Abstract.
We hence propose an effective framework of active label correction (ALC) based on a design of correction query to rectify pseudo labels of pixels.

Specifically, leveraging foundation models providing useful zero-shot predictions on pseudo labels and superpixels, our method comprises two key techniques: (i) an annotator-friendly design of correction query with the pseudo labels, and (ii) an acquisition function looking ahead label expansions based on the superpixels.

Experimental results on PASCAL, Cityscapes, and Kvasir-SEG datasets demonstrate the effectiveness of our ALC framework, outperforming prior methods for active semantic segmentation and label correction.
We obtained a revised dataset of PASCAL by rectifying errors in 2.6 million pixels in PASCAL dataset.

1. Introduction.
We propose an ALC framework which leverages foundation models and correction queries. Our correction query is designed to rectify the pseudo labels of pixels, only if these pseudo labels are incorrect. Unlike the standard classification query that directly requests a specific class (Cai et al., 2021; Kim et al., 2023a), our correction query allows annotators to skip labeling if the pseudo labels are correct, making it more annotator-friendly.

Specifically, we leverage useful zero-shot predictions on pseudo labels and superpixels from foundation models.

- Contribution.
1. We provide theoretical and empirical justifications on the efficacy of the correction query, compared to the classification query (Section 3.2 and 4.2).
2. We propose an active label correction framework, leveraging the correction query and foundation models, where the look-ahead acquisition function enables selecting informative and diverse pixels to be corrected (Section 3.3 and 3.4).
3. To achieve comparable performance with SOTA active semantic segmentation methods, we only use 33% to 50% of budgets on various datasets (Section 4.2).
4. Using the proposed framework, we correct 2.6 million pixel labels in PASCAL and provide a revised version, called PASCAL+ (Section 5.2).

2. Related Work

Active Learning for Segmentation.

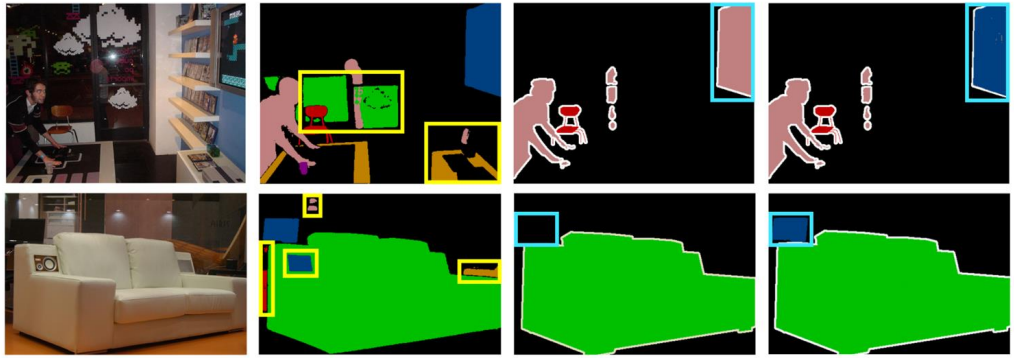
While conventional AL methods collect labels from scratch, the proposed method starts from the initial pseudo labels from foundation models, correcting erroneous labels.

Noisy Label Detection.

Our work is the first ALC method for semantic segmentation, correcting pixel labels and expanding them to their corresponding superpixels.

Efficient Query Design.

By employing the initial pseudo labels from foundation models, we suggest correction queries that only request the correct label when the given pseudo label is incorrect.



(a) Unlabeled image (b) Grounded-SAM (c) PASCAL (d) PASCAL+ (ours)

Figure 1: *Examples of noisy and corrected labels in PASCAL.* (a, b) Initial pseudo labels are generated by applying Grounded-SAM (G-SAM) to unlabeled images. As depicted by the yellow boxes, noisy pseudo labels result in a decline in performance, as shown in Table 7. (c) PASCAL also contains noisy labels in cyan boxes. (d) By employing the superpixels from G-SAM, we construct a corrected version of PASCAL, called PASCAL+. For instance, in the first row, we correct the object labeled as person to tvmonitor, and in the second row, the object labeled as background to tvmonitor. Here, the colors black, blue, red, green, and pink represent the background, tvmonitor, chair, sofa, and person classes, respectively.

3. Active Label Correction Framework

initial noisy dataset $\mathcal{D}_0$

Each query to an oracle annotator requests the accurate label $y \in \mathcal{C} := \{1, 2, ..., C\}$ for an associated pixel x.

In contrast to active learning (AL), which commences with an unlabeled image set, ALC focuses on progressively refining a labeled dataset $\mathcal{D}_0$ which may include noisy labels.

For each round t, we issue a batch $\mathcal{B}_t$ of B queries from a pixel pool $\mathcal{X}_t$ and train a model θ_t with the corrected annotations obtained so far.

