Abstract
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.

1. Introduction
Active learning (AL) offers an approach to alleviate the annotation cost by selectively querying only the most informative samples to annotators. 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.
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.
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).

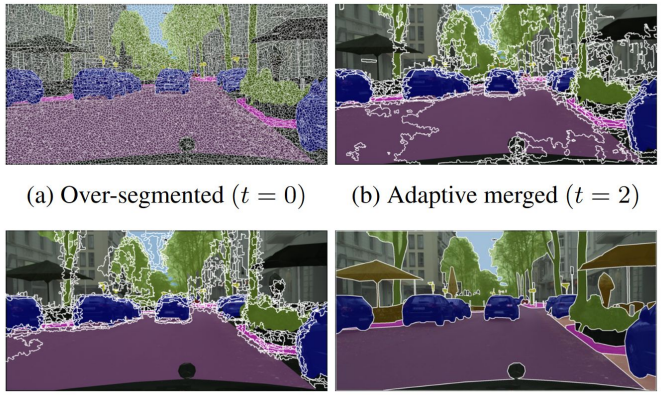

Figure 1: *Examples of adaptive superpixels.* (a) We begin active learning with over-segmented superpixels. (b, c) In each round t, we merge superpixels in an adaptive manner using the model from the previous round. (d) As the round progresses, adaptive superpixels look similar to oracle ones.

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.
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.

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.
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.
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.
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.

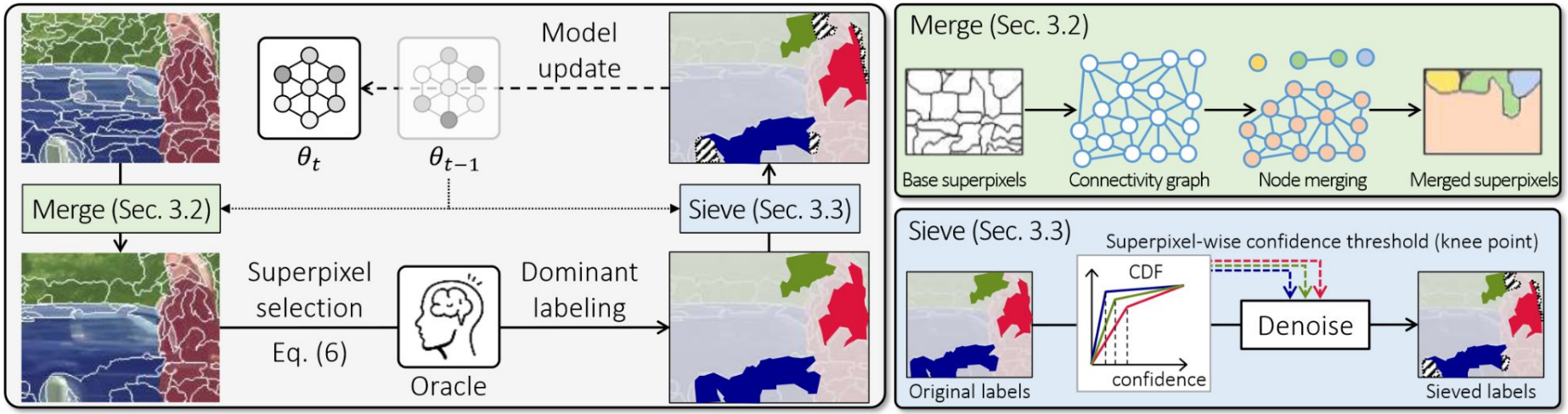

Figure 2: *An overview of the proposed framework.* In each round t, we merge superpixels with a graph using the latest model, and obtain dominant labels for selected superpixels. The dominant labels are selectively propagated to pixels with confidence above the detected knee point, resulting in the creation of a sieved dataset. Finally, we train a model with the sieved one.

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 clus-

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.

