In Figure 7a, we additionally provide a comparison between the proposed method (*AMSP+S*) and *SP* [4], where the experiment setup with Cityscapes is identical to that in Figure 3a except that the base superpixel size is 64 (Figure 7a) instead of 256 (Figure 7b). Our adaptive merging method (*AMSP+S*) is especially effective when the superpixel size is small in Figure 7a, thanks to the adaptive merging mechanism. This observation suggests more significant gain of our method with other choices of base superpixel size than that in Figure 3.
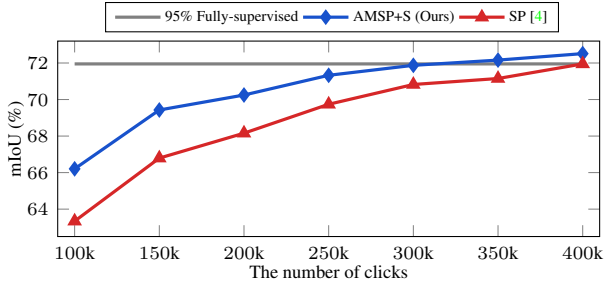

Figure 9: *Additional rounds experiments on Cityscapes.* We extend the experiments in Figure 3a up to a budget of 400k. The performance improvement remains consistent across various additional budgets.

| Methods	mIoU
SP [4]	63.77
AMSP+S (bottom 10%)	64.33
AMSP+S (top 10%)	65.99
AMSP+S (complete 100%)	**66.53**

Table 5: *Various levels of partial merging.* Experiments are conducted under the same setting of Figure 3a with 100k clicks (Cityscapes, superpixel size of 256).

Further gain on PASCAL. We also demonstrate a larger gap between the proposed method and existing one in PASCAL. In Figure 8, our adaptive merging method (*AMSP+S*) outperforms the baseline (*SP*) for various superpixel sizes as we observed in Figure 3. We stress that the gain of the proposed method is particularly larger than the one reported in Figure 3 when the base superpixel size is 4096, which is much larger than 256 used in Figure 3. This is because the sieving procedure to suppresses the noise from dominant labeling becomes more crucial when querying large superpixels. The experimental setup used in Figure 8 is identical to that of Figure 3d.

Further rounds on Cityscapes. To demonstrate the efficacy of our method across various budgets, we experiment by gradually increasing the budget as illustrated in Figure 9 on Cityscapes. The experimental setting in Figure 9 remains consistent with that of Figure 3a. The advantage of our method over SP [4] is continued in further rounds. We remark that the proposed method nearly achieves the 95% mIoU of the fully supervised model (71.95%) at 300k clicks, whereas SP does at 400k clicks.

B. Rationale for line 3 of Algorithm 2

We explain the rationale behind traversing nodes in the descending order of uncertainty in line 3 of Algorithm 2.
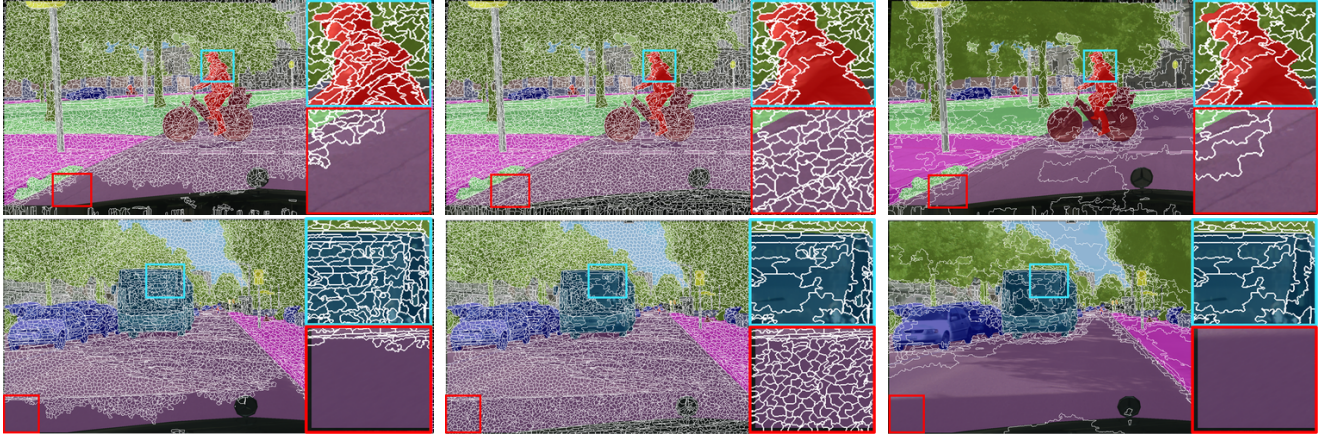
(a) Merging superpixels with low 10% uncertainty (b) Merging superpixels with high 10% uncertainty   (c) Merging all superpixels

