








# *Supplementary Material* for LiDAL: Inter-frame Uncertainty Based Active Learning for 3D LiDAR Semantic Segmentation

Zeyu Hu<sup>1</sup><sup>\*</sup>, Xuyang Bai<sup>1</sup>, Runze Zhang<sup>2</sup>, Xin Wang<sup>2</sup>, Guangyuan Sun<sup>2</sup>, Hongbo Fu<sup>3</sup>, and Chiew-Lan Tai<sup>1</sup>

<sup>1</sup> Hong Kong University of Science and Technology  
{zhuam,xbaiad,taicl}@cse.ust.hk

<sup>2</sup> Lightspeed & Quantum Studios, Tencent  
{ryanrzzhang,alexinwang,gerrysun}@tencent.com

<sup>3</sup> City University of Hong Kong  
hongbofu@cityu.edu.hk

**Abstract.** This supplementary document is organized as follows:

- Section 1 explains in more detail about the LiDAL implementation.
- Section 2 describes the baseline active learning methods.
- Section 3 enumerates detailed semantic segmentation results of the line charts in the main paper.
- Section 4 provides more ablation studies.

## 1 Implementation Details

As explained in Section 3.1 of the main paper, our LiDAL method consists of four steps: 1. Train the network to convergence using the currently labeled dataset  $D_L$ . 2. Calculate the model uncertainty scores for each region of  $D_U$ . 3. Select regions based on the uncertainty measures for active learning and self-training. 4. Obtain labels from human annotation and pseudo-labeling. In this section, we supplement implementation details to these steps. Note that the used symbols are the same as those in Section 3 of the main paper.

### 1.1 Network Training

All the experiments are conducted on a PC with 8 NVIDIA Tesla V100 GPUs. The batch sizes are set to 30 and 90 for the SemanticKITTI [1] and nuScenes [4] datasets, respectively.

For both datasets, we train the networks by minimizing the cross-entropy loss using Adam optimizer with an initial learning rate 1e-3. For fully-supervised baselines, the networks are trained for 80,000 iterations. For each round of active learning (including the initial round), the networks are trained or fine-tuned for 20,000 iterations.

---

\* intern at Tencent Lightspeed & Quantum Studios



Fig. 1: **An example of divided sub-scene regions in the SemanticKITTI dataset.** Points of the same regions are painted with the same colors.

The training settings are the same for SPVCNN [15] and MinkowskiNet [5] network architectures. On the ScanNet dataset, for SPVCNN, LiDAL consumes about 19 GPU hours for inference (uncertainty scoring) and 40 GPU hours for training in each round. For MinkowskiNet, LiDAL consumes about 18 GPU hours for inference and 34 GPU hours for training. On the nuScenes dataset, for SPVCNN, LiDAL consumes about 10 and 31 GPU hours in each round, respectively. For MinkowskiNet, LiDAL consumes about 8 and 27 GPU hours, respectively.

## 1.2 Correspondence Estimation

In Section 3.2 of the main paper, after the registration of each frame, we then find for each point its corresponding points in the neighboring frames to calculate inter-frame uncertainty measures. Since there are hundreds of thousands of points in a LiDAR frame, it is impractical to register all the LiDAR frames at the same time and then estimate correspondences for each point.

To address this issue, for each frame  $F_i$ , we retrieve its neighboring  $N_{nei}$  frames for correspondence estimation. After registration, for each point  $p$  of the frame  $F_i$ , we find its nearest point in each of the neighboring  $N_{nei}$  frames as the initial corresponding points. Since a certain position may be scanned in not all the frames due to occlusion and the movement of the scanning device, point  $p$  may not have proper corresponding points in some neighboring frames. We then filter out the corresponding points whose distances to  $p$  are larger than a threshold  $T_p$ . For both the SemanticKITTI and nuScenes datasets, we set  $N_{nei} = 24$  and  $T_p = 0.1m$ .

### 1.3 Region Division and Overlap Determining

We utilize the constrained K-means clustering [2] algorithm to divide a LiDAR frame  $F$  into multiple sub-scene regions. As an extension of the classical K-means algorithm, this algorithm forces the number of points in each of the  $K$  clusters in  $(N_{min}, N_{max})$ . For both the SemanticKITTI and nuScenes datasets, we set  $K = 20$ ,  $N_{min} = 0.95 * \frac{|F|}{K}$ , and  $N_{max} = 1.05 * \frac{|F|}{K}$ , where  $|F|$  is the number of points contained in frame  $F$ . An example of divided sub-scene regions of the SemanticKITTI dataset is shown in Fig 1.

