# LESS: Label-Efficient Semantic Segmentation for LiDAR Point Clouds

Minghua Liu[1⋆], Yin Zhou[2⋆⋆], Charles R. Qi[2], Boqing Gong[3], Hao Su[1], and Dragomir Anguelov[1]

[1]UC San Diego, [2]Waymo, [3]Google

In this supplementary material, we first present the implementation and training details of our proposed method and baseline methods (Sec. S.1). We then show the visual examples of our pre-segmentation results (Sec. S.2), the full results on the nuScenes dataset (Sec. S.3), and the multi-scan distillation results on the SemanticKITTI dataset (Sec. S.4). Finally, we analyze the generated label distribution (Sec. S.5) and the robustness to label noise (Sec. S.6).

## S.1 Implementation & training details

**Pre-segmentation & labeling** While some prior works require perfect pre-segmentation results, our proposed labeling and training pipeline (using weak and propagated labels) allows imperfect component proposals (e.g., a component with multiple categories or an object instance divided into multiple components), which greatly mitigates the impact of pre-segmentation quality on final performance. Our pre-segmentation heuristic only includes two key steps: ground removal and connected component construction. Compared to other complex heuristics, it has fewer hyperparameters. Also, thanks to the good property of outdoor point clouds (i.e., objects are well-separated), we find that, in our experiments, the hyper-parameters are intuitive and easy to select without much effort.

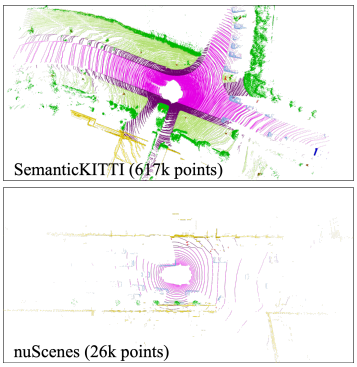
For example, during the ground removal, we find that the cell size and the RANSAC threshold are robust across datasets, and we set them to be $5m \times 5m$ and $0.2m$ for both datasets. When building connected components, the parameter $d$ should accommodate the LiDAR sensor (the sparser the points, the larger the $d$). We set $d$ to 0.01 and 0.02 for SemanticKITTI [1] and nuScenes [2] datasets, respectively. In our experiments, choosing hyper-parameters with visual inspection is convenient and sufficient to achieve satisfactory results.

For the SemanticKITTI [1] dataset, we fuse every 5 adjacent scans for the 0.1% setting and every 100 adjacent scans for the 0.01% setting. Fusing more adjacent scans will improve labeling efficiency, but may sacrifice pre-segmentation quality as points may become blurry, especially for dynamic objects. After constructing connected components, oversized components are subdivided along the $xy$ axes to ensure each component is within a fixed size (i.e., $2m \times 2m$ for non-ground components). We also ignore small components with no more than 100

⋆ Work done during internship at Waymo LLC.
⋆⋆ Corresponding to yinzhou@waymo.com.

points. For each component of size $s$, we randomly label 1 point for each category whose number of points is more than $0.05s$. The motivation here is to prevent those noisy and ambiguous points within each component from decreasing the component purity. In real applications, human labelers may also miss or ignore those noisy categories to accelerate the annotation.

For the nuScenes [2] dataset, we share the same hyperparameters as SemanticKITTI, except for the following. We fuse every 40 adjacent scans, and ignore small components with no more than 10 points. For each component proposal of size $s$, we randomly label 1 (or 4) point(s) for each category whose number of points is more than $0.01s$, corresponding to the 0.2% (0.9%) settings. These subtle differences are mainly due to the points in the nuScenes [2] dataset are much sparser (e.g., the right inset shows the fused points for 0.5 seconds), and we fuse more points and annotate more labels to compensate for the point sparsity.




**Network training** As for contrastive prototype learning, the momentum parameter $m$ is empirically set to 0.99, temperature parameter $\tau$ is set to 0.1. In multi-scan distillation, we fuse the scans at time $\{t + 0.5i; i \in [-2, 2]\}$ for SemanticKITTI, and $\{t + 0.5i; i \in [-3, 3]\}$ for nuScenes. We tried multiple sets of parameters (different numbers of scans and intervals). They do lead to some differences ($\sim$3% mIOU), and we choose the best empirically. We keep all points for scan $i = 0$, and use voxel downsampling to sub-sample $120k$ points from other scans. The temperature $T$ is set to 4.

