You Never Get a Second Chance To Make a Good First Impression: Seeding Active Learning for 3D Semantic Segmentation (*Supplementary Material*)

Nermin Samet[1], Oriane Siméoni[2], Gilles Puy[2], Georgy Ponimatkin[1], Renaud Marlet[1,2], Vincent Lepetit[1]
[1]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France [2]Valeo.ai, Paris, France

1. Supplementary material – Overview

This supplementary material is organized as follows.

- We explain our sparsification algorithm, which is used to eliminate redundant scenes in SemanticKITTI [2] (Section 2);
- We provide further details regarding model training for comparing active learning (AL) methods (Section 3);
- We present all the quantative results in a table obtained from all AL methods on S3DIS and SemanticKITTI (Section 4);
- Finally, we report the remaining results of our ablation experiments; (i) component analyses results obtained from AL methods rand, MC-Drop, S-conf, S-margin, S-ent and SegEnt, (ii) results obtained from additional 2D (MoCo-v3) and 3D (DepthContrast, SegContrast, ALSO) features for all AL methods (Section 5);

2. Sparsification of SemanticKITTI

SemanticKITTI consists of sequences of frames sampled at 10 Hz. Consequently, there is a high similarity between successive frames, which are thus somehow redundant. To address this issue and improve scalability, we use a greedy algorithm to sparsify the SemanticKITTI dataset.

For each sequence, we begin with the first frame, use it as reference, and calculate its similarity with subsequent frames. We then eliminate any subsequent frame whose similarity with the first frame is above a threshold. The first subsequent frame falling below the threshold is then itself used as a new reference, and the process continues for all frames in the sequence. With this simple sparsification, we increase the scalability of the dataset and reduce computational requirements for downstream processing.

The similarities are computed again using global DINO features for each frame. We set a threshold of 0.75 on the cosine similarity. Our algorithm reduces the size of SemanticKITTI by 95%.

3. Implementation and experiment details

Our AL seeding method (SeedAL) is implemented using PyTorch [5]. We run S3DIS [1] experiments on a single V100 GPU with a batch size of 4. We perform the training of segmentation networks for CoreSet, S-conf, S-margin, S-ent, MC-drop and SegEnt on SemanticKITTI using 2 A100 GPUs with a batch size of 32. The training of networks when using ReDAL, a region-based method, is about 5 times longer than when using scene-based methods because more point clouds need to be processed.

Running time includes: using a pretrained model to create the features (90 ms/image on a V100 GPU), clustering and sorting candidates (negligible time), and extracting the best ones within the budget by linear optimization (<1 min for S3DIS, <5 min for SemanticKITTI).

4. Quantitative Results

To make it easier to compare performance quantitatively across papers, we report in Table 1 the detailed quantitative results obtained from all AL methods on S3DIS and SemanticKITTI datasets. We compare our method SeedAL to the proposed baselines, random sets and also the random seed used in ReDAL's paper [8] to produce results, noted ReDAL's seed in the table.

