Supplementary Materials: Active Learning with Context Sampling and One-vs-Rest Entropy for Semantic Segmentation

Fei Wu[1], Pablo Marquez-Neila[1], Hedyeh Rafii-Tari[2], Raphael Sznitman[1]
[1]University of Bern
[2]Johnson & Johnson
{fei.wu,pablo.marquez,raphael.sznitman}@unibe.ch
hrafiita@its.jnj.com

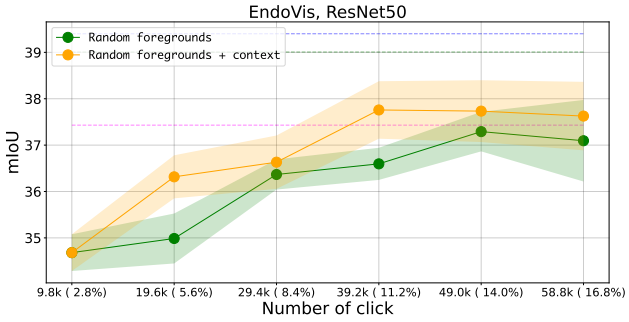

Figure 1. Comparison between a strategy that mainly samples foreground superpixels against a strategy that does the same but additionally samples superpixels in the immediate proximity of foreground superpixels as well. Values are averaged over 10 training-validation splits. Error bars indicate one standard deviation.

1. A simple experiment to test the efficiency of context sampling.

While we argue it is important to create context around objects we want to segment to further improve the segmentation accuracy, we wanted to confirm this behavior with a simple experiment. After splitting images into several superpixel patches, each superpixel is assigned the most prominent pixel label that it contains. We compare two sampling strategies: The first one selects superpixels from foreground classes with high probability and the second one acts the same as the first one but also selects background superpixels located around foreground superpixels (context) with high probability. The results displayed in Fig. 1 confirm our hypothesis that labeling the context around objects we want to segment improves the segmentation accuracy. Indeed the Active Leaning (AL) curve of *foregrounds + context* is always above the one of *foregrounds*.

2. OVR entropy and *OREAL*

In this section, we provide further details on *OREAL* .

2.1. Class Balacing

We go back to the Sec. 3.5 of the main paper and explain more in detail how *OREAL* balances the number of items per class using the OVR entropy. As mentioned in the main paper: "*For high OVR entropy H_c, the selected elements are the most uncertain for class c. However, uncertain samples of class c do not guarantee that they are actually from that class, potentially resulting in a deviation from the predetermined δ_c. However, in practice, this discrepancy does not significantly impact class balance, as any deviation from δ_c is compensated for in subsequent iterations.*"

Such behavior is illustrated in Fig. 2 through the application of *OREAL* on a toy dataset. The first batch selected for class 1 contains points from classes 2 and 3. However, batches selected for classes 2 and 3 contain points from class 1 as well, which compensated for the missing class-1 points in the first batch. Once all the points are selected, an oracle will provide the true label for all points no matter which class they were selected for.

2.2. Background class

Some datasets feature a background class that aggregates many different visual elements and exhibits high visual variability. For these datasets, balancing the background class with the other classes may result in insufficient coverage of its larger variability. We have observed a positive effect when oversampling the background class, which is achieved by simply enforcing $n_b = 0$ when computing class counts in Alg. 1, where b is the index of the background class. This results in more selected superpixels δ_b for the background class.

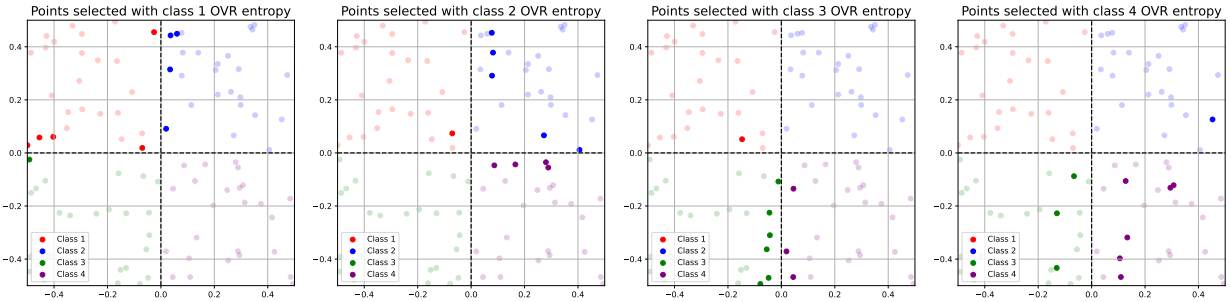

Figure 2. Illustration of our sampling method *OREAL* on a toy dataset. Starting with class 1, we calculate the OVR entropy of all samples for this class and select the samples with the highest OVR entropy. We repeat this process for the following classes iteratively. The number of samples to select per class depends on the number of labeled samples per class, the fewer labels we have for this class the more we select for it.

Algorithm 1 Required items per class

▷ Solution to Eq. 3 in the main paper

Input: $\{n_c\}_{c=1}^{C}$ (current number of labeled items per class in $\mathcal{A}_t$), Q (number of items to select for annotation)
Output: $\{\delta_c\}_{c=1}^{C}$ (number of items to select per class)

function ITEMSPERCLASS($\boldsymbol{n}$, Q)
  $\boldsymbol{\delta} \leftarrow \mathbf{0}$
  while $Q > 0$ **do**
    $c \leftarrow \arg\min_c n_c + \delta_c$
      ▷ Get the most underrepresented class c
    $\delta_c \leftarrow \delta_c + 1$
      ▷ Increment the queried items for the class c
    $Q \leftarrow Q - 1$ ▷ Decrement the available budget
  end while
  return $\boldsymbol{\delta}$
end function

2.3. OREAL ablations

. We perform different ablations of our method to measure the contribution of each component to the final performance. Results of the ablation are shown in Table 1 and Fig. 3. Our OVR entropy and the use of max_{agg}have the largest effect on the final performance, while the oversampling of the background classes helps to improve the performance by a lesser but not negligible margin.

OREAL $mean_{agg}$**.** We use the mean of pixel-wise OVR entropy scores instead of their maximum to compute superpixel OVR entropy.