Figure 10: *Qualitative results for partial merging.* The cyan boxes encompass superpixels exhibiting the highest 10% uncertainty, while the red boxes encompass superpixels with the lowest 10% uncertainty. (b) By merging only a portion of superpixels in the order of high uncertainty, we can reduce time complexity, as it creates similar merged superpixels compared with the cyan box in (c).

Methods	mIoU
SP [4]	63.77
AMSP+S ($\phi(s; \theta) = 0.0$)	65.35
AMSP+S ($\phi(s; \theta) = 0.2$)	61.80
AMSP+S ($\phi(s; \theta) = 0.4$)	57.77
AMSP+S ($\phi(s; \theta) = 0.6$)	45.84
AMSP+S ($\phi(s; \theta) = 0.8$)	38.99
AMSP+S (Kneedle [30])	**66.53**

Table 6: *Various sieving methods.* Experiments are conducted on Cityscapes dataset with an average superpixel size of 256, using 100k costs for two rounds.
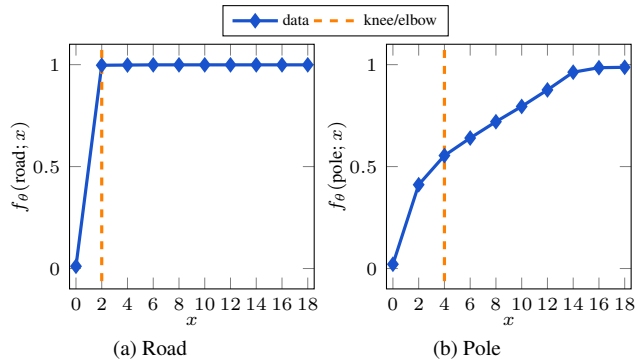

(a) Road   (b) Pole

Figure 11: *Examples of knee points on Cityscapes.* We obtain (a) a high knee value for the common road class and (b) a low knee value for the rare pole class.

Our merging process requires a linear time complexity proportional to the size of the base superpixels graph. However, due to the advantage of merging in descending uncertainty order, we are able to acquire merged superpixels with considerable uncertainty at the beginning of merging. To reduce merging time complexity, we only merge the top 10% of base superpixels with the highest uncertainty as query candidates. Table 5 shows that it is important to prioritize the merging highly uncertain superpixels, and merging along the ascending order of uncertainty degenerates the performance.

In Figure 10, we exemplify the merged superpixels from the partial merging in the ascending or descending order of uncertainty, and the full merging, where the cyan boxes contain higher values of acquisition function than the red boxes. The partial merging with the ascending order of uncertainty regrettably merges the superpixels that would not be selected in AL, while that with the ascending order efficiently combines the base superpixels of which selection is highly like. This difference indeed results in a huge gap in the final performance as shown in Table 5.