In Section 3.3 of the main paper, for a specific region  $r$ , we need to retrieve the set of regions overlapping with  $r$  for further processing. To determine if two regions overlap, we may check the Earth Mover’s distance [9] or the Chamfer distance [3] between the two regions. However, we find that a simpler solution based on the distance between the weight centers of two regions yields similar results. Considering the efficiency of this simple solution, we determine that two regions overlap if the distance between their weight centers is less than  $T_r$ . For both the SemanticKITTI and nuScenes datasets, we set  $T_r = 5m$ .

### 1.4 Label Acquisition

For active learning, instead of using a human annotator, we simulate annotation by using the ground-truth annotation of the dataset as the annotation from a human annotator. For self-training, we use the network predictions averaged over 8 augmented inference runs as the pseudo-labels.

## 2 Baseline Active Learning Methods

In this section, we describe the implementation of the baseline active learning methods used in our experiments (Section 4.2 of the main paper).

**Random Selection (RAND<sub>fr</sub> and RAND<sub>re</sub>).** In each round of active learning, this baseline method randomly selects a portion of LiDAR frames or point cloud regions from the unlabeled dataset for label acquisition. It is a commonly used baseline strategy in the literature [17,14,12,6].

**Segment-entropy (SEGENT).** Based on the assumption that points within a region are supposed to share the same label, segment-entropy is proposed to serve as a metric for active selection [10]. In this method, the distribution of predicted labels within a region  $r$  is estimated by:

$$E_{seg} = - \sum_c q(c) \log q(c), \quad (1)$$

$$\hat{y}^p = \arg \max_c P^p, \quad (2)$$

$$q(c) = \frac{1}{|r|} \sum_{p \in r} f(\hat{y}^p, c), \quad (3)$$

$$f(\hat{y}^p, c) = \begin{cases} 1, & \text{if } \hat{y}^p = c \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where  $E_{seg}$  is the proposed segment-entropy,  $P^p$  is the probability distribution of point  $p$ ,  $\hat{y}^p$  is the predicted label of point  $p$ , and  $q(c)$  is the percentage of points predicted as class  $c$ . The segment-entropy score of a frame is the average of the scores of all the points inside this frame. The frames with the largest segment-entropy scores are selected for label acquisition. In the implementation of this method, we utilize the same region division results as our LiDAL for a fair comparison.

**Softmax Margin (MAR).** Some previous active learning methods [8,11,16] rank all the samples in order of the model decision margin, which is the difference of softmax probabilities between the most probable label and the second most probable label, and then select the samples with the least differences. For a point  $p$ , the softmax margin is calculated as:

$$MAR^p = P^p(\hat{y}^1) - P^p(\hat{y}^2), \quad (5)$$

where  $P^p$  is the probability distribution of point  $p$ ,  $\hat{y}^1$  is the most probable label class, and  $\hat{y}^2$  is the second most probable label class.

The softmax margin of a frame is calculated by averaging the values of all the points inside it. The frames with the least softmax margin values are selected for label acquisition.

**Softmax Confidence (CONF).** Similar to MAR, the softmax probability of the most probable label is considered as a confidence score in some previous methods [13,16]. For point  $p$ , the softmax confidence is calculated as:

$$CONF^p = P^p(\hat{y}^1), \quad (6)$$

where  $P^p$  is the probability distribution of point  $p$ , and  $\hat{y}^1$  is the most probable label class.

For a frame, the softmax confidence score is the average result of the scores of all the associated points. The frames with the least confidence scores are selected for label acquisition.

**Softmax Entropy (ENT).** Unlike MAR and CONF, which consider only the top two most probable classes, softmax entropy takes into account probabilities of all classes to measure the information of a probability distribution [7,16]. For point  $p$ , the softmax entropy score is calculated as:

$$ENT^p = - \sum_c P^p(c) \log(P^p(c)), \quad (7)$$

where  $P^p(c)$  is the probability of point  $p$  belonging to class  $c$ .

For a frame, the softmax entropy score is the average result of the scores of all the associated points. The frames with the largest entropy scores are selected for label acquisition.

**Core-set Selection (CSET).** Core-set refers to a small subset that captures the diversity of the whole dataset [12], and thus a model trained on this subset yields similar performance to that trained on the whole dataset. This method first extracts features for each sample of the dataset using the currently trained network. Operating on the feature space, it then selects a small set of samples for labeling utilizing the furthest point sampling strategy. In the implementation, we use the intermediate results of the second-last layers of the networks as the features. The feature of a frame is averaged over all the associated points.

