Active Learning for Semantic Segmentation with Multi-class Label Query

—Supplementary Material—

Anonymous Author(s)
Affiliation
Address
email

This document, 'supplement_5115.pdf', is part of our supplementary material, which consists of three distinct files: (1) 'code' containing the implementation of our framework, (2)'pre-survey.pdf' that includes the instructional material provided to participants for the user study, and (3) this document itself which provides extensive analysis and additional findings that have been omitted in the main paper due to the page limit. Sec. A presents a thorough explanation of the user study conducted to compare the cost of dominant class labeling and multi-class labeling. In Sec. B, we explain further details of our experiment, including configurations of our implementation (Sec. B.1) and our code (Sec. B.2). Section C presents an in-depth analysis of our framework, including the effect of hyper-parameters (Sec. C.1), the budget size (Sec. C.2), and the size of the local regions (Sec. C.3). Lastly, a comparison with partial label learning loss baselines (Sec. D.1), a comparison with pseudo labeling baselines (Sec. D.2), and a qualitative result of our final model (Sec. D.3) are provided in Section D.

A Details of user study

We conducted a user study to compare the dominant class labeling and multi-class labeling in terms of actual labeling cost and accuracy versus the number of classes in region queries. The examples of the questionnaire are illustrated in Fig. 1 and the results are summarized in Table 1. As shown in Fig. 1(a), for each question, annotators received an instruction, an image patch along with a marked local region, and class options. They were requested to select the relevant class options as directed by the instruction. The instructions for dominant class labeling and multi-class labeling were as follows:

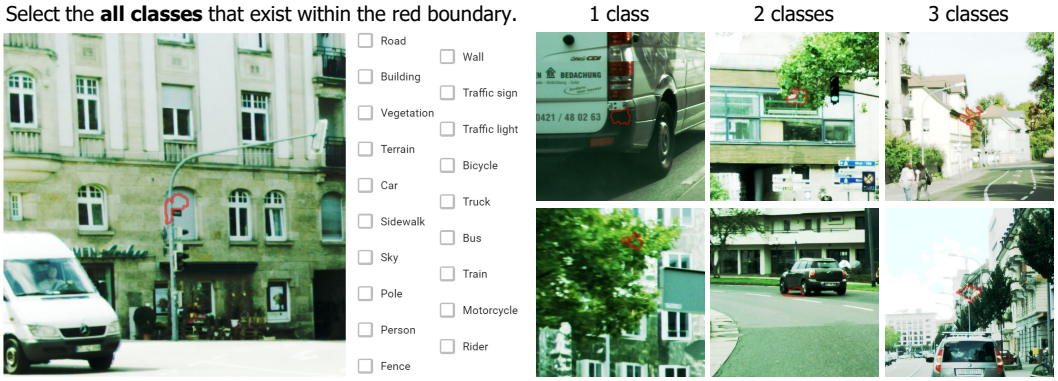
> "Select the dominant class that corresponds to the inside of the red boundary.",
> "Select the all classes that exist within the red boundary.".

Prior to the survey, we ensured that every participant reviewed the pre-survey instructional material. This material covered the class composition of Cityscapes, offered the definition of the dominant class and the multi-class labeling, and provided example questions. The pre-survey instructional material is included in the supplementary materials under the name 'pre-survey.pdf'.

As shown in Fig. 1(b), each image patch was a 360-pixel square mostly centered on a local region. Using ground-truth segmentation mask, we divided regions into three groups based on the number of classes (from 1 to 3) present in each region. Twenty regions were then randomly selected from each group for each survey, excluding those containing pixels irrelevant to the original 19 classes, referred to as the 'undefined' class.

A total of 45 volunteers participated in the survey. We report the results excluding five cases considered outliers in terms of time and accuracy. A unique survey was prepared for each group of regions, categorized by the number of classes. Given three groupings and two labeling methods, a total of six unique forms were prepared. If an annotator annotates the same region twice, there would

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.



(a) Questionnair for multi-class labeling (b) Example queries with different number of classes

Figure 1: Questionnaire and local region examples used in the user study. (a) Questionnaire of multi-class labeling survey, consisting of instruction, image patch along with local region marked with red boundary, and class options allowing multiple selections. (b) Examples of local regions used in the user study according to the number of classes present in each region.

Table 1: The result of user study showing the labeling time (second) and accuracy (%) of dominant class labeling and multi-class labeling according to the number of classes within each region.