C. Rationale for the adaptive threshold $\phi(s; \theta)$ in the sieving

We provide the reason for introducing the threshold function $\phi(s)$ personalized for each superpixel s, described in Section 3.3. We obtain the dominant label $D(s)$ for a queried superpixel s, however, we only propagate the label to pixels $x \in s$ that are predicted to have a positive impact
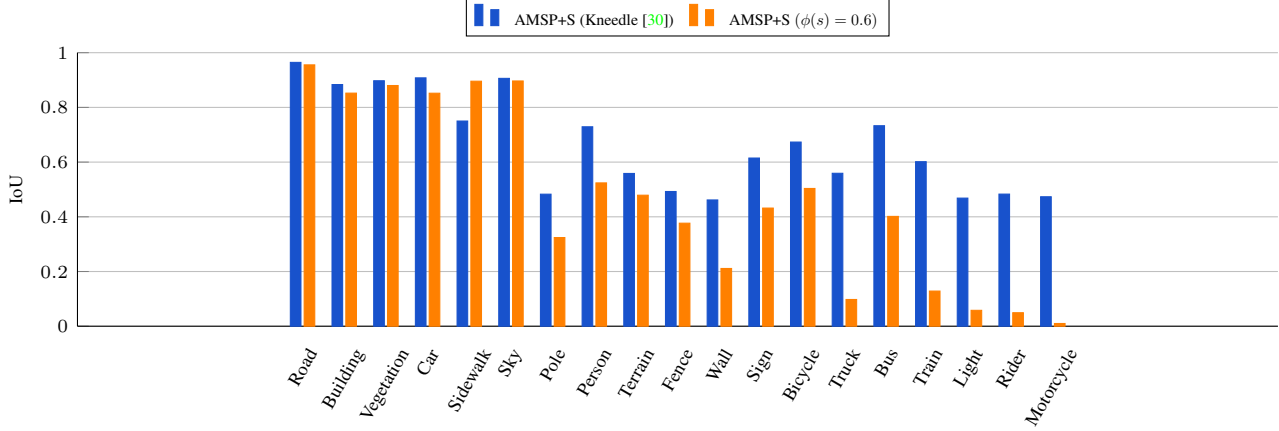
Figure 12: *Class-wise IoU according to $\phi(s;\theta)$.* Applying the same $\phi(s)$ of 0.6 to all pixels results in excessive sieving for relatively rare classes, leading to decreased performance for these classes (*e.g.* Light, Rider, and Motorcycle). Based on the ground-truth, class labels are organized in order of the total pixel count for each class.

(a) Semantic segmentation

(b) Panoptic segmentation
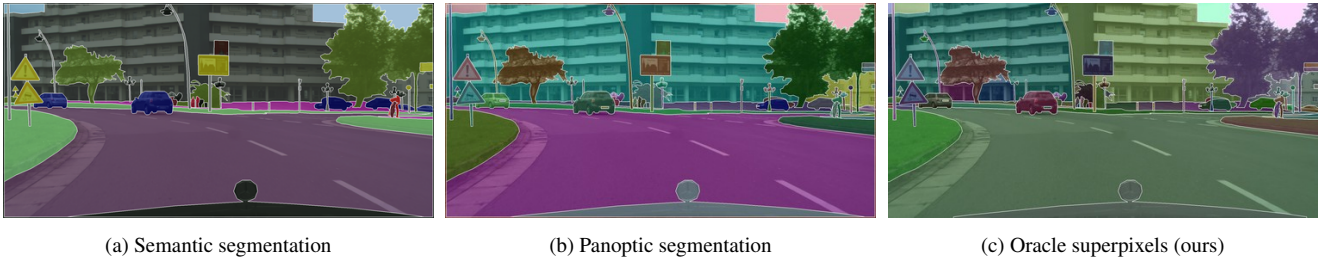
(c) Oracle superpixels (ours)

Figure 13: *Difference between conventional segmentations and oracle superpixels.* (a) When sharing the same class label, they are depicted as identical superpixels (*i.e.* green color on separate trees). (b) Although a building is divided by a pole, it is represented as a single superpixel (*i.e.* cyan color). (c) We consider a building as two distinct superpixels (*i.e.* cyan and light yellow colors).

on the training of model θ as:

$$h(s;\theta) := \{x \in s : f_\theta\big(\mathrm{D}(s);x\big) \geq \phi(s;\theta)\}, \quad (16)$$

where $f_\theta(\mathrm{D}(s);x)$ implies the confidence of pixel x to dominant label $\mathrm{D}(s)$ given θ and $\phi(s;\theta)$ determines the degree of sieving. In Table 6, we study the effect of various $\phi(s;\theta)$. When the same $\phi(s;\theta)$ is applied to all pixels, it causes class imbalance by leaving relatively easy classes as described in Figure 12. To avoid this issue, we utilize the Kneedle algorithm [30] to obtain different $\phi(s;\theta)$ for each superpixel s. Specifically, $\phi(s;\theta)$ is a knee point of the cumulative distribution function of values of $f_\theta(\mathrm{D}(s);x)$ in superpixel $x \in s$. However, for the Kneedle algorithm to work accurately, the curve of cumulative distribution must be either convex or concave. In addition, the algorithm may provide inaccurate knee points on very smooth curves. To address this issue, we use a subset of uniformly sampled values based on $f_\theta(\mathrm{D}(s);x)$, instead of using the distribution for all pixels. We sample 20 and 5 pixels for Cityscapes and PASCAL datasets, respectively. In Figure 11, different knee points are detected according to the dominant class of superpixels.