**ReDAL.** Region-based and diversity-aware active learning (ReDAL) [17] is a recent state-of-the-art method designed for 3D semantic segmentation of both indoor and outdoor scenes. This method first divides a 3D scene into sub-scene regions and then estimates the region information utilizing three metrics: softmax entropy, color discontinuity, and structural complexity. With the estimated region information scores, this method further designs a diversity-aware selection algorithm to avoid visually similar regions appearing in a querying batch for labeling. Since both the SemanticKITTI and nuScenes datasets do not provide colored point clouds, the color discontinuity metric is discarded in the implementation following the instruction of ReDAL’s official code.

### 3 Detailed Experimental Results

In this section, we provide more details on our experimental results, for benchmarking purposes with future works. The results of fully-supervised networks are reported in Table 1. Detailed scores for Fig. 5 in the main paper are shown in Tables 2 and 3. For Fig. 6, the detailed scores are presented in Tables 4 and 5.

Table 1: **Mean intersection over union scores of fully-supervised networks.**

Network \ Dataset	SemanticKITTI	nuScenes
SPVCNN	64.5	71.7
MinkowskiNet	61.4	70.6

Table 2: Mean intersection over union scores on SemanticKITTI Val with SPVCNN.

Percentage of Labeled Points	Init (1%)	2%	3%	4%	5%
RAND <sub>fr</sub>	48.8	52.1	53.6	55.6	57.2
RAND <sub>re</sub>	48.8	51.7	55.0	56.1	58.2
SEGENT	48.8	49.8	48.3	49.1	48.2
MAR	48.8	49.4	50.0	48.7	49.3
CONF	48.8	48.0	48.9	50.4	51.6
ENT	48.8	49.6	48.5	50.1	49.9
CSET	48.8	53.1	52.9	53.2	52.6
ReDAL <sub>reported</sub>	41.9	51.7	55.8	56.9	58.2
ReDAL <sub>retrained</sub>	48.8	51.3	54.0	58.6	58.1
LiDAL (ours)	48.8	57.1	58.7	59.3	59.5

Table 3: Mean intersection over union scores on SemanticKITTI Val with MinkowskiNet.

Percentage of Labeled Points	Init (1%)	2%	3%	4%	5%
RAND <sub>fr</sub>	47.3	51.4	55.8	57.7	56.6
RAND <sub>re</sub>	47.3	50.1	55.8	55.9	58.5
SEGENT	47.3	49.8	48.8	49.5	47.7
MAR	47.3	50.2	49.8	49.4	50.1
CONF	47.3	48.5	48.5	51.4	51.7
ENT	47.3	49.9	48.8	49.0	50.2
CSET	47.3	52.6	55.9	56.4	57.6
ReDAL <sub>reported</sub>	37.5	48.9	55.3	58.4	59.8
ReDAL <sub>retrained</sub>	47.3	51.4	52.5	58.4	58.1
LiDAL (ours)	47.3	56.7	58.7	59.5	60.1

Table 4: Mean intersection over union scores on nuScenes Val with SPVCNN.

Percentage of Labeled Points	Init (1%)	2%	3%	4%	5%
RAND <sub>fr</sub>	51.8	58.4	60.5	60.6	63.2
RAND <sub>re</sub>	51.8	60.3	62.3	63.7	63.6
SEGENT	51.8	55.5	56.1	55	57.8
MAR	51.8	55.2	56.4	57.0	57.7
CONF	51.8	55.1	54.9	55.4	56.0
ENT	51.8	55.4	56.7	56.6	57.2
CSET	51.8	59.4	62.3	62.9	63.0
ReDAL	51.8	54.3	57.0	57.2	58.3
LiDAL (ours)	51.8	60.8	65.6	67.6	68.2

Table 5: Mean intersection over union scores on nuScenes Val with MinkowskiNet.

Percentage of Labeled Points	Init (1%)	2%	3%	4%	5%
RAND <sub>fr</sub>	49.7	57.9	60.5	61.8	61.7
RAND <sub>re</sub>	49.7	58.7	60.9	62.0	63.1
SEGENT	49.7	54.8	55.3	56.5	58.5
MAR	49.7	53.9	55.0	56.7	59.1
CONF	49.7	54.4	55.7	56.8	55.5
ENT	49.7	54.9	56.4	57.2	57.6
CSET	49.7	58.5	62.0	63.2	63.6
ReDAL	49.7	54.5	53.9	56.7	57.2
LiDAL (ours)	49.7	62.3	64.7	66.5	67.0

## 4 More Ablation Studies

In this section, we provide more ablation studies to examine the design decision of our self-training strategy and to analyse the actively selected labels and pseudo-labels. We also evaluate the effects of data augmentation and K-means clustering techniques used in our method. Moreover, we showcase the overall trend of performance given more labeling budget.