OREAL **Proba**$\times$**Entropy.** Substitute the OVR entropy for an uncertainty measure obtained as the product of entropy times the predicted probability vector. The goal of this ablation is to compare the use of OVR entropy and the vanilla entropy [10] when using class balance.

OREAL δ**.** We omit the special treatment of the background class (Sec. 2.2) for the datasets where it is applied (MONARCH, EndoVis, Pascal VOC).

OREAL 95%**.** Instead of using the max aggregation, we take the 95% percentile pixel value to represent the superpixel.

3. Experimental Settings

We further detail some aspects of our experiments in this section.

3.1. Baselines

We provide below a detailed explanation of baselines presented in the main paper:

Random selects Q random superpixels from $\mathcal{U}_t$ at each iteration.

BvSB **[9]** estimates the uncertainty of a pixel using the ratio between the most confident and second most confident class posterior. The uncertainty of a superpixel is the average of the pixel uncertainty within it. It then selects the superpixels with the highest uncertainty.

Revisiting SP **[3]** uses the *BvSB* uncertainty [9] weighted by the posterior of the class distributions to promote class balancing. Class distributions are computed for each class as the proportion of superpixels predicted to be that class. Their strategy selects the superpixels with the highest weighted uncertainty.

PixelBal **[8]** extends the *Revisiting SP* method computing the class distribution for class c as the sum of the predicted probabilities over all pixels for that class divided by the total number of pixels. The superpixel score is computed as the average of the pixel-wise weighted uncertainties. The weighting requires a hyperparameter ν, which is set to $\nu = 6$ for Cityscapes and $\nu = 12$

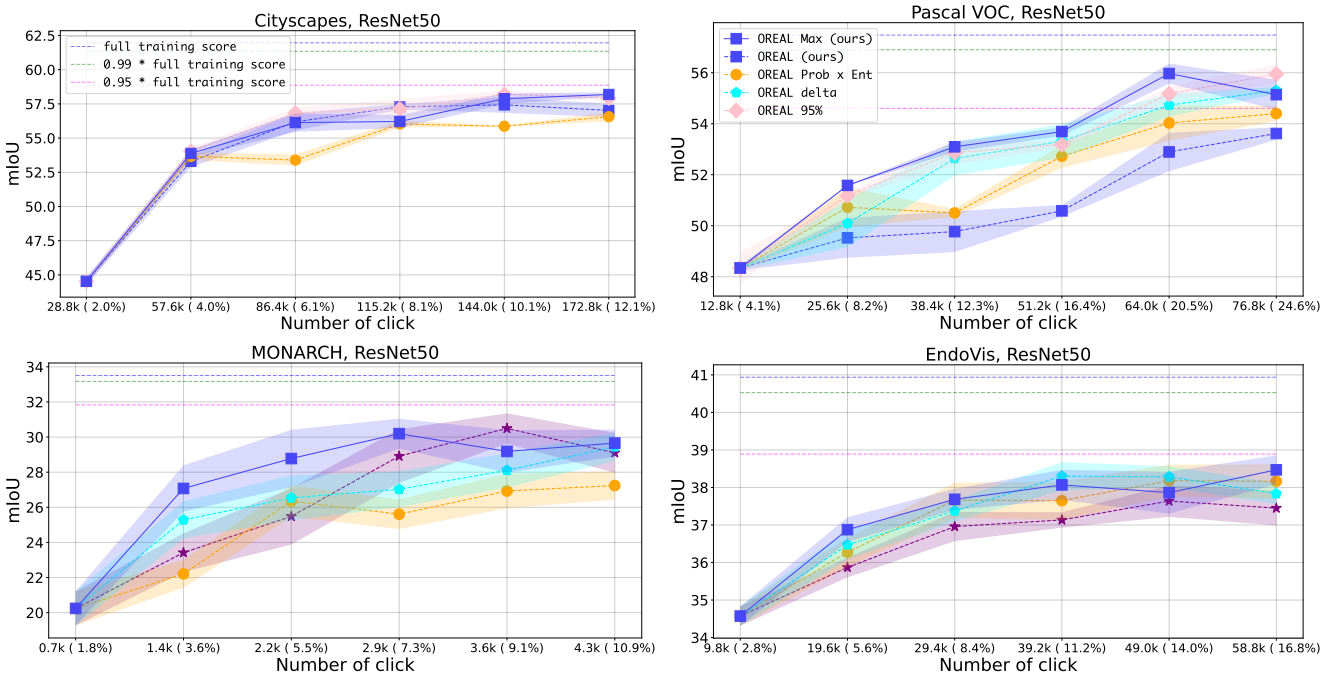

Figure 3. Comparison of different ablation strategies. Values are averaged over 3 training-validation splits for Cityscapes [5], Pascal VOC [6], and 10 training-validation splits for EndoVis [1], MONARCH [11]. Error bars indicate one standard deviation.

Strategy	Cityscapes	Pascal VOC	MONARCH	EndoVis	Average	Avg /Cityscapes
OREAL	68.0	**77.2**	**69.7**	**76.2**	**72.8**	**74.4**
OREAL Mean	67.9	73.6	66.2	74.8	70.6	71.5
OREAL Prob x Ent	66.5	75.2	62.1	75.8	70.0	71.0
OREAL δ	.	76.2	65.6	0.760	.	72.6
OREAL 95%	**68.5**	76.6	.	.	.	.

Table 1. AuALC of all ablations across all datasets averaged over 10 runs (MONARCH, EndoVis) and 3 runs (Cityscapes, Pascal VOC) on the ResNet50 backbone. Metrics are computed at the end of the 6 active learning steps. Best scores are in bold, and second best are underlined.

for Pascal VOC, following the authors' original values. For EndoVis and MONARCH we set $\nu = 12$. As above, their strategy selects superpixels with the highest weighted uncertainty.

CBAL **[2]** scores the uncertainty of a sample by calculating a residual entropy obtained from the standard entropy by subtracting the L^1-norm between the predicted probability vector and the class counts vector $\boldsymbol{\delta}$. The L^1-norm is scaled by a factor λ, which we set to $\lambda = 2$ for all datasets. To adapt the original work from classification to segmentation, we average the pixel-wise entropies and probability vectors over all pixels within each superpixel before computing the residual entropy scores at the superpixel level. The strategy selects the superpixels with the highest residual entropy.