Effect of sieving. Our sieving method exhibits a significant effect on larger superpixels, as illustrated in Figure 3c and Figure 8. Especially, in Figure 8 with a large base superpixel size of 4096, the first sieving excises 45.87% of the mislabeled pixels that disagree with their dominant labels. Furthermore, we observe that the sieving is progressively refined round by round. For instance, in Figure 3a, the portion of the mislabeled labels removed by the sieving increases over four rounds as follows: 3.58%, 8.54%, 10.46%, and 12.43%. Our sieving technique enhances label quality by retaining only high-confidence labels and continuously improves through multiple rounds.

D. Further discussion on the oracle superpixels

In Section 4.1, we introduce the oracle superpixels, which we believe is an achievable optimal set of superpixels for active learning. For clarification, we provide the detailed
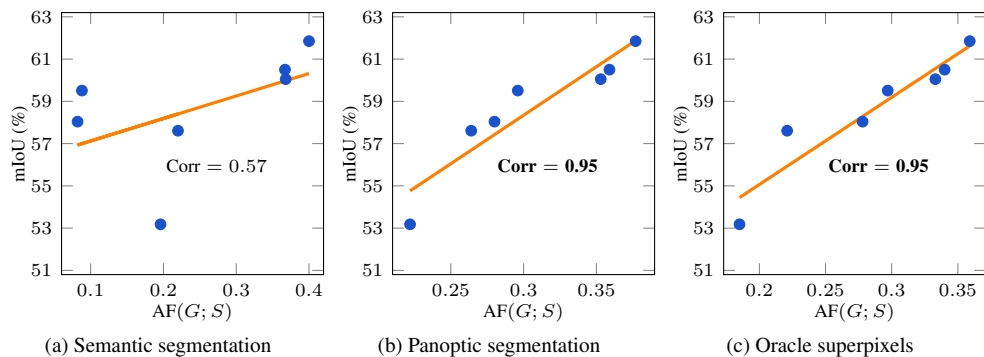
(a) Semantic segmentation   (b) Panoptic segmentation   (c) Oracle superpixels

Figure 14: *Relationship between $AF(G; S)$ and mIoU varying G.* $AF(G; S)$ and mIoU exhibit a high correlation when ground-truth G is represented by the panoptic segmentation and oracle superpixels in Figure 13. For the correlation calculation, *Oracle* in Table 1 is excluded.
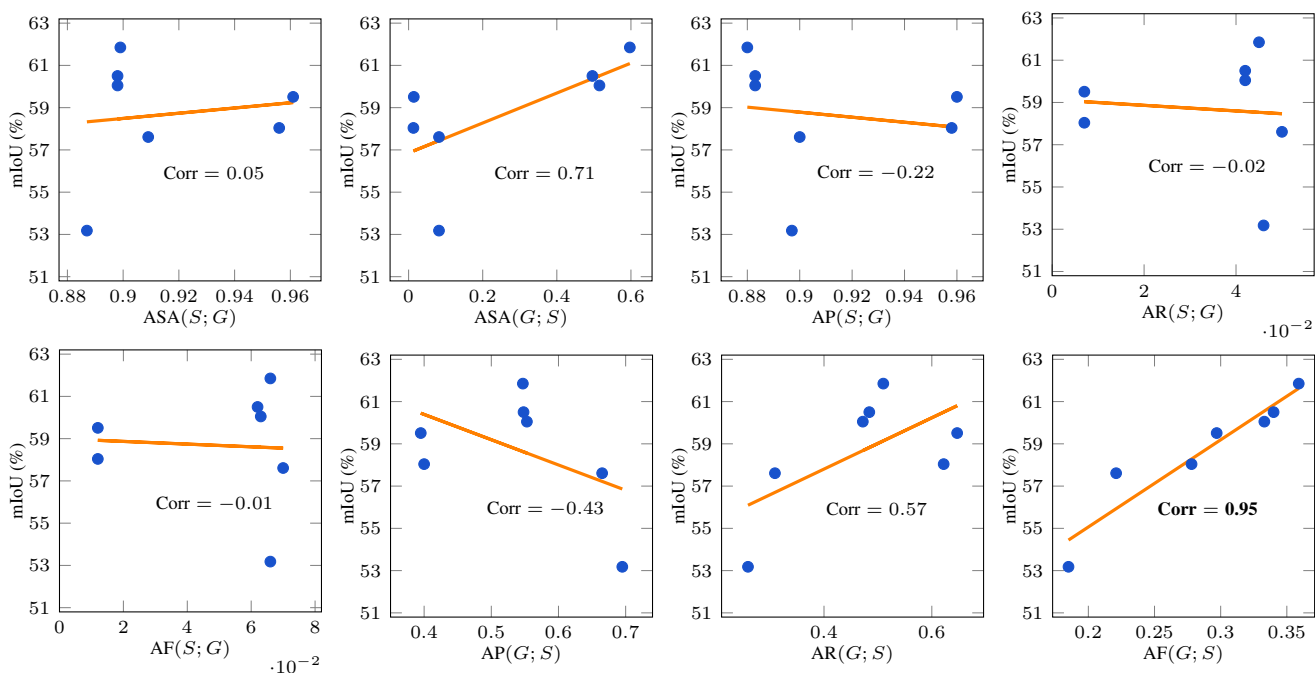