Query	# of classes	Total time (s)	Total clicks	Time per click (s)	Accuracy (%)
Dominant	1	127.6±39.4	20.0±0.0	6.38±1.97	95.63±6.00
	2	160.5±33.0	20.0±0.0	8.02±1.65	72.05±5.36
	3	172.1±35.8	20.0±0.0	8.60±1.79	65.83±6.07
	average	153.4±41.1	20.0±0.0	7.67±2.05	77.84±14.24
Multi-class	1	145.5±41.8	21.6±2.1	6.75±1.65	95.97±4.10
	2	191.6±65.8	39.1±3.7	4.89±1.54	87.14±5.21
	3	295.8±65.3	49.0±8.6	6.37±1.51	71.52±8.42
	average	211.0±86.3	36.5±12.5	6.01±1.75	84.88±11.76

be a risk of memorizing the image during the first annotation. To avoid this, we asked each participant to answer three out of the six forms, ensuring no region was annotated twice by the same person.

The responses from annotators are evaluated by calculating the Jaccard Similarity (JS) between the ground-truth class set and the responded class set. We define the JS of annotator u as follows:

$$\text{JS}(u) = \frac{1}{|X|} \sum_{i \in X} \frac{|G_i \cap Y_{i,u}|}{|G_i \cup Y_{i,u}|} , \tag{1}$$

where X is a set of regions, G_i is ground-truth multi-class label and $Y_{i,u}$ is the set of classes selected by annotator u for region i. Note that for dominant class labeling, $|G_i| = |Y_{i,u}| = 1$. We compute the final accuracy as the average JS of all annotators, given by:

$$\text{Accuracy} = \frac{1}{|U|} \sum_{u \in U} \text{JS}(u) . \tag{2}$$

As shown in Table 1, multi-class labeling demonstrates comparable efficiency to dominant class labeling for regions with a single class. Moreover, when it comes to regions with multiple classes, multi-class labeling requires less annotation time per click compared to the dominant class labeling.

B Further experiment details

B.1 Implementation details

Configurations. We implement our method using the PyTorch framework [7]. Following the previous literature [5], we make a slight modification to the original ResNet architecture by replacing the

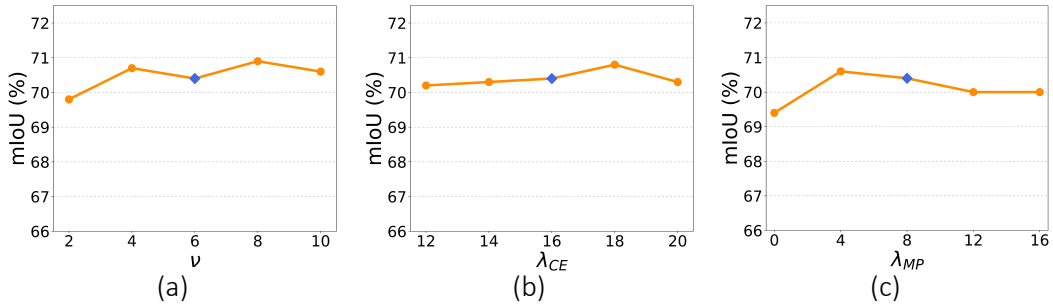


Figure 2: Average accuracy of our stage 1 model over 5 rounds, represented as mIoU (%), as a function of varying hyperparameters. The model is evaluated on Cityscapes using a ResNet50 backbone in combined with PixBal sampling. (a) The class balancing regulation term ν. (b) Loss balancing term λ_{MP}. (c) Loss balancing term λ_{CE}. The blue diamond marker indicates the value selected for our final model.

initial 7×7 convolutional layer with two 3× 3 convolutional layers. The output stride of the network is set to 16. We set the learning rate of the backbone to be ten times lower than the standard rate, and we apply a weight decay of $1\text{e}{-5}$. During the training phase, we incorporate several data augmentation techniques, including random scaling ranging from 0.5 to 2.0, random cropping, and random horizontal flipping. To ensure reproducibility and to test the robustness of our approach, we conduct three independent experiments, each initialized with different seed values: 0, 1, and 2.

Label generation. Following previous work [2], we assign both dominant class labels and multi-class labels to each region using the ground-truth mask labels. For the dominant class label, we assign the class that dominates the majority of pixels within each region, in accordance with its definition. For the multi-class labels, we attribute all existing classes present within each region. Notably, we disregard classes that appear minimally along the region's boundary during this process. This procedure reflects realistic scenarios where a labeler might fail to recognize classes represented insignificantly on the boundary. More specifically, we implement a binary dilation operation with a 5×5 kernel along the region boundaries, consequently excluding classes that appear on these expanded boundaries. In the user study, we reflect this by marking each local region with a thick, translucent boundary.