**Self-training Strategy.** In Section 3.3 of the main paper, we inject pseudo-labels to the training set at each active learning round to further boost the performance. We have considered three commonly used strategies for self-training:

- **S1:** Enlarge the pseudo-label set in each round with the newly selected regions. (The selection criterion is discussed in the main paper.)
- **S2:** Keep the size of the pseudo-label set constant, and replace in each round with the newly selected regions.
- **S3:** (Our design choice) Keep the size of the pseudo-label set constant, and replace in each round with the newly selected regions that are not already in the last pseudo-label set.

The results of these three self-training strategies on the SemanticKITTI dataset with SPVCNN are shown in Table 6. As shown in the table, both two alternative strategies generate more inferior results to our design choice. We assume that, for **S1**, it is easily susceptible to label drifting as its size of pseudo-label set increases over time. For **S2**, since the previous pseudo-label set used for training is also considered for the pseudo-labeling of the current round, it tends to select a stable set of regions that are less and less helpful during training.

**Class Distribution of Actively Selected Samples.** To gain a better understanding of the property of inter-frame constraints, we count the class distribution of samples selected in all 4 rounds of LiDAL operating on the SemanticKITTI dataset with SPVCNN network. As shown in Table 7, LiDAL

Table 6: Mean intersection over union scores of different self-training strategies on SemanticKITTI Val with SPVCNN.

Percentage of Labeled Points	Init (1%)	2%	3%	4%	5%
<b>S1</b>	48.8	56.9	57.2	59.1	58.8
<b>S2</b>	48.8	57.2	58.5	58.9	59.0
<b>S3</b> (Our design choice)	48.8	57.1	58.7	59.3	59.5

focuses more on less-represented but highly important classes like person and bicyclist. This is foreseeable since the networks struggle to generate consistent predictions for these hard samples. This is a valuable property that can benefit downstream tasks like autonomous driving, which poses great significance on safety issues.

Table 7: Class distributions of labels(%). We present samples selected in all 4 rounds of LiDAL operating on the SemanticKITTI dataset with SPVCNN network.

Method	Total	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
Full	$10^3$	43.68	0.17	0.42	2.02	2.40	0.36	0.13	0.04	205.22	15.19	148.59	4.03	137.00	74.69	275.57	6.23	80.67	2.95	0.63
LiDAL	$10^3$	36.75	0.25	1.06	4.01	7.45	0.89	0.39	0.07	146.42	22.30	154.42	11.91	127.15	98.29	277.80	9.01	95.91	4.34	1.57

**Accuracy of Pseudo-labels.** The main challenge with pseudo-labels is to ensure their accuracy and to avoid drifting. In Section 4.4 of the main paper, we evaluate the effects of injecting different numbers of pseudo-labels into the training set. Here we quantitatively measure the accuracy of added pseudo-labels in Table 8. The study is conducted in the first training round of SPVCNN on the SemanticKITTI dataset. As shown in the table, the generated pseudo-labels maintain high accuracy in general, but the accuracy drops when more and more pseudo-labels are selected. This confirms our conjecture in the main paper that adding a reasonable number of pseudo-labels will improve network performance, but redundant pseudo-labels might introduce unhelpful training bias and label noise.

**Data Augmentation.** For 3D semantic segmentation, commonly used data augmentation techniques include random scaling, rotation around the gravity axis, spatial translation, spatial elastic distortion, chromatic translation, point jittering, and context mixing. These techniques can significantly increase the



Table 8: **Accuracy of pseudo-labels.** Samples are selected in the first training round of SPVCNN on SemanticKITTI dataset.

Range of Added Pseudo-labels	Mean Accuracy
0-1%	97.58%
1-2%	97.04%
2-3%	93.05%

amount of training data to regularize the convergence of models and improve their performance. However, they require a large amount of fully labeled data and rely heavily on domain knowledge.

In Section 3.2 (L202-212) of the main paper, we perform data augmentations to attain robust probability predictions for uncertainty scoring. With the same setting as in Section 4.4, we turn off the data augmentation to investigate its effect. As shown in Table 9, the removal of data augmentation results in a slight performance drop of 0.2%.

Table 9: **Ablation study: Data augmentation.**

Number of Augmented Inference Runs	mIoU (%)
1 (no augmentation)	56.9
8 (used in the paper)	57.1

**K-means Clustering.** In LiDAL, to construct region-based query units, we utilize the constrained K-means clustering algorithm for its simplicity and efficiency. The clustering is performed on the 3D point positions alone and a visual example is provided in Supplementary Fig. 1. With the same setting as in Section 4.4, we vary the number of clusters to measure the effect of clustering size. As shown in Table 10, when the number of clusters is too small ( $K = 10$ ), the performance drops due to the limited context range and the colossal number of uninformative points in the selected large regions. When  $K$  is too large ( $K = 40, 80$ ), the performance drops as well. It is possibly because of the unstable gradient flows from the labeled points constituting only a small fraction of the input points of each training batch. Since LiDAL is orthogonal to the used query unit, it can easily benefit from the improvements of region division methods.