Figure 15: *Relationship between metrics and mIoU.* The correlation between $AF(G; S)$ and mIoU is especially high. For the correlation calculation, *Oracle* in Table 1 is excluded.

process of generating the proposed oracle superpixels. In addition, we provide further insights into the achievable notion of optimal superpixels.

The Cityscapes dataset is equipped with the ground-truth annotations for semantic segmentation, represented by dense pixel-wise labels: *i.e.*, each pixel in an annotated image is assigned an ID that represents a ground-truth semantic category (Figure 13a). In such annotation, each group of pixels that share the same ID aligns perfectly with the boundary of semantic objects. However, each such group is not guaranteed to be a single-connected component of pixels. For example, different cars in Figure 13a are assigned

the same blue color despite being physically separated, and a car divided into two parts due to an obstructing pole is still colored blue. This is opposed to what we hope to achieve by merging two adjacent superpixels repeatedly. To address this issue, we subdivide each superpixel as necessary to ensure that every pixel within a superpixel is adjacent to each other. We utilize OpenCV [3] and Shapely [14] to identify the maximal connected component of pixels sharing the same semantic. We apply the same procedure to annotated images in the PASCAL dataset Figure 13 illustrates the distinction between conventional semantic and our oracle superpixels.

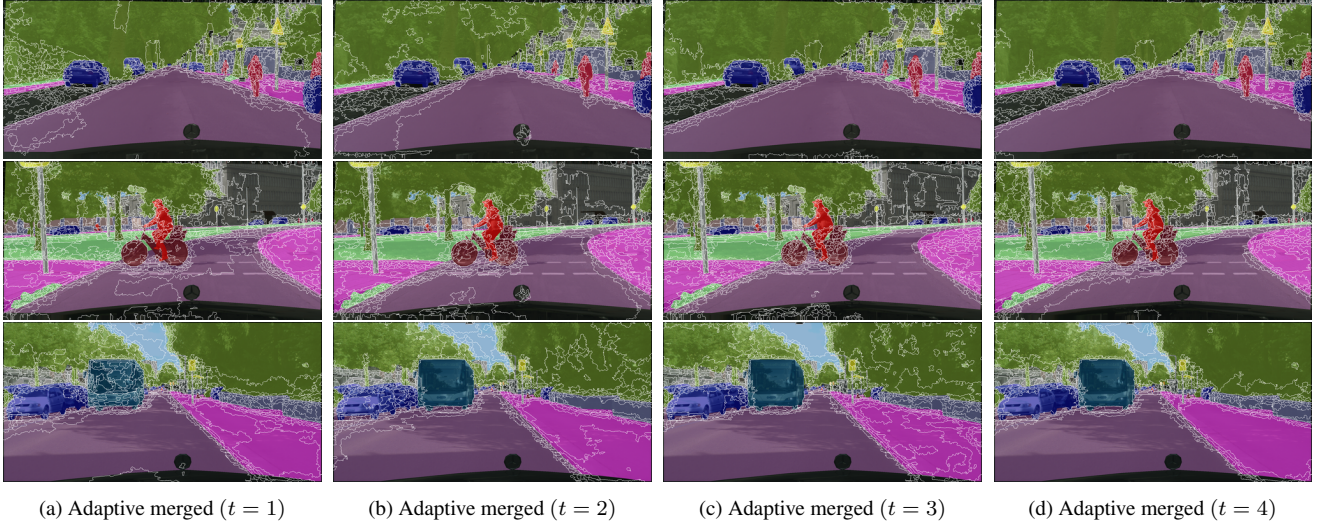| (a) Adaptive merged ($t = 1$) | (b) Adaptive merged ($t = 2$) | (c) Adaptive merged ($t = 3$) | (d) Adaptive merged ($t = 4$) |