Handling undefined class. In the Cityscapes dataset [3], pixels not covered by the original 19 semantic classes are typically ignored when training segmentation models. On the other hand, in multi-class labeling setting, the precise locations of such uncovered pixels remain unspecified since the multi-class label only provide partial labels. Treating such pixels as belonging to one of the 19 semantic classes naively can misguide the model by providing confusing supervision. Furthermore, active sampling methods like BvsB, ClsBal, PixBal tend to prefer uncertain regions, often leading to the selection of regions containing these uncovered pixels, despite their lack of utility. To address this, we assign an additional *undefined* class for pixels not covered by the initial 19 classes and train the model to predict these undefined classes. As for active sampling, we introduce an extra condition to exclude regions where the predicted dominant class is the undefined class. This undefined class handling strategy is implemented for both dominant class labeling and multi-class labeling.

B.2 Code

In the supplementary materials, we include a implementation of our proposed framework, along with pre-trained checkpoint files. Detailed instructions for training and running the model are available in the 'README.md' file. The script for our sampling method can be found in the 'active_selection/my_bvsb_predclsbal_pwr_banignore.py' file. The implementation of $\mathcal{L}_{\text{MP}}$ and $\mathcal{L}_{\text{PP}}$ are located at lines 64-70 and lines 97-128 of the 'trainer/active_joint_multi_predignore_lossdecomp.py' file, respectively. The techniques used for intra-region label localization and label expansion are coded into lines 120-305 of the 'trainer/eval_save_cosplbl_prop_includeonehot.py' file. We would like to acknowledge that parts of our code were borrowed from the implementation of D2ADA [8][1]. Due to the parallelization of training each region via GPU, the time complexity remains comparable

[1] https://github.com/tsunghan-wu/D2ADA/tree/main

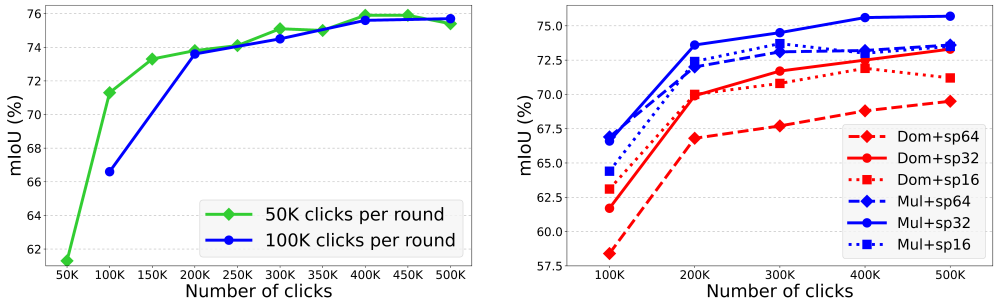


Figure 3: Accuracy in mIoU (%) versus the number of clicks (budget) evaluated on Cityscapes using ResNet50 combined with PixBal sampling. (a) The accuracy of our final model (Mul) with different budget sizes and the number of rounds: 50K $\times$ 10 rounds, and 100K $\times$ 5 rounds. (b) The accuracy of the proposed model (Mul) and dominant class labeling model (Dom) with different region sizes: 16$\times$16 (sp16), 32$\times$32 (sp32), and 64$\times$64 (sp64).

to that of the dominant class labeling approach. The training process of our model consists of two stages and includes the generation of pseudo labels for five rounds. This process takes approximately 65 hours when executed on a single RTX3090 GPU.

C Further analysis of our framework

C.1 Effect of hyper-parameters

In Fig. 2, we evaluate the sensitivity of our stage 1 model to variations in the hyperparameters: ν, λ_{MP}, and λ_{CE}. This evaluation is conducted on Cityscapes using a ResNet50 backbone and combined with PixBal sampling. Our model demonstrates robustness to these hyperparameter changes, with accuracy fluctuations of less than 1.5%. It's noteworthy that our final model doesn't use the optimal hyperparameter values. This indicates that we didn't exhaustively tune these parameters using the validation set.

C.2 Effect of budget size

In Fig. 3(a), we evaluate the accuracy of our final model (Mul) under different budget sizes and round numbers, namely 50K over 10 rounds and 100K over 5 rounds. This analysis is performed on Cityscapes, using a ResNet50 backbone combined with PixBal sampling. As depicted in Fig. 3(a), when the total budget is kept constant, increasing the frequency of active sampling and model training enhances performance. This improvement can be attributed to the more frequent interactions between the model and the Oracle, leading to more informative active sampling.