In the following, we first prepare an initial dataset for correction (Section 3.1). After that, we present a correction query that requests for rectifying pseudo labels of pixels (Section 3.2). To fully enjoy the corrections, we introduce a look-ahead acquisition function, which selects from a diversified pixel pool (Section 3.3), considering the effect of label expansion (Section 3.4). The overall procedure is summarized in Algorithm 1.

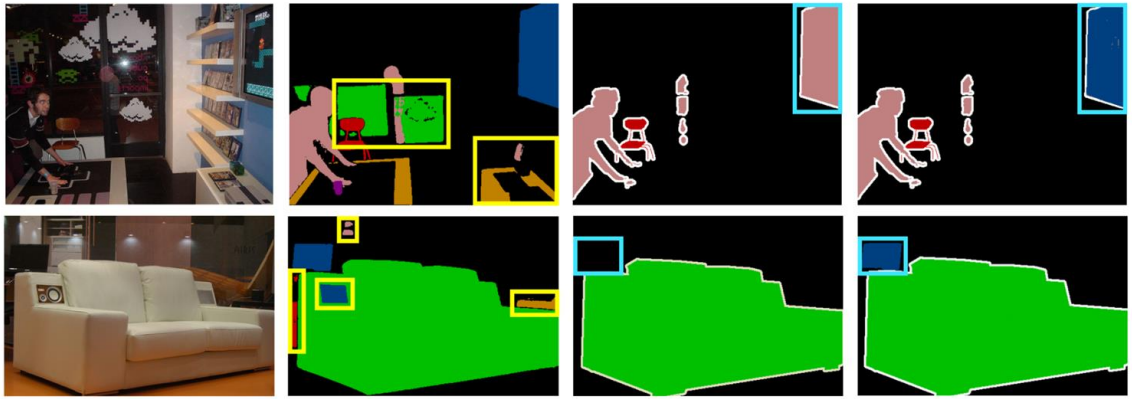
Algorithm 1 Proposed Framework

Require: Batch size B, and final round T.
1: Prepare initial dataset $\mathcal{D}_0$ requiring label correction
2: Obtain model θ_0 training with $\mathcal{D}_0$ via (1)
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Construct diversified pixel pool $\mathcal{X}_t^d$ via (4)
5: Correct labels of selected B pixels $\mathcal{B}_t \subset \mathcal{X}_t^d$ via (9)
6: Expand corrected labels to corresponding superpixels
7: Obtain model θ_t training with corrected $\mathcal{D}_t$ via (11)
8: **end for**
9: **return** $\mathcal{D}_T$ and θ_T

3.1. Initial Dataset Preparation

AL typically builds datasets through random pixel (Shin et al., 2021) or superpixel labeling (Cai et al., 2021) leading to lots of budgets and rounds, as it starts from unlabeled images, commonly known as the cold-start problem (Mahmood et al., 2021). Away from conventional AL methods, we utilize recent foundation models to construct segmentation datasets.

Grounded-SAM = Grounding DINO + SAM
Figures 1a and 1b display examples of the unlabeled images in PASCAL and corresponding initial pseudo labels generated by Grounded-SAM.



(a) Unlabeled image (b) Grounded-SAM (c) PASCAL (d) PASCAL+ (ours)

Figure 1: *Examples of noisy and corrected labels in PASCAL.* (a, b) Initial pseudo labels are generated by applying Grounded-SAM (G-SAM) to unlabeled images. As depicted by the yellow boxes, noisy pseudo labels result in a decline in performance, as shown in Table 7. (c) PASCAL also contains noisy labels in cyan boxes. (d) By employing the superpixels from G-SAM, we construct a corrected version of PASCAL, called PASCAL+. For instance, in the first row, we correct the object labeled as person to tvmonitor, and in the second row, the object labeled as background to tvmonitor. Here, the colors black, blue, red, green, and pink represent the background, tvmonitor, chair, sofa, and person classes, respectively.

Warm-start.
In contrast to the cold-start problem in AL, our ALC benefits from warm-start thanks to the initial labels provided by foundation models.

To obtain θ_0 *, we initialize* θ *to a model pre-trained on ImageNet (Deng et al., 2009).*

We then train it to reduce the following cross-entropy (CE) loss.

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

$f_\theta(x) \in \mathbb{R}^{|\mathcal{C}|}$: estimated class probability for pixel x by the model θ

Here, the difference lies in D0: AL uses only partial y, while ALC can access all y for each pixel x.
However, compared to ground-truth in Figure 1c, the initial pseudo-labels in Figure 1b contain noisy labels.
Therefore, active label correction is essential for rectifying these noisy labels.

3.2. Correction Query
Once we prepare the initial dataset for correction, we use our correction query to rectify the pseudo labels of pixels.