Table 10: **Ablation study: K-means clustering.**

Number of Clusters	10	20 (used in the paper)	40	80
mIoU (%)	56.4	57.1	57.0	56.5

**More Training Budget.** To demonstrate the overall trend of performance given more labeling budget, we further perform three rounds of active learning process based on the results in Fig. 5 of the paper. For fast iteration, we drop the self-training part and reduce the training iterations of each round to 5K. As shown in Fig. 2, with more than 5% of labeled data, the performance of the network tends to saturate. It may be caused by the biases introduced in the initial round (e.g., consistent mis-predictions for objects of certain classes).

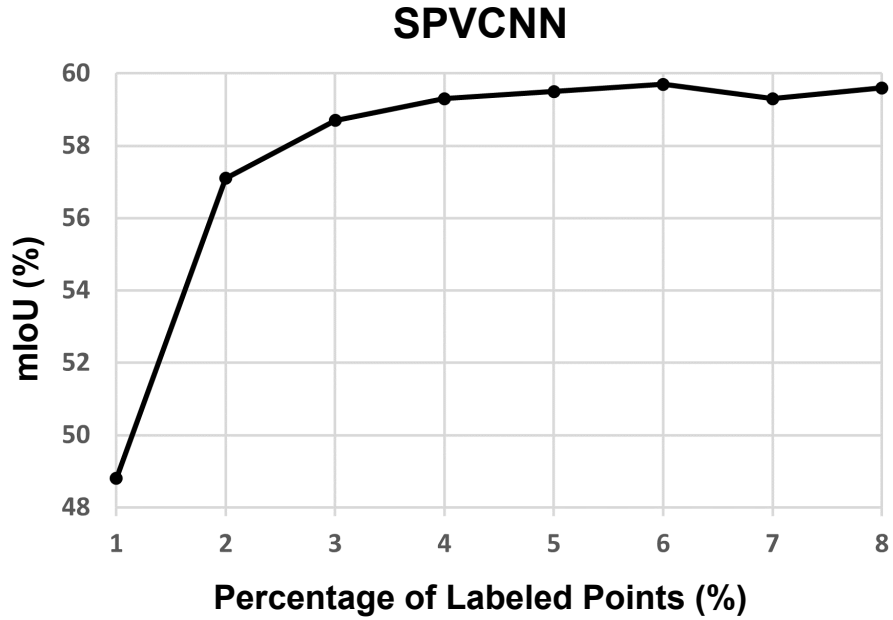


Fig. 2: Mean IoU scores on SemanticKITTI Val.

## References

1. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9297–9307 (2019)
2. Bradley, P.S., Bennett, K.P., Demiriz, A.: Constrained k-means clustering. Microsoft Research, Redmond **20**(0), 0 (2000)
3. Butt, M.A., Maragos, P.: Optimum design of chamfer distance transforms. IEEE Transactions on Image Processing **7**(10), 1477–1484 (1998)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous

- driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
5. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
  6. Feng, Q., He, K., Wen, H., Keskin, C., Ye, Y.: Active learning with pseudo-labels for multi-view 3d pose estimation. arXiv preprint arXiv:2112.13709 (2021)
  7. Hwa, R.: Sample selection for statistical parsing. *Computational linguistics* **30**(3), 253–276 (2004)
  8. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 2372–2379. IEEE (2009)
  9. Levina, E., Bickel, P.: The earth mover’s distance is the mallows distance: Some insights from statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 251–256. IEEE (2001)
  10. Lin, Y., Vosselman, G., Cao, Y., Yang, M.: Efficient training of semantic point cloud segmentation via active learning. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* **5**(2) (2020)
  11. Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: European Conference on Machine Learning. pp. 413–424. Springer (2006)
  12. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489 (2017)
  13. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: proceedings of the 2008 conference on empirical methods in natural language processing. pp. 1070–1079 (2008)
  14. Siddiqui, Y., Valentin, J., Nießner, M.: Viewal: Active learning with viewpoint entropy for semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9433–9443 (2020)
  15. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: European conference on computer vision. pp. 685–702. Springer (2020)
  16. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology* **27**(12), 2591–2600 (2016)
  17. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15510–15519 (2021)