Figure 16: *Qualitative results with varying round.* (a-d) Superpixels generated with proposed adaptive merging at rounds 1 to 4. Thanks to the improved model, we observe that the merging becomes more accurate as the round increases. We use the model reported in Figure 3a.
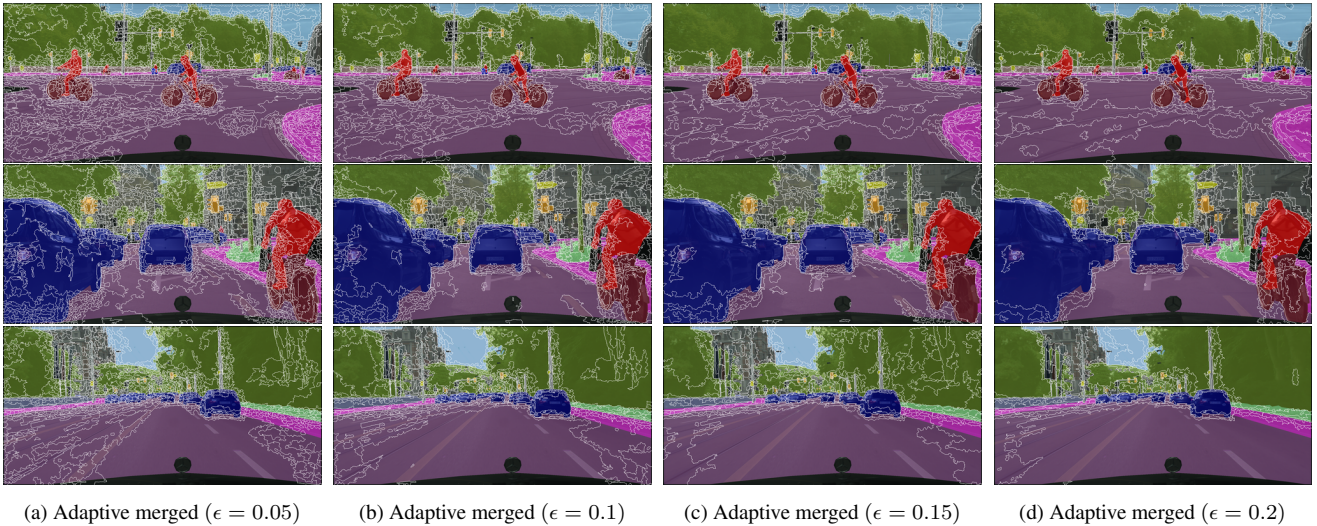

| (a) Adaptive merged ($\epsilon = 0.05$) | (b) Adaptive merged ($\epsilon = 0.1$) | (c) Adaptive merged ($\epsilon = 0.15$) | (d) Adaptive merged ($\epsilon = 0.2$) |

Figure 17: *Qualitative results with varying ϵ.* (a-d) Superpixels are generated with proposed adaptive merging with ϵ: 0.05, 0.1, 0.15, 0.2. We observe that an increase in ϵ gives more aggressive merging. Merging is conducted on Cityscapes with a base superpixel size of 256.

The Cityscapes and PASCAL datasets are divided into 327k and 16k oracle superpixels, respectively. It is worth noting that the PASCAL has a lower number of oracle superpixels due to the smaller number of classes per image. In other words, only a few objects are of interest in each image, and the rest are simply treated as the background.