In the following, we information-theoretically compare the expected costs of classification and correction queries, denoted by Ccls and Ccor, respectively.

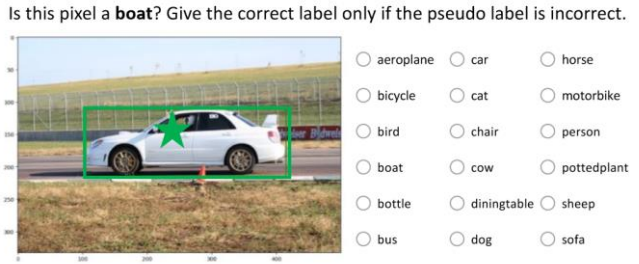


Figure 2: *An example of correction query*. Correction query presents an instruction requesting a label for a representative pixel (green star), an image displaying an object within a bounding box (green rectangle), and possible class options.

Theorem 3.1.

$\log_2 L$: information-theoretic annotation cost (Hu et al., 2020) of selecting one out of L possible options

$L \geq 2$: the number of classes

p : probability that the pseudo label is correct

$C_{\text{cls}}(L) = \log_2 L$: expected costs of classification queries

$C_{\text{cor}}(L, p) = p + (1-p)\log_2 L$: expected costs of correction queries

$$1 - \frac{C_{\text{cor}}(L,p)}{C_{\text{cls}}(L)} = \left(1 - \frac{1}{\log_2 L}\right) p \geq 0 \,. \qquad (2) \qquad \textit{for any } p \in [0,1] \textit{ and } L \geq 2$$

: cost-saving rate using the correction query instead on the classification one

The costs of both correction and classification queries are the same if L = 2. Indeed, those are logically identical when L = 2.

In (2), the cost-saving rate using the correction query instead on the classification one is computed as $\left(1 - \frac{1}{\log_2 L}\right) p$*, which is increasing in p and L.*

Hence, using the correction query is particularly beneficial when the number of classes is large or the pseudo labels can be obtained accurately.

3.3. Diversified Pixel Pool

$\mathcal{X}^d$: diversified pixel pool : subset of the total pixel set $\mathcal{X}$

$$\mathcal{X}^d := \{x_1, x_2, \ldots, x_{|\mathcal{S}|}\}\ , \qquad (3)$$

x_i : a key pixel from the superpixel s_i within the set of superpixels $\mathcal{S}$

Specifically, starting with a model θ_{t-1} trained on the dataset $\mathcal{D}_{t-1}$ from the previous round, we construct a diversified pixel pool $\mathcal{X}_t^d := \{x_{t1}, x_{t2}, \ldots, x_{t|\mathcal{S}|}\}$ for the current round t.

For ease of explanation,

$$\theta_{t-1} \longrightarrow \theta$$
$$x_{ti} \longrightarrow x_i$$
$$\mathcal{X}_t^d \longrightarrow \mathcal{X}^d$$

We select a representative pixel xi from each superpixel si based on the highest cosine similarity

$$x_i := \arg\max_{x \in s_i} \frac{f_\theta(x) \cdot f_\theta(s_i')}{\|f_\theta(x)\| \|f_\theta(s_i')\|}\ , \qquad (4)$$

We select a representative pixel xi from each superpixel si based on the highest cosine similarity

$$x_i := \arg\max_{x \in s_i} \frac{f_\theta(x) \cdot f_\theta(s'_i)}{\|f_\theta(x)\| \|f_\theta(s'_i)\|} , \qquad (4)$$

$f_\theta(s) := \frac{\sum_{x \in s} f_\theta(x)}{|\{x : x \in s|\}}$: averaged class prediction for superpixel s

To address the flaws in superpixels and ensure more uniformity of pixel labels within them, we employ a subset s' rather than the complete set s.

$\mathrm{D}_\theta(s)$: pseudo dominant label : representative label for superpixel s according to model θ

$$\mathrm{D}_\theta(s) := \arg\max_{c \in \mathcal{C}} |\{x \in s : y_\theta(x) = c\}| , \qquad (5)$$

$y_\theta(x) := \arg\max_{c \in \mathcal{C}} f_\theta(c; x)$: estimated label for pixel x using model θ

step1

s' : subset consisting of pixels that align with the pseudo dominant label $\mathrm{D}_\theta(s)$