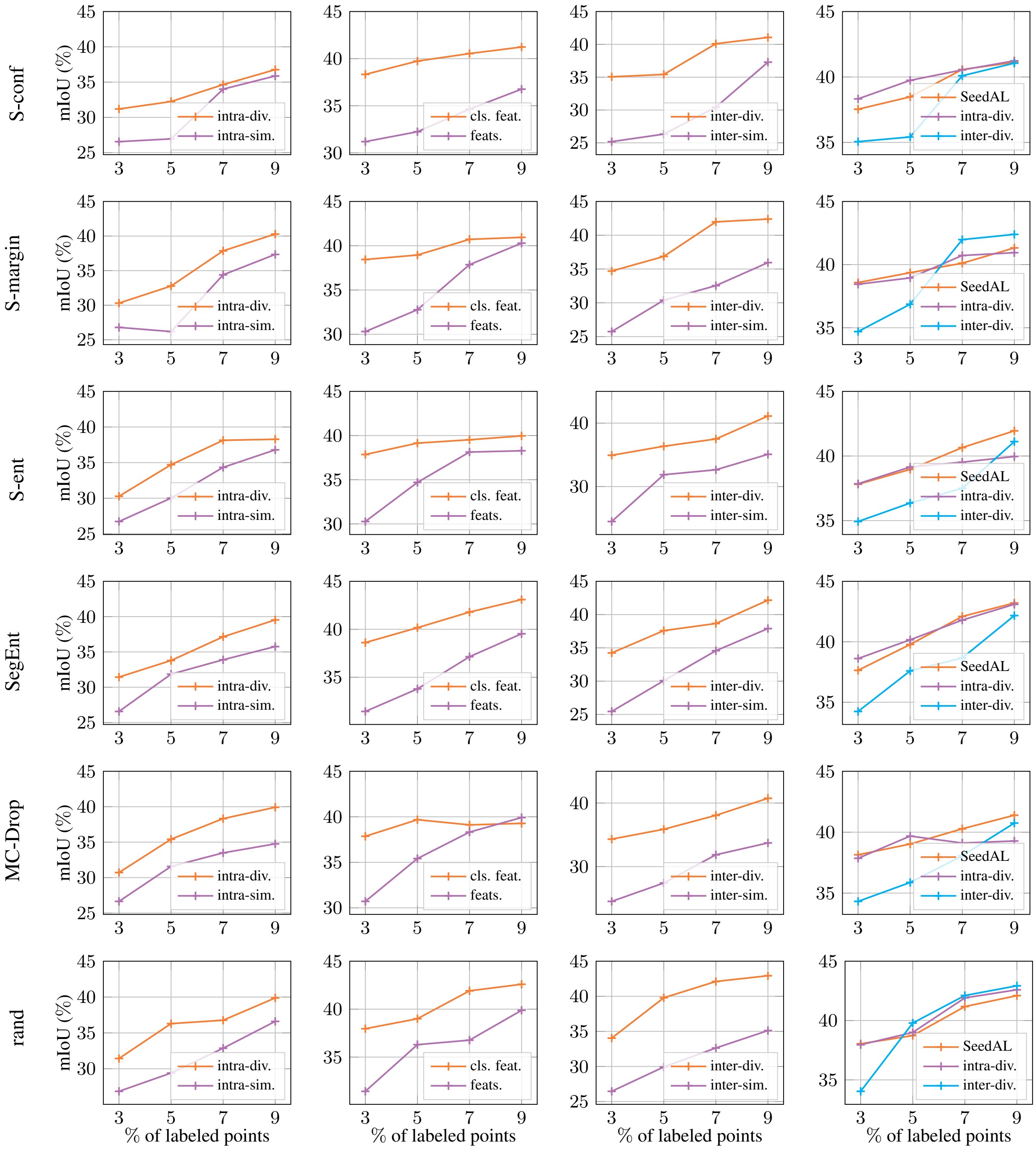
5. Ablation experiments

Figure 1 shows the remaining results of our ablation experiments obtained from AL methods rand, MC-Drop, S-conf, S-margin, S-ent and SegEnt. These results corroborate what is presented in Figure 7 and Section 5.4 of the paper, namely that: (a) intra-scene diversity is particularly relevant, compared intra-scene similarity; (b) clustering features leads to better AL seeds than just exploiting intra-scene diversity; (c) inter-scene diversity leads to better AL seeds than inter-scene similarity; (d) the proposed combination of intra- and inter-scene diversity (i.e., SeedAL) generally performs on par or better than both intra- or inter-scene diversity, independently.

As a complement to Figure 8 in the paper, Figure 2 and Figure 3 present the results for all active learning methods with different 2D and 3D self-supervised features on S3DIS and SemanticKITTI, respectively. We do not provide results with ReDAL on SemanticKITTI due to its massive training cost.