3.2. Data Augmentations

We apply data augmentation for training our segmentation model. For MONARCH [11] and EndoVis [1], we use the same data augmentations as in [11]: MONARCH and EndoVis images are first resized to 220×220 and 224×224, respectively. We then take crops with random scale factors in the range $(0.85, 1)$ and with random aspect ratios in the range $\left(\frac{3}{4}, \frac{4}{3}\right)$. All crops are re-scaled back to the initial size of 220×220 or 224×224, followed by a random horizontal flipping with a probability of 0.5.

For Cityscapes [5] and Pascal VOC [6], we start by resizing the images to 769×769 and 513×513 respectively, as done in [3]. Next, we upscale the images by adding a random number of ℓ pixels in both the horizontal and vertical dimensions. The value of ℓ follows a uniform distri-

bution $\mathcal{U}(0, 300)$ for Cityscapes and $\mathcal{U}(0, 200)$ for Pascal VOC. After upscaling, we center-crop the images to their original size of 769×769 or 513×513 and randomly flip them horizontally with a probability of 0.5.

3.3. Running Time

The time needed to run our AL pipeline for one seed is respectively 24, 8, 5, and 6 hours for Cityscapes, Pascal VOC, MONARCH, and EndoVis. This time corresponds to a complete AL pipeline, from iterations 0 to 5, per seed. The time needed to complete the AL pipeline is roughly the same for all AL methods. We used a single RTX3090Ti (24GB) for our experiments.

3.4. Reproducibility

For all AL methods, the initial training phase, denoted as round 0, utilizes the same training dataset to train the segmentation model. An identical test score is observed across different methods at the conclusion of round 0. Although setting the seed for random number generators in numpy and pytorch contributed to reproducibility, it proved insufficient for ensuring perfect consistency. A crucial modification was made to the segmentation model, specifically to the interpolation layer within DeepLabV3 [4]. By altering the interpolation mode from bilinear to nearest, we achieved deterministic computation, ensuring that the model behavior became predictable and reproducible. Since different AL methods when trained on the same dataset at round 0 achieve the same test score, in the following AL iterations (1 to 5), the difference observed between AL methods is only attributable to the labeled set each of them built.

3.5. Weak Labeling of Patches

OREAL was designed under the assumption that superpixels only require one click to be annotated. Given a budget of Q clicks, we can thus annotate Q superpixels. However, when patches are weakly labeled [8], the annotator declares all the classes present in the superpixel and requires one click per class. Thus, some superpixels, with more than one class, would require more than one click to be annotated. With a budget Q, we will then have less than Q annotated superpixels.

In the case of *OREAL*, once the list of superpixel numbers to be selected per class δ (the numbers in δ sum up to Q) is computed, starting with the first class, we select all the required number of superpixels for this class before proceeding for the next class. When superpixels only require one click to be annotated, we are sure to select the required number of superpixels for all classes. However, when superpixels require more than one click, our budget can be exhausted before we can start selecting superpixels for the last classes. To resolve this issue, we simply select one superpixel per class and proceed to the next class. This way we ensure all classes are equally selected. Our selection procedure stops when the annotation budget is exhausted and the rate at which this budget decreases will depend on the number of classes per superpixel we encounter.

3.6. Other details

AL initialization. After validation split, the set of labeled frames $\mathcal{A}_1$ is initialized with Q random superpixels throughout the training set. The remaining superpixels are assigned to the complementary set of unlabeled frames $\mathcal{U}_1$.

Background class. We applied the special treatment for background classes of Sec. 2.2 to the MONARCH, EndoVis, and Pascal VOC datasets. This treatment was omitted for the *void* class of Cityscapes because it is not used to train or evaluate the model.

4. Additional Results

We show in this section complementary results that did not fit in the main paper due to lack of space.

4.1. Active Learning Curves

Fig. 4 shows the Active Learning curves using the ResNet101 backbone and Fig. 5 shows the Active Learning curves using the ViT backbone.

4.2. mIoU improvement using max aggregation

Tab. 2 shows the average mIoU improvement across all AL iterations from using $mean_{agg}$ to max_{agg}. On average, all AL methods benefit from using max_{agg} instead of $mean_{agg}$ with the Pascal VOC that has an average mIoU improvement of 1.54 across all AL methods and backbones.

4.3. Selected superpixels on additional datasets

The main paper showed selected superpixels for the Pascal VOC and EndoVis datasets. Fig. 6 shows additional superpixels selected for the Cityscapes and MONARCH datasets. Similar to the selected superpixels on the Pascal VOC and EndoVis datasets, max_{agg} has selected more regions at the boundaries between objects.

4.4. Superpixel annotated frames

Fig. 7 shows examples of superpixel annotated frames using the dominant labeling scheme. For datasets like Pascal VOC, Endovis, and MONARCH the qualitative quality of the superpixel annotated labels is correct when compared to the original segmentation masks. This explains the small performance gap between using the original segmentation masks and the superpixel annotated masks in Tab. 2 of the main paper. In the case of Cityscapes which display much more complex sceneries, the use of superpixel annotation had a bigger impact on decreasing the segmentation mask quality. As such, the performance gap in Tab. 2 of the main paper for Cityscapes becomes bigger.