We sum up all loss terms with equal weights and train the models on 4 NVIDIA A100 GPUs. For SemanticKITTI, the batch size is 12 and 8 for the single-scan and the multi-scan model, respectively. For nuScenes, the batch size is 16 and 12 for the single-scan and the multi-scan model, respectively. We utilize the Adam optimizer, and the learning rate is initially set to 1e-3 and then decayed to 1e-4 after convergence. During distillation, the learning rate is set to 1e-4. Other training parameters are the same as Cylinder3D [12].

**Baseline Methods** We adopt the author released code to train OneThingOneClick [7] and ContrastiveSceneContext [4] on SemanticKITTI and nuScenes. For other methods, the results are either obtained from the literature or correspondences with the authors.

For **ContrastiveSceneContext** [4], we first compute the overlapping ratio between every pair of scans within each sequence, where the voxel size is set to $0.3m$. We then use pairs of scans whose overlapping ratio is no less than 30% for contrastive pre-training. During pre-training, we train the model with

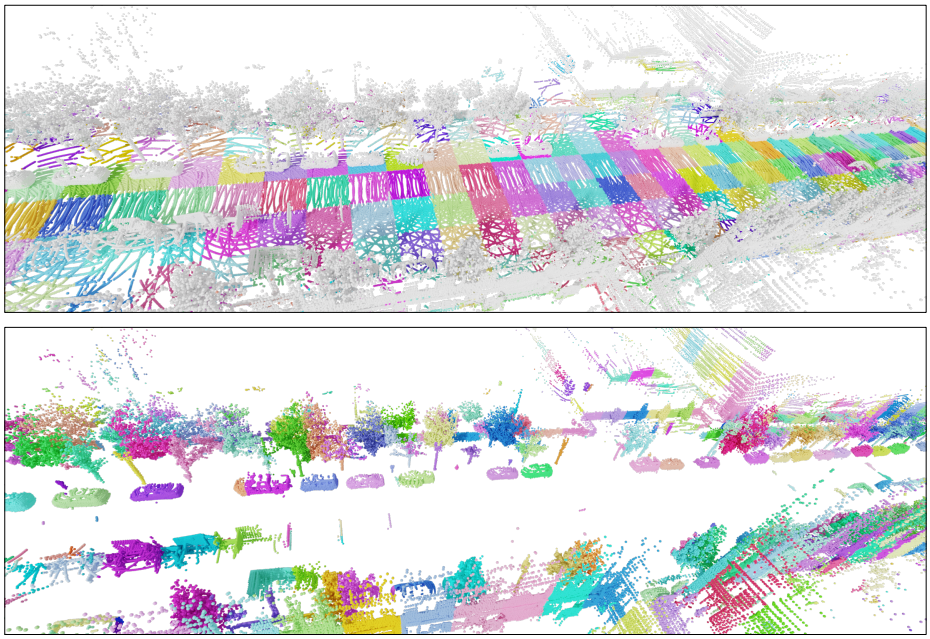


Fig. S1: **Examples of the pre-segmentation results.** First row: detected ground points of each cell. Non-ground points are colored in gray. Each other color indicates a proposed ground component. Second row: connected components of the non-ground points. Each color indicates a connected component. The example is from the nuScenes dataset, where 40 scans are fused.

a voxel size of $0.15m$ for 100k iterations. The batch size is 12 and 20 for SemanticKITTI and nuScenes, respectively. We then follow the provided pipeline to infer the point features and select points for labeling. After that, we train the segmentation network with the pre-trained weights for 30k iterations. The voxel size is set to $0.1m$, and the batch size is set to 18 and 36 SemanticKITTI and nuScenes, respectively. We disable the elastic distortion and the color-related data augmentation.

For **OneThingOneClick** [7], we first apply the geometrical partition described in [5] to generate the super-voxels, where only the point coordinates are used as input. We then randomly label a subset of super-voxels for a given annotation budget. We follow the authors' guidance to train the modules for three iterations. In each iteration, we train the 3D-U-Net for 32 epochs (51k iterations) and the RelationNet for 64 epochs (102k iterations). During training, the voxel size is set to $0.1m$, and the batch size is set to 12. We disable the elastic distortion for the data augmentation.