AL method	AL seeding method	S3DIS (% of labeled points)				SemanticKITTI (% of labeled points)			
		3	5	7	9	1	2	3	4
rand	random	30.1	35.3	38.5	40.4	46.1	50.8	53.9	55.9
	std dev	5.5	3.7	2.9	2.9	3.5	1.0	1.5	1.4
	KMcentroid	33.5	36.9	39.8	40.9	41.0	48.9	53.6	54.7
	KMfurthest	36.2	36.8	40.9	**42.8**	39.8	51.9	53.8	56.7
	ReDAL's seed	26.1	30.0	35.9	39.8	48.0	51.9	54.6	56.6
	SeedAL	**38.0**	**38.7**	**41.2**	42.1	**51.3**	**55.0**	**56.6**	**58.0**
S-conf [7]	random	30.3	33.1	35.6	37.9	46.1	48.2	49.9	52.6
	std dev	5.8	3.5	3.0	3.7	3.7	4.2	3.8	3.2
	KMcentroid	33.4	33.5	36.7	38.2	41.4	48.6	50.1	53.4
	KMfurthest	36.5	36.6	39.2	41.1	39.9	46.9	50.4	52.5
	ReDAL's seed	26.1	26.6	29.3	34.6	47.9	50.2	51.7	54.2
	SeedAL	**37.5**	**38.5**	**40.6**	**41.1**	**51.7**	**53.7**	**54.4**	**56.6**
S-margin [7]	random	30.1	33.1	34.9	36.9	45.6	48.3	50.1	51.6
	std dev	5.6	4.1	4.0	3.7	3.3	2.6	2.4	2.4
	KMcentroid	33.6	36.1	36.5	39.1	40.5	46.6	49.1	50.4
	KMfurthest	36.8	38.7	38.5	41.2	40.5	45.5	48.5	50.3
	ReDAL's seed	26.1	28.3	35.5	39.9	46.2	50.0	50.3	50.9
	SeedAL	**38.6**	**39.4**	**40.1**	**41.3**	**52.2**	**51.8**	**54.7**	**56.2**
S-ent [7]	random	29.6	32.3	34.9	37.2	45.7	47.9	49.8	52.1
	std dev	5.6	5.2	4.0	3.1	3.2	4.2	3.7	3.3
	KMcentroid	33.2	34.1	36.5	41.3	41.8	46.6	50.1	53.1
	KMfurthest	35.8	35.7	38.0	41.1	39.8	47.6	49.5	51.2
	ReDAL's seed	27.4	29.9	32.9	38.3	47.4	50.0	50.1	51.8
	SeedAL	**37.8**	**39.0**	**40.7**	**41.9**	**52.4**	**53.0**	**55.2**	**56.8**
CoreSet [6]	random	30.1	33.6	36.2	37.5	45.3	49.5	53.5	55.1
	std dev	5.5	4.2	3.4	2	3.7	2.6	1.4	1.0
	KMcentroid	33.5	37.2	36.6	39.2	40.6	46.6	52.5	54.5
	KMfurthest	36.4	39.3	**41.1**	39.7	40.1	46.4	50.2	54.7
	ReDAL's seed	26.3	30.2	32.3	34.9	46.4	49.5	52.1	54.1
	SeedAL	**37.7**	**40.1**	40.9	**41.9**	**52.1**	**53.8**	**55.2**	**56.9**
MC-Drop [3]	random	30.4	32.9	35.3	37.3	46.4	48.2	50.3	52.1
	std dev	5.5	4.3	3.0	3.3	3.3	5.2	4.5	4.4
	KMcentroid	33.5	33.6	37.5	39.7	40.9	48.3	50.7	52.3
	KMfurthest	37.1	37.4	38.9	**42.4**	39.4	43.5	49.0	51.7
	ReDAL's seed	26.9	28.9	31.5	32.1	48.6	50.6	52.4	54.2
	SeedAL	**38.1**	**39.0**	**40.3**	41.4	**50.4**	**53.4**	**53.6**	**55.6**
SegEnt [4]	random	30.2	33.6	38.2	39.8	45.4	50.6	52.4	54.2
	std dev	5.5	2.5	1.9	2.1	3.8	1.5	0.4	0.8
	KMcentroid	33.7	34.3	35.9	38.0	41.2	50.4	52.1	53.9
	KMfurthest	35.9	38.2	38.8	40.5	40.1	49.8	51.3	54.0
	SeedAL	**37.6**	**39.8**	**42.1**	**43.2**	**51.1**	**52.4**	**55.0**	**55.5**
ReDAL [8]	random	30.7	38.6	44.3	49.4	44.9	51.8	55.8	57.9
	std dev	5.2	1.0	0.5	0.7	3.2	1.8	0.8	0.6
	KMcentroid	32.6	37.6	45.3	48.3	38.5	**53.9**	55.7	57.2
	KMfurthest	35.1	41.5	47.5	**51.7**	38.3	48.6	57.0	58.6
	ReDAL's seed	24.9	37.5	43.8	45.5	46.1	53.8	**56.7**	58.4
	SeedAL	**37.5**	**42.8**	**48.6**	**51.7**	**50.5**	**53.9**	55.8	**58.9**

Table 1: Performance (% mIoU) of the AL seeding methods on several AL methods for S3DIS and SemanticKITTI. Noted 'random' is the average over three and six random seeds for S3DIS and SemanticKITTI respectively (we also report the standard deviation "std dev"). "ReDAL's seed" is the random seed used in the experiments reported in ReDAL's paper [8]. We report the results for the ReDAL method obtained after our re-training.