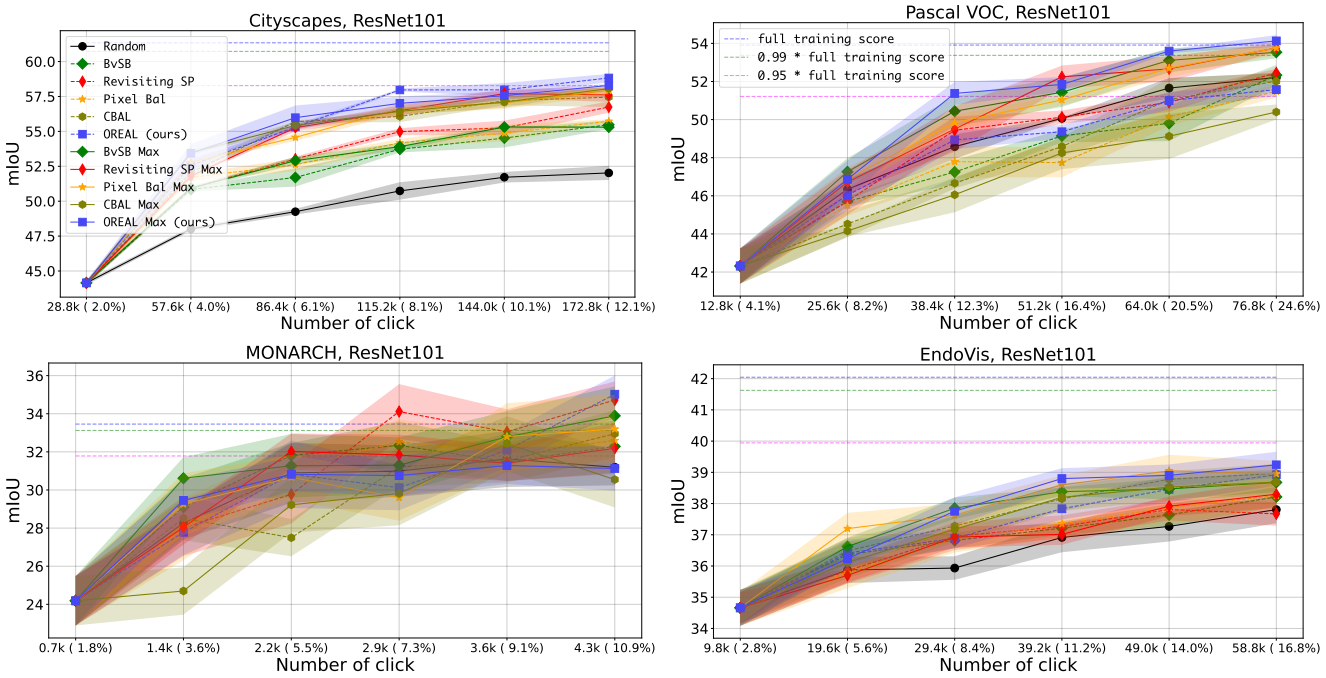

Figure 4. Comparison of different sampling strategies. Values are averaged over 3 training-validation splits for Cityscapes [5], Pascal VOC [6], and 10 training-validation splits for EndoVis [1], MONARCH [11] on the ResNet101 backbone. Error bars indicate one standard deviation.

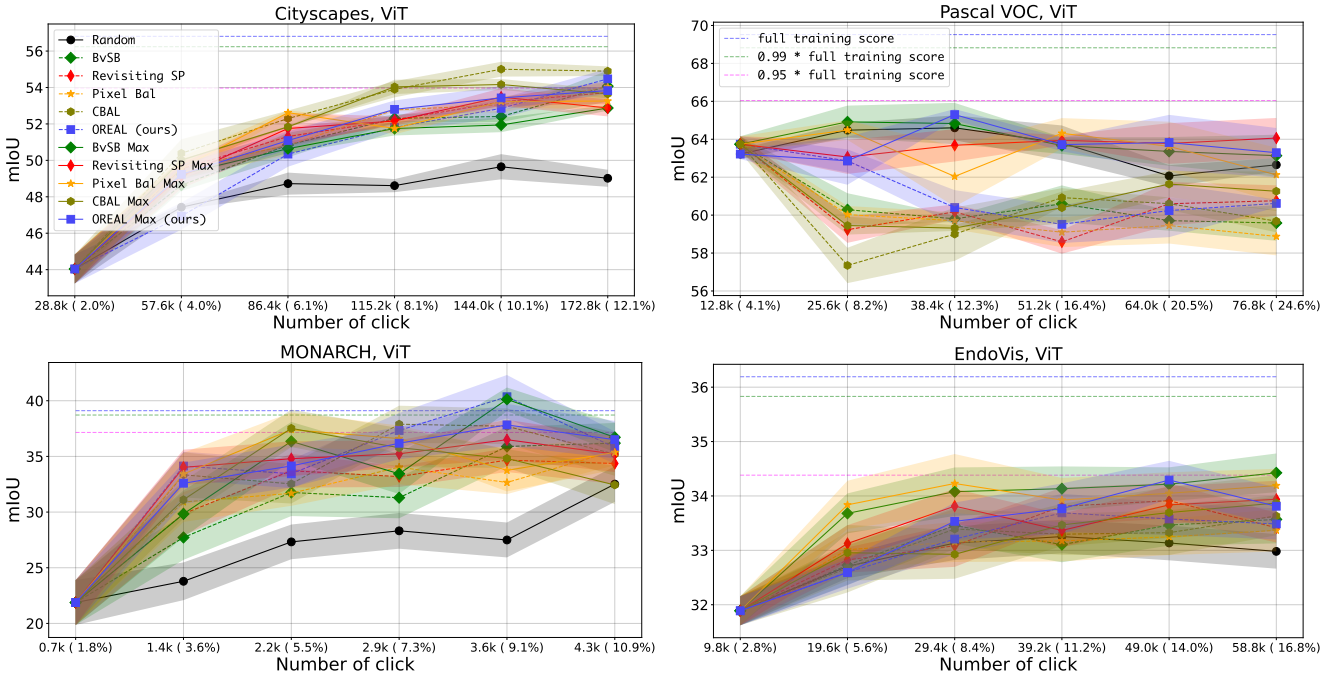

Figure 5. Comparison of different sampling strategies. Values are averaged over 3 training-validation splits for Cityscapes [5], Pascal VOC [6], and 10 training-validation splits for EndoVis [1], MONARCH [11] on the ViT backbone. Error bars indicate one standard deviation.