## S.2 Visual results of pre-segmentation

Fig. S1 shows the examples of our pre-segmentation results.

| Method | Anno. | mIoU | barrier | bicycle | bus | car | construction-vehicle | motorcycle | pedestrian | traffic-cone | trailer | truck | drivable-surface | other-flat | sidewalk | terrian | manmade | vegetation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (AF)2-S3Net [3] | | 62.2 | 60.3 | 12.6 | 82.3 | 80.0 | 20.1 | 62.0 | 59.0 | 49.0 | 42.2 | 67.4 | 94.2 | 68.0 | 64.1 | 68.6 | 82.9 | 82.4 |
| RangeNet++ [8] | | 65.5 | 66.0 | 21.3 | 77.2 | 80.9 | 30.2 | 66.8 | 69.6 | 52.1 | 54.2 | 72.3 | 94.1 | 66.6 | 63.5 | 70.1 | 83.1 | 79.8 |
| PolarNet [11] | | 71.0 | 74.7 | 28.2 | 85.3 | 90.9 | 35.1 | 77.5 | 71.3 | 58.8 | 57.4 | 76.1 | 96.5 | 71.1 | 74.7 | 74.0 | 87.3 | 85.7 |
| SPVNAS [9] | | 74.8 | 74.9 | 39.9 | 91.1 | 86.4 | 45.8 | 83.7 | 72.1 | 64.3 | 62.5 | 83.3 | 96.2 | 72.7 | 73.6 | 74.1 | 88.3 | 87.4 |
| **Cylinder3D [12]** | 100% | 75.4 | 75.3 | 41.7 | 91.6 | 86.1 | 52.9 | 79.3 | 79.2 | 66.1 | 61.5 | 81.7 | 96.4 | 72.3 | 73.8 | 73.5 | 88.1 | 86.5 |
| AMVNt [6] | | 77.0 | 77.7 | 43.8 | 91.7 | 93.0 | 51.1 | 80.3 | 78.8 | 65.7 | 69.6 | 83.5 | 96.9 | 71.4 | 75.1 | 75.3 | 90.1 | 88.3 |
| RPVNet [10] | | 77.6 | 78.2 | 43.4 | 92.7 | 93.2 | 49.0 | 85.7 | 80.5 | 66.0 | 66.9 | 84.0 | 96.9 | 73.5 | 75.9 | 76.0 | 90.6 | 88.9 |
| ContrastiveSC [4] | 0.2% | 63.5 | 65.6 | 0.0 | 82.7 | 87.3 | 42.8 | 46.3 | 57.1 | 32.2 | 59.0 | 76.4 | 94.2 | 62.5 | 65.9 | 68.8 | 87.8 | 86.8 |
| LESS (Ours) | 0.2% | **73.5** | 73.7 | 38.3 | 92.0 | 89.7 | 46.9 | 75.6 | 70.9 | 58.4 | 64.8 | 83.0 | 95.6 | 67.6 | 70.9 | 71.8 | 89.2 | 87.3 |
| ContrastiveSC [4] | 0.9% | 64.5 | 64.0 | 12.7 | 80.7 | 87.6 | 41.1 | 55.8 | 61.6 | 37.5 | 59.1 | 75.2 | 94.2 | 65.6 | 67.0 | 70.1 | 88.0 | 87.2 |
| LESS (Ours) | 0.9% | **74.8** | 75.0 | 42.3 | 91.9 | 89.9 | 51.0 | 80.0 | 72.6 | 60.1 | 64.9 | 83.6 | 95.7 | 67.5 | 71.7 | 73.1 | 89.5 | 87.6 |

Table S1: **Comparison of different methods on the nuScenes validation set.** Cylinder3D [12] is our fully supervised counterpart.

| | car | bicycle | motorcycle | truck | other-vehicle | person | bicyclist | motorcyclist | road | parking | sidewalk | other-ground | building | fence | vegetation | trunk | terrain | pole | traffic sign |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sparse (×0.1%) | 0.8 | **2.7** | 0.9 | 0.6 | 0.8 | **1.8** | **1.8** | **2.7** | 0.4 | 0.7 | 0.6 | 1.4 | 0.8 | 1.0 | 1.0 | **2.0** | 1.0 | **3.1** | **4.1** |
| propagated (%) | 79 | 12 | 75 | 77 | 75 | 52 | 64 | 48 | 16 | 6 | 9 | 17 | 77 | 25 | 55 | 29 | 32 | 28 | 9 |

Table S2: **The coverage of sparse labels and propagated labels for the SemanticKITTI dataset.** The numbers are the ratios between the number of sparse labels (and propagated labels) and the number of points within each category.

### S.3 Full results on nuScenes

Tab. S1 shows the full results on the nuScenes validation set.