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** $t = 1, 2, \dots, T$ **do**
4: Adaptively merge the base superpixels and obtain $\mathcal{S}_t \leftarrow \bigcup_{i \in \mathcal{I}} \mathrm{AM}(S_0(i), \theta_{t-1})$
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

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

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 $\mathcal{S}_t \leftarrow \bigcup_{i\in\mathcal{I}} \mathrm{AM}(S_0(i), \theta_{t-1})$
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

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

Algorithm 1 Proposed Framework

Require: Image set $\mathcal{I}$, batch size B, and final round T.

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

Algorithm 2 Adaptive Merging (AM)

Require: Base superpixels S, model θ, and threshold ϵ.

Set $S' \leftarrow \emptyset$ and $\mathcal{G}(S) \leftarrow (S, \mathcal{E}(S))$
Mark s as unexplored for each $s \in S$
for $s \in S$ in descending order of $u_\theta(s)$ **do**
    if s is unexplored **then**
        $S' \leftarrow S' \cup \{\textsc{Merge}(s, f_\theta(s); \mathcal{G}, \theta)\}$
return S'
procedure $\textsc{Merge}(s, f; \mathcal{G}, \theta)$
    Mark s as explored and set $s' \leftarrow s$
    for each neighbor n of s in $\mathcal{G}$ **do**
        if n is unexplored and $d_{\mathrm{JS}}(f \,\|\, f_\theta(n)) < \epsilon$ **then**
            $s' \leftarrow s' \cup \textsc{Merge}(n, f; \mathcal{G}, \theta)$
    return s'

Adaptive merging.

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

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

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,

Algorithm 2 Adaptive Merging (AM)

Require: Base superpixels S, model θ, and threshold ϵ.

```
Set S' ← ∅ and G(S) ← (S, E(S))
Mark s as unexplored for each s ∈ S
for s ∈ S in descending order of u_θ(s) do
    if s is unexplored then
        S' ← S' ∪ {MERGE(s, f_θ(s); G, θ)}
return S'
procedure MERGE(s, f; G, θ)
    Mark s as explored and set s' ← s
    for each neighbor n of s in G do
        if n is unexplored and d_JS(f ‖ f_θ(n)) < ε then
            s' ← s' ∪ MERGE(n, f; G, θ)
    return s'
```

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

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.

Acquisition function.

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).

(6) 식 관련하여,
Spx 논문은 클래스 인기도를 슈퍼픽셀 수준에서 계산하였지만 $\frac{|\{s:\mathrm{D}_\theta(s)=c, s\in\mathcal{S}_t\}|}{|\{s:s\in\mathcal{S}_t\}|}$,

Ours(MerSpx)는 클래스 인기도를 픽셀 수준에서 계산하였다.

$$u_\theta(x) := \frac{\max_{c\in\mathcal{C}\setminus\{y_\theta(x)\}} f_\theta(c;x)}{\max_{c\in\mathcal{C}} f_\theta(c;x)}, \quad (4)$$: 픽셀 x의 uncertainty : 픽셀 x의 bvsb

$$u_\theta(s) := \frac{\sum_{x\in s} u_\theta(x)}{|\{x : x\in s\}|}, \quad (5)$$: 슈퍼픽셀 s의 uncertainty

$$y_\theta(x) := \arg\max_{c\in\mathcal{C}} f_\theta(c;x)$$: 픽셀 x의 estimated dominant label in a given model θ

$$p(c;\theta) := \frac{|\{x : \exists s\in\mathcal{S}_t, \mathrm{D}_\theta(s)=c, x\in s\}|}{|\{x : \exists s\in\mathcal{S}_t, x\in s\}|}, \quad (6)$$: 모델의 클래스 c 예측 빈도

$$\mathrm{D}_\theta(s) := \arg\max_{c\in\mathcal{C}} |\{x\in s : y_\theta(x)=c\}|$$: majority of predicted labels in superpixel s

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:

$$a(s;\theta) := u_\theta(s) \exp\left(-p(\mathrm{D}_\theta(s);\theta)\right) . \quad (7)$$

: acquisition function

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

tain $\mathcal{S}_t$ 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

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

: pixels to be sieved

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

: sieved dataset

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

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 $\mathcal{S}_t \leftarrow \bigcup_{i\in\mathcal{I}} \mathrm{AM}(S_0(i), \theta_{t-1})$
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