(a) (b) (c) (d)

Figure 1: *Ablation study.* [Complement to Figure 7 in the paper] We evaluate here results obtained with different seeding strategies. (a) Seeds made of scenes with high intra-diversity (intra-div.) or high intra-similarity (intra-sim.). (b) Seeds selected with two different intra-diversity metrics:view features (feats.)or computed after clustering the view features (cls. feat.). (c) Seeds made of scenes with high inter-diversity (inter-div.) or high inter-similarity (inter-sim.). (d) Seeds selected with SeedAL, considering only inter-diversity (inter-div.) or intra-diversity (intra-div.).

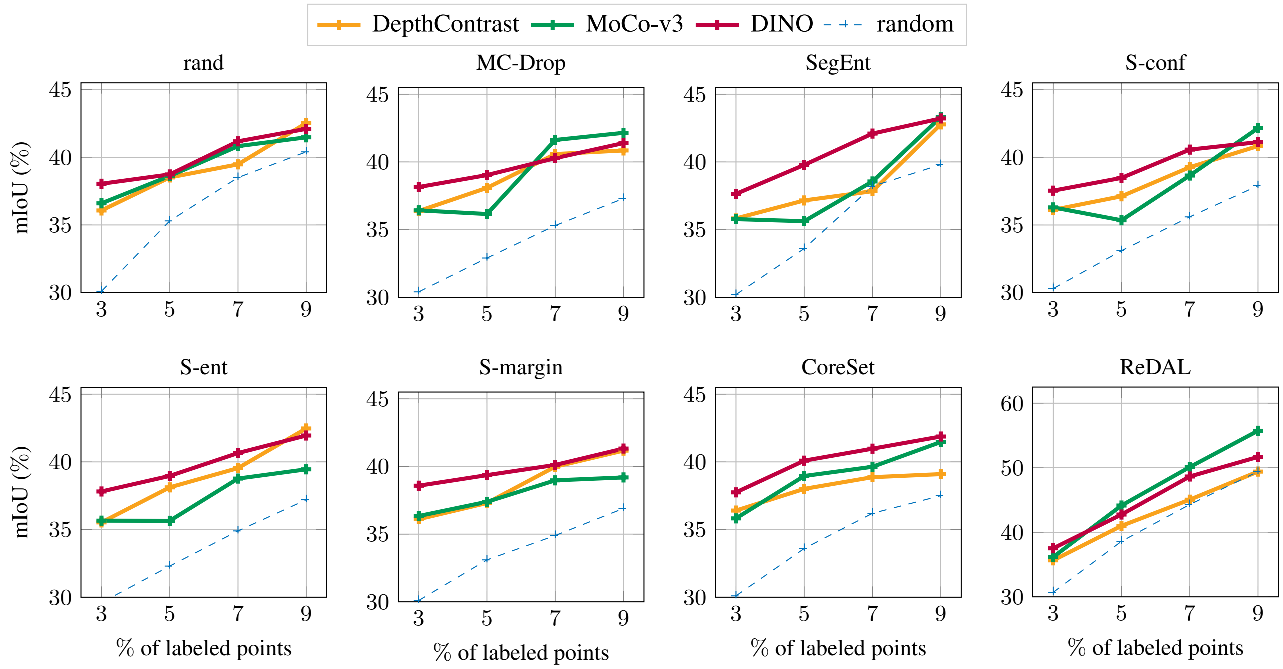


Figure 2: SeedAL results on S3DIS using features from MoCo-v3 and DINO. Rand is an average over the random seeds.