### S.4 Full table of multi-scan distillation

Tab. S4 shows the full results of the multi-scan distillation. The multi-scan teacher model leverages the richer semantics via temporal fusion and achieves significantly better performances in the underrepresented categories, such as bicycle, person, and bicyclist. Through knowledge distillation from the teacher model, the student model also improves a lot in those categories.

### S.5 Label distribution

Tab. S2 and Tab. S3 summarize the distributions of the generated sparse labels and the propagated labels. By leveraging our proposed pre-segmentation and

| | barrier | bicycle | bus | car | construction-vehicle | motorcycle | pedestrian | traffic-cone | trailer | truck | drivable-surface | other-flat | sidewalk | terrian | manmade | vegetation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sparse (×0.1%) | 2.4 | **20.9** | 4.0 | 4.6 | 4.8 | **8.0** | **19.9** | **12.2** | 3.4 | 3.4 | 0.6 | 1.7 | 1.9 | 3.1 | 4.8 | 7.9 |
| propagated (%) | 16 | 16 | 53 | 52 | 54 | 46 | 29 | 20 | 49 | 59 | 32 | 2 | 2 | 11 | 62 | 55 |

Table S3: **The coverage of sparse labels and propagated labels for the nuScenes dataset.** The numbers are the ratios between the number of sparse labels (and propagated labels) and the number of points within each category.

| Method | mIOU | car | **bicycle** | motorcycle | **truck** | other-vehicle | **person** | **bicyclist** | motorcyclist | road | parking | sidewalk | other-ground | building | fence | vegetation | trunk | terrain | pole | traffic sign |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| single-scan (before) | 64.9 | 97 | **46** | 72 | **91** | 69 | **73** | **88** | 0 | 92 | 39 | 77 | 4 | 90 | 58 | 88 | 66 | 73 | 61 | 52 |
| multi-scan teacher | 66.8 | 97 | **52** | 82 | **94** | 72 | **78** | **92** | 0 | 93 | 40 | 79 | 1 | 89 | 54 | 87 | 70 | 72 | 64 | 53 |
| single-scan (after) | 66.0 | 97 | **50** | 73 | **94** | 67 | **76** | **92** | 0 | 93 | 40 | 79 | 3 | 91 | 60 | 87 | 68 | 71 | 62 | 51 |

Table S4: **Results of the multi-scan distillation on the SemanticKITTI validation set.** 0.1% annotations are used.

labeling policy, we put more emphasis on the underrepresented categories. For example, the ratios of sparse labels for bicycle and road are 2.68 vs. 0.36 in the SemanticKITTI dataset, and 20.85 vs. 0.63 in the nuScenes dataset. As for the propagated labels, we find the distributions are unbalanced. For categories, such as car and building, they are easier to be separated and form pure components, thus having high coverages of propagated labels. However, some categories, such as bicycle, road, sidewalk, and parking, are prone to be connected with other categories, thus having low coverages of propagated labels. The discrepancy between the distributions of the two types of labels confirms that we need to treat them separately instead of simply merging them with a single loss function.

### S.6 Robustness to label noise

In the paper, we use point labels from the original datasets to mimic the annotation policy, and no extra noise is added.

To evaluate the robustness of our method to label noise, we randomly change 3% (or 10%) of the sparse point labels to a random category, which alters weak labels and propagated labels accordingly. The resulting mIoU drops 2.1% (or 3.7%), which is within a reasonable range and verifies that our method will not be significantly affected by the label noise.

## 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 International Conference on Computer Vision (ICCV). pp. 9297–9307 (2019) 1
2. 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 (CVPR). pp. 11621–11631 (2020) 1, 2
3. Cheng, R., Razani, R., Taghavi, E., Li, E., Liu, B.: Af2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12547–12556 (2021) 4
4. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15587–15597 (2021) 2, 4
5. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4558–4567 (2018) 3
6. Liong, V.E., Nguyen, T.N.T., Widjaja, S., Sharma, D., Chong, Z.J.: Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. arXiv preprint arXiv:2012.04934 (2020) 4
7. Liu, Z., Qi, X., Fu, C.W.: One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1726–1736 (2021) 2, 3
8. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4213–4220. IEEE (2019) 4
9. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 685–702. Springer (2020) 4
10. Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S.: Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. arXiv preprint arXiv:2103.12978 (2021) 4
11. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9601–9610 (2020) 4
12. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9939–9948 (2021) 2, 4