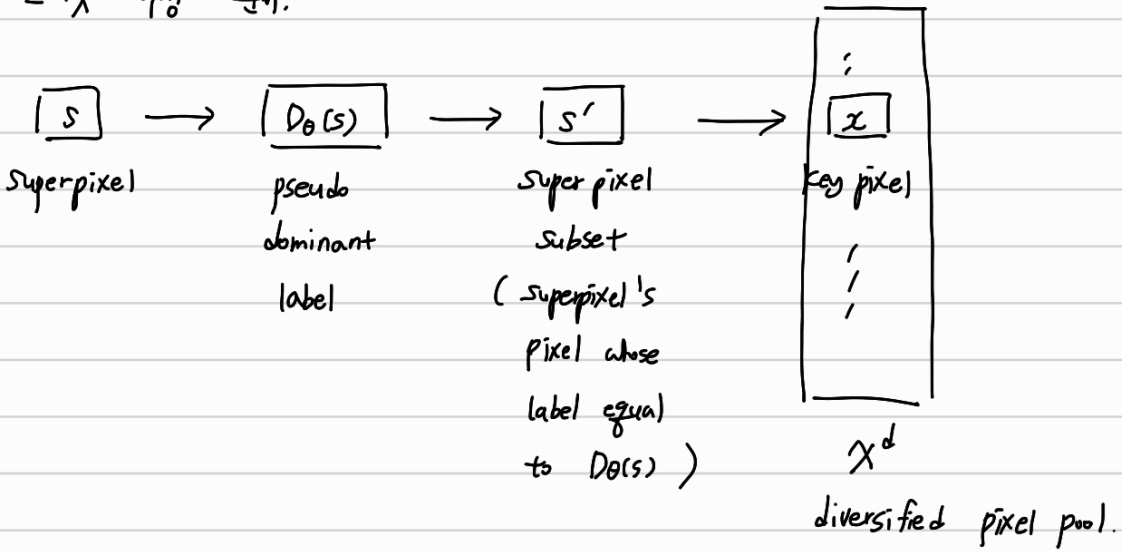
$$s' := \{x \in s : y_\theta(x) = \mathrm{D}_\theta(s)\} . \qquad (6)$$

step2

After that, we select the pixel that best represents s′ for each superpixel based on (4), contributing to the formation of a diverse pixel pool in (3).

step3

- χ^d 가 중요.


s
Superpixel
$D_\theta(s)$
pseudo dominant label
s'
Super pixel subset (superpixel's pixel whose label equal to $D_\theta(s)$)
x
Key pixel
χ^d
diversified pixel pool.

Remarks.
Therefore, we opt to organize superpixels based on the objects identified by Grounded-SAM.

3.4. Look-Ahead Acquisition Function

Once the set of pixels $\mathcal{X}_t^d$ for examination through an acquisition function is established, we select a pixel batch $\mathcal{B}_t \subset \mathcal{X}_t^d$ of size B to be corrected.

In each round t, we iteratively select the most informative pixel, guided by the acquisition $a(x; \theta_{t-1})$

$$x^* := \arg\max_{x \in \mathcal{X}_t^d} a(x; \theta_{t-1}) \ . \qquad (7)$$

$$a_{\text{CIL}}(x; \theta) := 1 - f_\theta(y; x) \ . \qquad (8)$$

: confidence in label (CIL), which evaluates the confidence of a given label y for a pixel x, using the predictions of the model θ

To enhance the efficiency of pixel-wise query, we introduce a label expansion technique, which involves extending the corrected label of a pixel x into pixels in the same superpixel s.

Look-ahead acquisition function : 아래 둘다 고려.
1) unreliability of a pixel x as described in (8)
2) effect of label expansion into the superpixel s

$$a_{\text{SIM}}(x_r; s, \theta) := \sum_{x \in s} \frac{f_\theta(x_r) \cdot f_\theta(x)}{\|f_\theta(x_r)\| \|f_\theta(x)\|} a_{\text{CIL}}(x; \theta) \ , \qquad (9)$$

: acquisition function (representative pixel x_r of s)

cosine similarity between two feature vectors is related to the likelihood of correctly expanding the correct label of pixel xr to another pixel x

We note that previous acquisitions including CIL in (8) can be transformed easily to its look-ahead counterparts.

For instance, the look-ahead CIL (LCIL) acquisition can be defined by adjusting the weight of each pixel from the cosine similarity to the inverse of the superpixel size.

$$a_{\mathrm{LCIL}}(x_r; s, \theta) := \sum_{x \in s} \frac{1}{|s|} a_{\mathrm{CIL}}(x; \theta) \ . \qquad (10)$$

Finally, in round t, we select the B most informative pixels from the diversified pixel pool $\mathcal{X}_t^d$ in order of SIM acquisition to form query batch $\mathcal{B}_t$.

After obtaining the clean labels of selected B pixels, we expand them to the associated superpixels.

We finally construct the dataset $\mathcal{D}_t$ for round t by combining the previous dataset $\mathcal{D}_{t-1}$ with the updated annotations.

Analogously to the warm-start, we initialize θ_t to a model pre-trained on ImageNet, minimizing the following CE loss.

$$\hat{\mathbb{E}}_{(x,y)\sim\mathcal{D}_t}[\mathrm{CE}(y, f_\theta(x))] \ . \qquad (11)$$