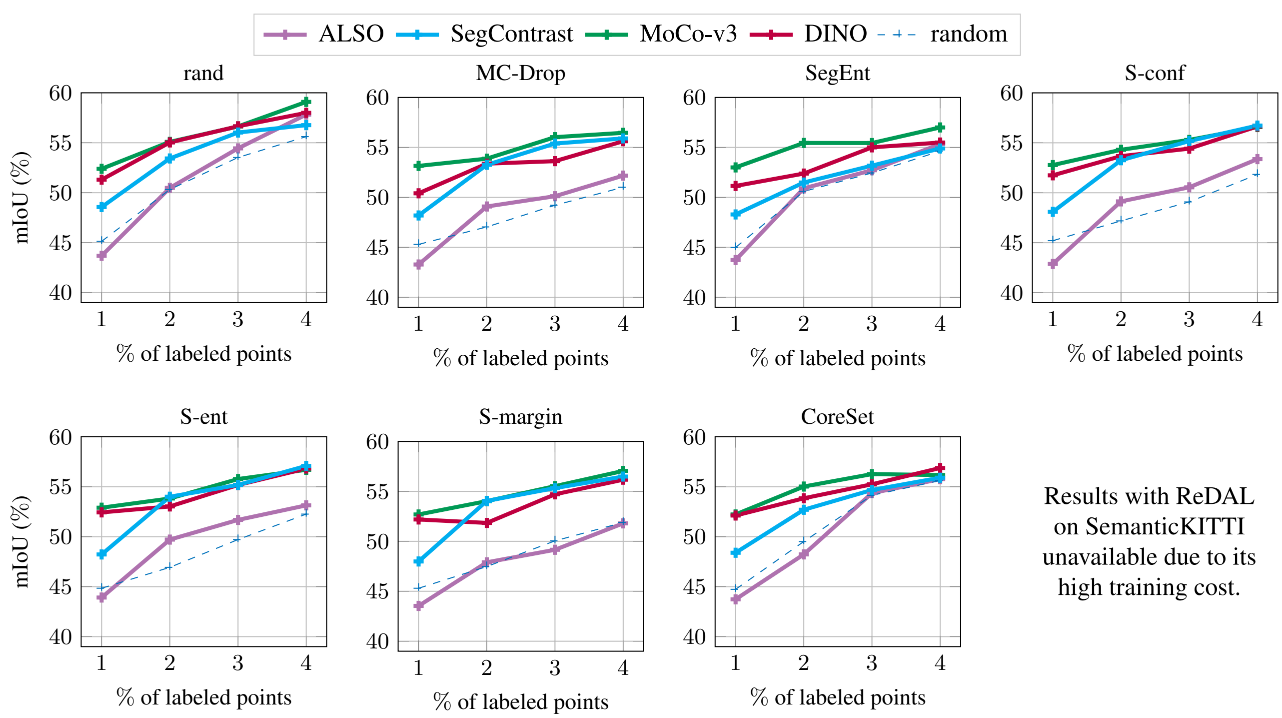


Figure 3: SeedAL results on SemanticKITTI using features from DepthContrast, SegContrast, ALSO, MoCo-v3, DINO. Rand is an average over the random seeds.

References

[1] Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3D Semantic Parsing of Large-Scale Indoor Spaces. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1

[2] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: International Conference on Computer Vision (ICCV). pp. 9297–9307 (2019) 1

[3] Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian Active Learning with Image Data. In: International Conference on Machine Learning. pp. 1183–1192 (2017) 2

[4] Lin, Y., Vosselman, G., Cao, Y., Yang, M.Y.: Efficient Training of Semantic Point Cloud Segmentation via Active Learning. SCIENCE **2**, 243–250 (2020) 2

[5] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019) 1

[6] Sener, O., Savarese, S.: Active Learning for Convolutional Neural Networks: A Core-Set Approach. In: International Conference for Learning Representations (ICLR) (2018) 2

[7] Wang, D., Shang, Y.: A New Active Labeling Method for Deep Learning. In: International Joint Conference on Neural Networks. pp. 112–119 (2014) 2

[8] Wu, T.H., Liu, Y.C., Huang, Y.K., Lee, H.Y., Su, H.T., Huang, P.C., Hsu, W.H.: ReDAL: Region-Based and Diversity-Aware Active Learning for Point Cloud Semantic Segmentation. In: International Conference on Computer Vision (ICCV). pp. 15510–15519 (2021) 1, 2