C.3 Effect of region size

In Fig. 3(b), we evaluate the accuracy of the proposed model (Mul) and the dominant class labeling baseline (Dom) across different region sizes: 16$\times$16 (sp16), 32$\times$32 (sp32), and 64$\times$64 (sp64). Both labeling methods achieve their best performance with the 32$\times$32 region size. However, when the region size increases from 32$\times$32 to 64$\times$64, the dominant class labeling model suffers a significant performance drop due to increased label noise. In contrast, our proposed model with a region size of 64$\times$64 shows a smaller decrease in performance from its 32$\times$32 counterpart. This suggests that the multi-class labeling effectively mitigates label noise introduced by dominant class labeling.

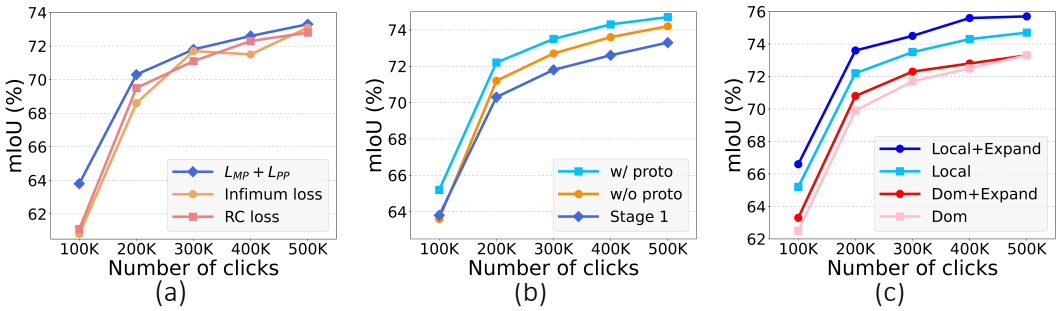


Figure 4: Accuracy in mIoU (%) versus the number of clicks (budget) evaluated on Cityscapes using ResNet50 backbone, combined with PixBal sampling. (a) The accuracy of our stage 1 model trained with proposed losses ($\mathcal{L}_{\text{MP}} + \mathcal{L}_{\text{PP}}$), compared with baseline partial label learning losses: Infimum loss [1], and RC loss [4, 6]. (b) The accuracy of stage 1 multi-class labeling model (stage 1), stage 2 multi-class labeling model solely employing intra-region label localization (w/ proto), and a baseline pseudo labeling method without prototype (w/o proto). (c) The accuracy of stage 2 multi-class labeling model with (Local+Expand) and without (Local) label expansion, compared with dominant class labeling model with (Dom+Expand) and without (Dom) label expansion.

D Additional results

D.1 Comparison with partial label learning loss baselines

In Fig. 4(a), we compare our proposed losses with baseline partial label learning losses: Infimum loss [1], and RC loss [4, 6]. This comparison is performed on Cityscapes, using a ResNet50 backbone combined with PixBal sampling. As shown in Fig. 4(a), the model with proposed losses ($\mathcal{L}_{\text{MP}} + \mathcal{L}_{\text{PP}}$) demonstrates superior performance over the models using the baseline losses, particularly in the early rounds.

D.2 Comparison with pseudo labeling baselines

In Fig. 4(b), we conduct an ablation study comparing our proposed intra-region label localization method (denoted as 'w/ proto') with a baseline intra-region pseudo labeling method that assigns the most confident class among multi-class labels as the pixel-wise pseudo label (denoted as 'w/o proto'). This comparison takes place on the Cityscapes dataset, utilizing a ResNet50 backbone in conjunction with PixBal sampling. While both of the intra-region pseudo-labeling methods improve upon the stage 1 model, the proposed prototype-based label localization demonstrates superior performance over the baseline, specifically in the initial round, where the stage 1 model may lack accuracy.

In Fig. 4(c), we compare the performance improvement brought by label expansion when applied to both the multi-class labeling model (denoted as 'Local+Expand') and the dominant class labeling model (denoted as 'Dom+Expand'). This comparison is also conducted on the Cityscapes dataset, using a ResNet50 backbone paired with PixBal sampling. As shown in Fig. 4(c), label expansion proves to be more beneficial when used with multi-class labeling, as it allows the spread of pseudo labels across multiple classes, resulting in a broader expansion of pseudo labels.

D.3 Qualitative results of our final model

Fig. 5 provides a qualitative result of the predictions of our final model at different rounds. As illustrated in Fig. 5, the quality of the predictions markedly improves as the rounds progress. Notably, the predictions produced by our final model at round 5 exhibit impressive quality, especially when taking into account that it requires only 9.8% of the labeling cost associated with a fully supervised model.