Method	Datasets / Backbone	MONARCH	PASCAL VOC	ENDOVIS	CITYSCAPES	Average
BvSB	RN50	1.47	1.36	0.29	-0.19	0.73
revisiting SP	RN50	1.67	0.32	-0.18	0.41	0.56
Pixel Bal	RN50	-1.31	1.54	0.39	-0.41	0.05
CBAL	RN50	-0.31	-0.10	-0.14	0.07	-0.12
OREAL (ours)	RN50	1.25	2.18	0.65	0.18	1.07
BvSB	RN101	0.48	1.93	0.62	0.35	0.85
revisiting SP	RN101	-0.70	1.04	0.05	1.32	0.43
Pixel Bal	RN101	-0.10	1.93	0.89	1.64	1.09
CBAL	RN101	-0.83	-0.81	-0.16	0.34	-0.36
OREAL (ours)	RN101	-0.40	1.82	0.40	-0.06	0.44
BvSB	ViT	2.27	3.32	0.70	-0.30	1.50
revisiting SP	ViT	1.66	3.20	0.17	-0.10	1.23
Pixel Bal	ViT	1.95	3.24	0.71	0.03	1.48
CBAL	ViT	-0.84	0.75	0.10	-0.46	-0.11
OREAL (ours)	ViT	-0.64	2.47	0.24	0.66	0.68
Average		0.37	1.54	0.33	0.30	0.64

Table 2. Average mIoU improvement from using $mean_{agg}$ to max_{agg} across all AL iterations of all strategies across all datasets. A positive number n means, max_{agg} has achieved a higher mIoU than $mean_{agg}$ by a margin of n.

4.5. Weak Labeling of Patches

Fig. 8 shows the AL curves for the results presented in Tab. 4 of the main paper. Finally, Tab. 3 shows the average mIoU improvement across all AL iterations from using $mean_{agg}$ to max_{agg} in the case of the weak labeling scheme. While the performance gap is marginal on Pascal VOC, it is significant on Cityscapes which benefits from max_{agg}.

4.6. Gini Index

To assess how well each AL method balances the different classes of a dataset, we calculate the Gini Index [7] for datasets built by each AL method at the end of the AL iterations and show the results in Tab. 4. The Gini index is equal to 0 for a list of uniform values and converges to 1 otherwise. Overall we can see that *OREAL* built a better class-balanced dataset than the other AL methods. In the case of *OREAL*, using mean aggregation should be more precise at balancing the dataset than using max aggregation. This is the case for datasets such as Cityscapes and EndoVis but not for MONARCH and Pascal VOC. We think this behavior is caused by the fact that images from MONARCH and Pascal VOC display on average less than 2 classes per image against 6 to 12 classes for EndoVis and Cityscapes. Since the task of balancing classes is more difficult when having multiple classes per image, the difference between a better and worse balancing metric is more noticeable.

4.7. AuALC improvement per class

Fig. 9 and 10 display the AuALC improvement per class of *OREAL* against all baselines for the Cityscapes and Pascal VOC dataset respectively. Overall, *OREAL* could achieve better results than baseline on tail classes, whether AL methods are using the mean or max aggregation.

4.8. Entropy Map

Fig. 11 shows examples of entropy maps for Pascal VOC images. For the majority of superpixels located at the border of objects, we observe that only a few pixels in the proximity of the object boundary have high entropy. Hence using max aggregation for those superpixels greatly differs from using mean aggregation and allows the uncertainty-based AL methods to select such superpixels.

References

[1] Max Allan, Satoshi Kondo, Sebastian Bodenstedt, Stefan Leger, Rahim Kadkhodamohammadi, Imanol Luengo, Félix Fuentes-Hurtado, Evangello Flouty, Ahmed Kedir Mohammed, Marius Pedersen, Avinash Kori, Alex Varghese, Ganapathy Krishnamurthi, David Rauber, Robert Mendel, Christoph Palm, Sophia Bano, Guinther Saibro, Chi-Sheng Shih, Hsun-An Chiang, Juntang Zhuang, Junlin Yang, Vladimir Iglovikov, Anton Dobrenkii, Madhu Reddiboina, Anubhav Reddy, Xingtong Liu, Cong Gao, Mathias Unberath, Mahdi Azizian, Danail Stoyanov, Lena Maier-Hein, and Stefanie Speidel. 2018 robotic scene segmentation challenge. *CoRR*, abs/2001.11190, 2020. 3, 5