E. Further analysis on the achievable metrics

In Table 1, we evaluate various superpixels using eight metrics with oracle superpixels as ground-truth G. Figure

15 shows the correlation between each metric and mIoU. We observe that our $\text{AF}(G; S)$ can be utilized to look-ahead a model's performance in active learning without training. In addition, we examine how different ground-truth G impacts $\text{AF}(G; S)$. In the field of semantic segmentation, two conventional segmentations, semantic and panoptic segmentations in Figure 13, are widely used as ground-truth. Figure 14 indicates that using panoptic segmentation and oracle superpixels for G results in higher correlation between $\text{AF}(G; S)$ and mIoU than semantic segmentation. However,

(a) Base superpixels [35] (b) Merged superpixels (Ours) (c) Oracle superpixels
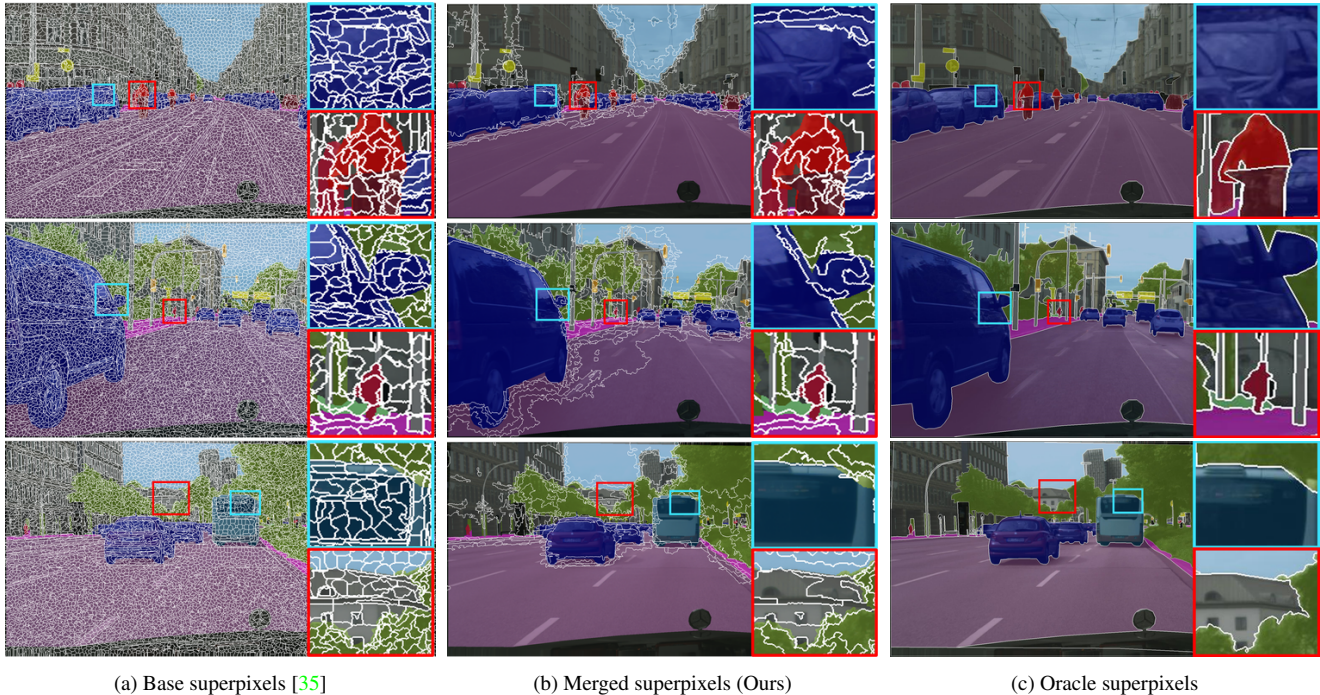
Figure 18: *Qualitative results of adaptive superpixels.* (a) Base superpixel generated by SEEDS [35] with size 256. (b) Superpixels generated with proposed adaptive merging at round 4. (c) Oracle superpixels generated from the ground truth.
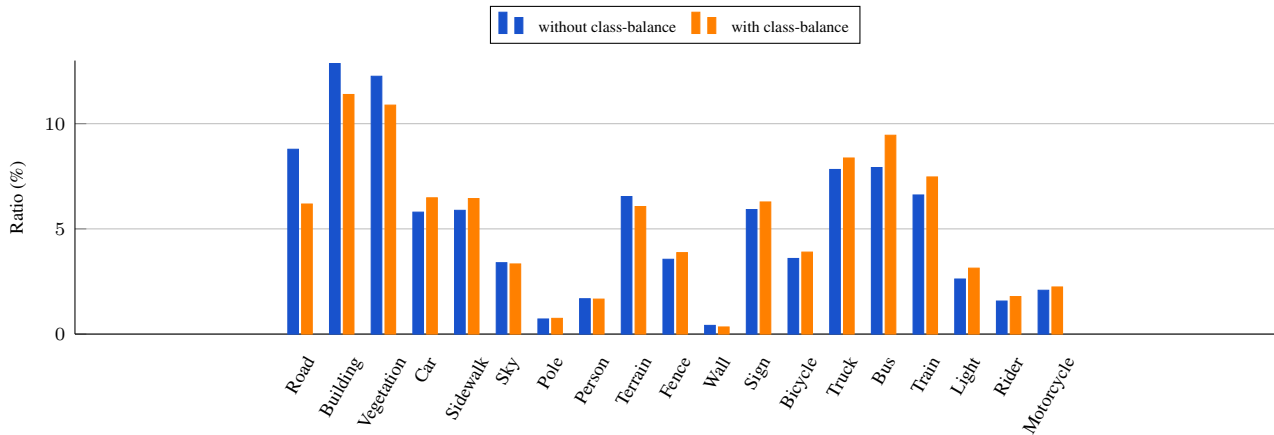