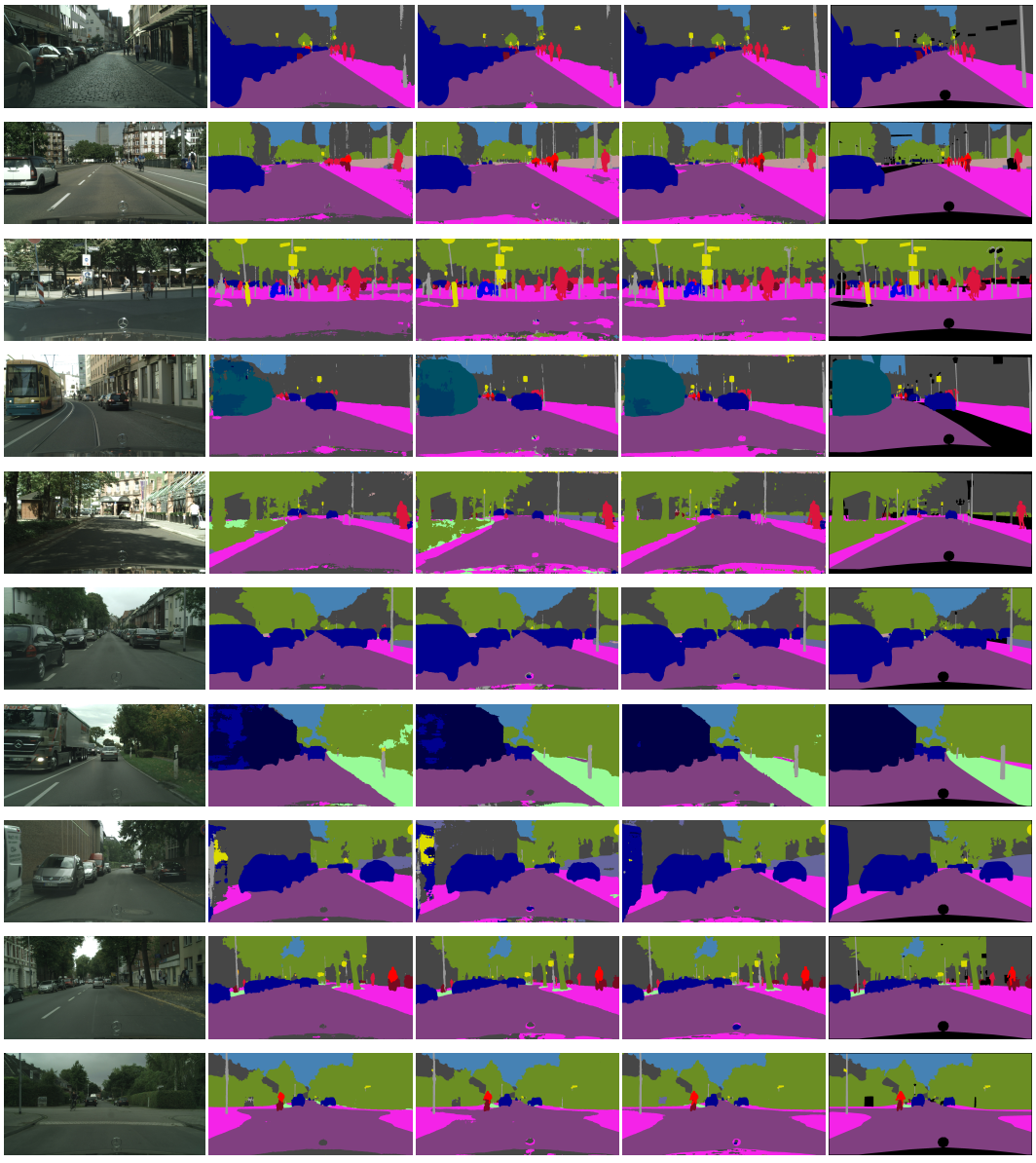


Figure 5: Qualitative results of our final model on the Cityscapes dataset. (a) Inputs. (b) Prediction in round 1. (c) Prediction in round 3. (d) Prediction in round 5. (e) Ground Truth.

References

[1] Vivien Cabannnes, Alessandro Rudi, and Francis Bach. Structured prediction with partial labelling through the infimum loss. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

[2] Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[3] 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. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in neural information processing systems*, 33:10948–10960, 2020.

[5] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019.

[6] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

[7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *AutoDiff, NIPS Workshop*, 2017.

[8] Tsung-Han Wu, Yi-Syuan Liou, Shao-Ji Yuan, Hsin-Ying Lee, Tung-I Chen, Kuan-Chih Huang, and Winston H Hsu. D2ada: Dynamic density-aware active domain adaptation for semantic segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, 2022.