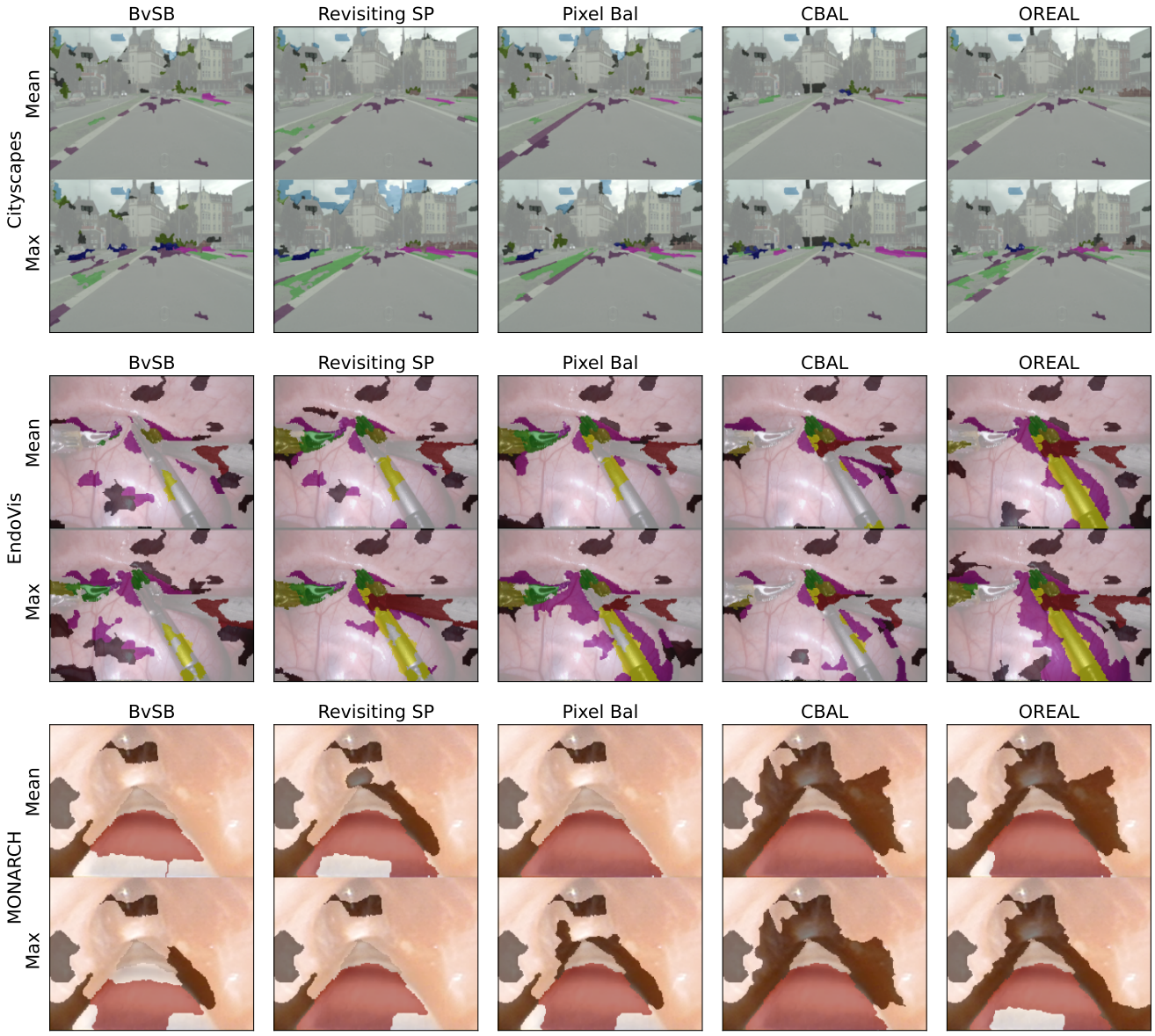

Figure 6. Selected superpixel regions of different sampling strategies. On average, we observe that when using the max aggregation, all sampling methods tend to select more regions around the boundary of objects.

[2] Javad Zolfaghari Bengar, Joost van de Weijer, Laura Lopez Fuentes, and Bogdan Raducanu. Class-balanced active learning for image classification, 2021. 3

[3] 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 (CVPR)*, pages 10988–10997, June 2021. 2, 3

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 4

[5] 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. 3, 5

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 3, 5

[7] Frank Farris. The gini index and measures of inequality. *American Mathematical Monthly*, 117:851–864, 12 2010. 6

[8] Sehyun Hwang, Sohyun Lee, Hoyoung Kim, Minhyeon Oh, Jungseul Ok, and Suha Kwak. Active learning for semantic segmentation with multi-class label query, 2023. 2, 4, 8, 9

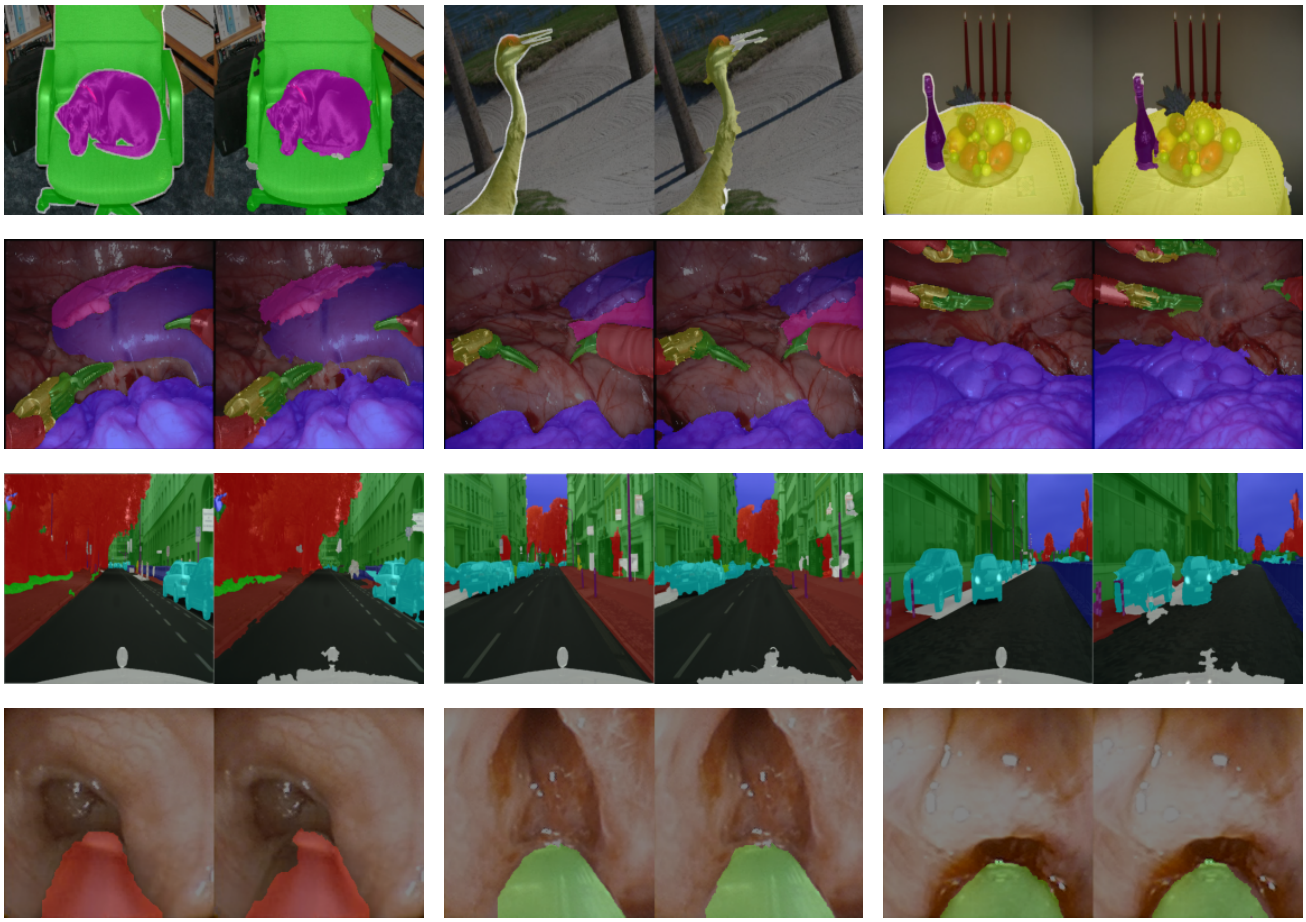

Figure 7. For each pair of images, the original segmentation mask is on the left and the segmentation mask annotated using superpixel dominant labeling is on the right. Each row shows respectively pair of images from the Pascal VOC, EndoVis, Cityscapes, MONARCH datasets. The complete mIoU of the dominant labels are 93.7, 83.3, 74.6, and 82.3, for Pascal VOC, EndoVis, Cityscapes, and MONARCH respectively.