Figure 19: *Effect of class-balanced acquisition function.* According to the ground-truth, class labels are arranged based on the total pixel count for each class, *i.e.* classes become rarer in images as you move from left to right along the x-axis. We observe that classes on the left are selected less with the class-balanced term, while classes on the right are selected more.

obtaining panoptic segmentation requires more costs than semantic segmentation since it utilizes additional instance information. It is worth noting that our oracle superpixels (Figure 13c) can be easily generated even in cost-limited practical situations as they are produced from semantic segmentation (Figure 13a).

F. Additional qualitative adaptive superpixels

To facilitate comprehension of the merged superpixels, we display superpixels generated across diverse settings. The appearance of merged superpixels is mainly determined by the model's performance and ϵ. Figure 16 highlights that as the round progresses, the model's performance improves, leading to more accurate merging. With the model fixed at round 4, Figure 17 shows the impact of adjusting ϵ. As ϵ

Notations	Description
$\mathcal{I}$	the set of unlabeled images
$\mathcal{C}$	the set of class labels
t	a round
x	a pixel
s	a superpixel
$S_t(i)$	the set of superpixels in an image i in round t
$\mathcal{S}_t$	the set of superpixels in all images in round t, $\mathcal{S}_t := \bigcup_{i \in \mathcal{I}} S_t(i)$
B	the query budget per round
$\mathcal{B}_t$	the set of B selected superpixels in round t, $B_t \subset \mathcal{S}_t, |B_t| = B$
θ_t	the model at the end of round t
$y_\theta(x)$	the estimated dominant label of pixel x given θ
$\mathrm{D}(s)$	the true dominant label of superpixel s
$\mathrm{D}_\theta(s)$	the estimated dominant label of superpixel s given θ
$\mathcal{G}(S) := (S, \mathcal{E}(S))$	the graph consisting of the superpixels in S as nodes and the edge set $\mathcal{E}(S)$ such that $(s, n) \in \mathcal{E}(S)$ for each pair of adjacent superpixels $s, n \in S$.
ϵ	the hyperparameter for merging in (2)

Table 7: *Notations.* The notations used in the paper are defined.

grows, the merging process intensifies, ultimately decreasing the overall number of superpixels. In addition, Figure 18 shows further examples of our merged superpixels.

G. Class-balanced sampling

To observe the impact of the class-balanced acquisition function in (7), we analyze the class distribution of selected superpixels both with and without the class-balanced term. In Figure 19, where class labels are sorted such that the left (road) and right (motorcycle) ends represent the most and least popular classes, it is evident that the class-balanced term results in a higher selection of rarer classes, as intended.