[9] 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, 2009. 2

[10] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. 2

[11] Fei Wu, Pablo Marquez-Neila, Mingyi Zheng, Hedyeh Rafii-Tari, and Raphael Sznitman. Correlation-aware active learning for surgery video segmentation, 2023. 3, 5

Method	Stage	Pascal VOC mean - max	Cityscapes mean - max
BvSB	1	-0.13	0.40
revisiting SP	1	0.41	0.98
Pixel Bal	1	-0.15	0.24
OREAL	1	-0.17	0.26
BvSB	2	-0.11	0.42
revisiting SP	2	-0.14	1.33
Pixel Bal	2	-0.34	0.42
OREAL	2	0.07	0.16
Average		-0.07	0.65

Table 3. Average mIoU improvement from using $mean_{agg}$ to max_{agg} across all AL iterations of all strategies across all datasets when patches are annotated using the weak labeling scheme [8]. A positive number n means, max_{agg} has achieved a higher mIoU than $mean_{agg}$ by a margin of n.

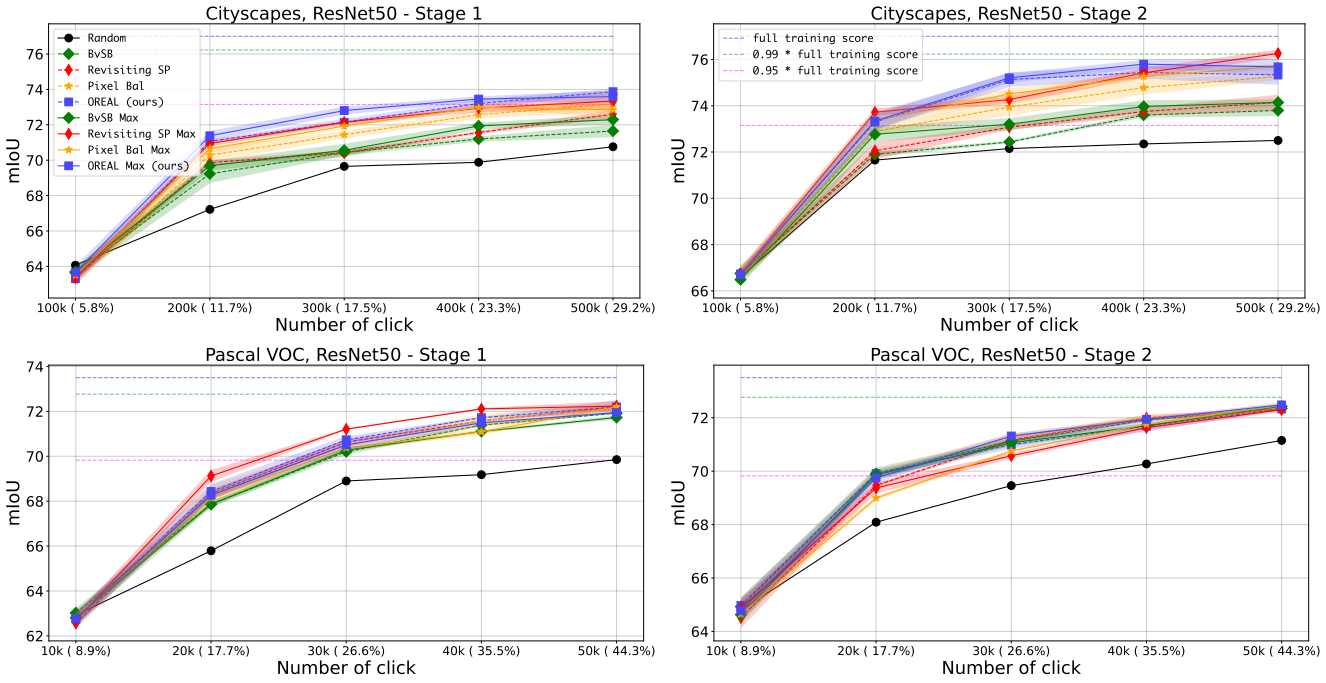

Figure 8. Comparison of different sampling strategies when patches are weakly annotated following the method proposed by [8]. Values are averaged over 3 training-validation splits. Error bars indicate one standard deviation.

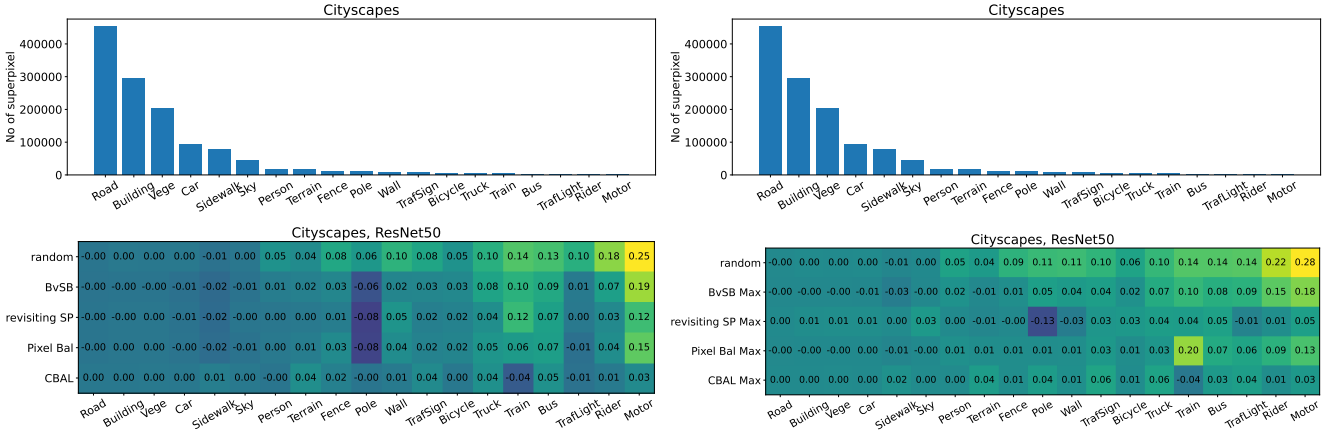

Figure 9. Top 2 graphs show the distribution of Cityscapes classes. The bottom 2 graphs show the relative AuALC improvement of OREAL vs. baselines for each Cityscapes class (positive indicates OREAL is better). The bottom left graph is for methods using the mean aggregation and the bottom right is for methods using the max aggregation.

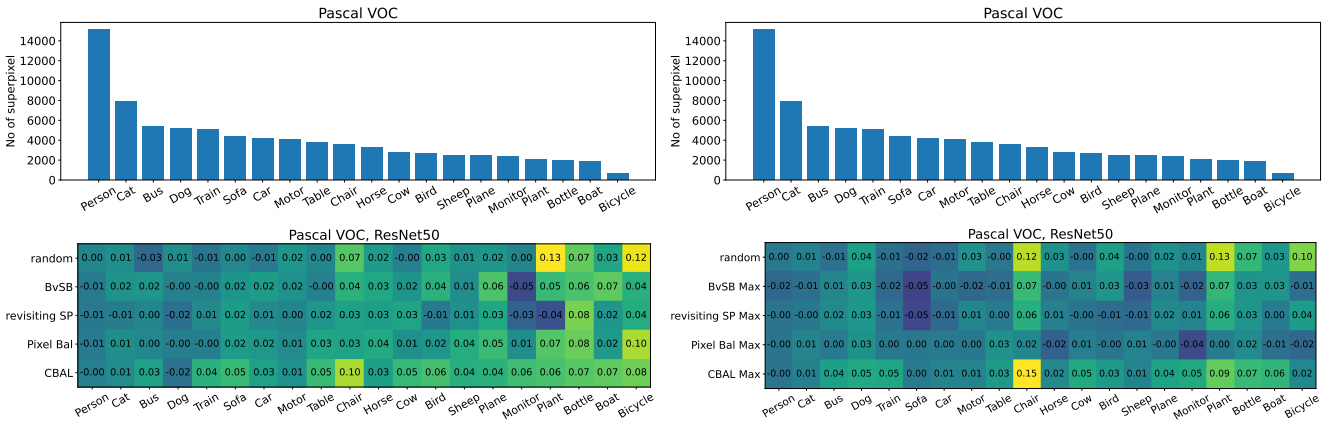

Figure 10. Top 2 graphs show the distribution of Pascal VOC classes. The bottom 2 graphs show relative AuALC improvement of OREAL vs. baselines for each Pascal VOC class (positive indicates OREAL is better). The bottom left graph is for methods using the mean aggregation and the bottom right is for methods using the max aggregation.

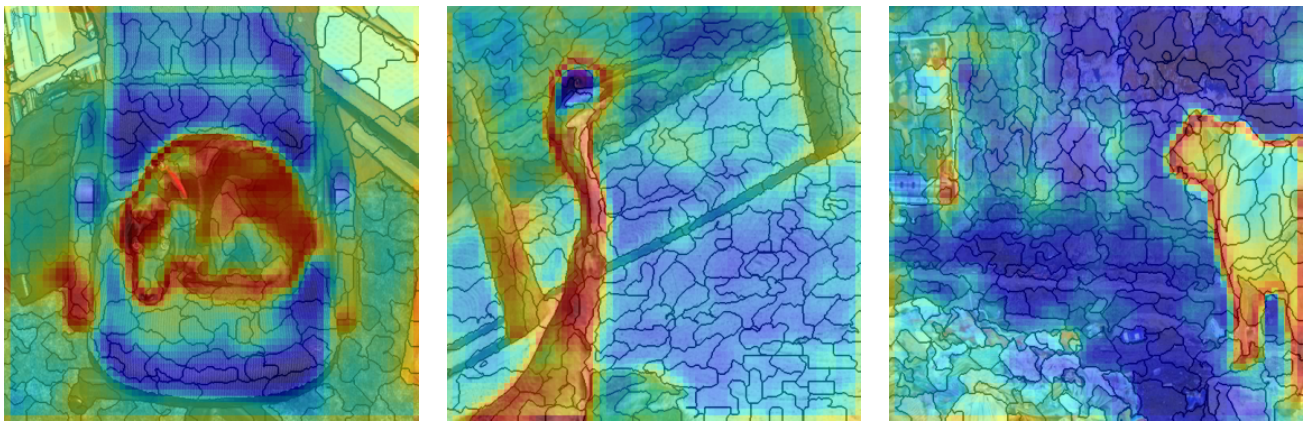

Figure 11. Entropy map of 3 images from Pascal VOC. The images are split into superpixels. Pixel with high entropy belongs respectively to the Dog, Bird, and Cow classes. On the border of these classes, some pixels overflow and belong to superpixels surrounding the classes. Hence using max aggregation for those border superpixels will emphasize the few pixels that have high entropy located at the border with the classes.

Methods	MONARCH	PASCAL VOC	CITYSCAPES	ENDOVIS
	mean - max	mean - max	mean - max	mean - max
Random	0.42	0.67	0.25	0.41
BvSB	0.35 - **0.40**	0.61 - **0.63**	**0.35** - **0.35**	0.45 - **0.46**
Revisiting SP	0.40 - **0.47**	0.63 - **0.66**	0.42 - **0.56**	0.49 - **0.52**
Pixel Bal	0.42 - **0.53**	0.60 - **0.72**	**0.39** - 0.38	0.50 - **0.52**
CBAL	0.56 - **0.59**	0.65 - **0.66**	**0.48** - 0.47	**0.55** - **0.55**
OREAL (ours)	0.56 - **0.59**	0.83 - **0.86**	**0.54** - 0.51	**0.57** - 0.56

Table 4. ”1 - Gini Index” of class distribution of datasets built by different AL methods at the end of AL iterations on the ResNet50 model. The best value between each pair of mean - max aggregation is in bold and the best methods for each column of mean - max aggregation is underlined.