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[2]

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

Abstract. Semantic segmentation of LiDAR point clouds is an important task in autonomous driving. However, training deep models via conventional supervised methods requires large datasets which are costly to label. It is critical to have label-efficient segmentation approaches to scale up the model to new operational domains or to improve performance on rare cases. While most prior works focus on indoor scenes, we are one of the first to propose a label-efficient semantic segmentation pipeline for outdoor scenes with LiDAR point clouds. Our method co-designs an efficient labeling process with semi/weakly supervised learning and is applicable to nearly any 3D semantic segmentation backbones. Specifically, we leverage geometry patterns in outdoor scenes to have a heuristic pre-segmentation to reduce the manual labeling and jointly design the learning targets with the labeling process. In the learning step, we leverage prototype learning to get more descriptive point embeddings and use multi-scan distillation to exploit richer semantics from temporally aggregated point clouds to boost the performance of single-scan models. Evaluated on the SemanticKITTI and the nuScenes datasets, we show that our proposed method outperforms existing label-efficient methods. With extremely limited human annotations (*e.g.*, 0.1% point labels), our proposed method is even highly competitive compared to the fully supervised counterpart with 100% labels.

1 Introduction

Light detection and ranging (LiDAR) sensors have become a necessity for most autonomous vehicles. They capture more precise depth measurements and are more robust against various lighting conditions compared to visual cameras. Semantic segmentation for LiDAR point clouds is an indispensable technology as it provides fine-grained scene understanding, complementary to object detection. For example, semantic segmentation help self-driving cars distinguish drivable and non-drivable road surfaces and reason about their functionalities, like parking areas and sidewalks, which is beyond the scope of modern object detectors.

Based on large-scale public driving-scene datasets [4,5], several LiDAR semantic segmentation approaches have recently been developed [68,59,9,62,50]. Typically, these methods require fully labeled point clouds during training. Since

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

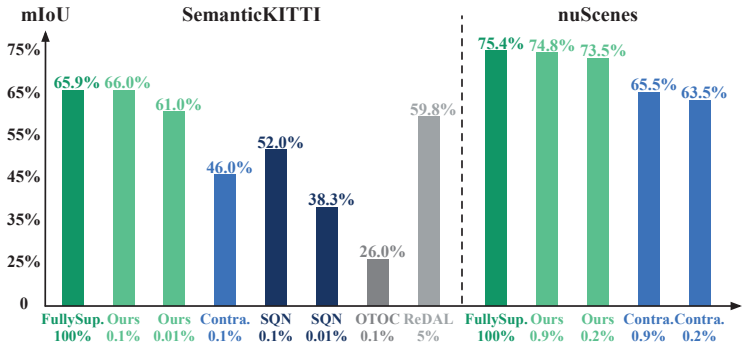


Fig. 1: We compare LESS with Cylinder3D [68] (our fully-supervised counterpart), ContrastiveSceneContext [23], SQN [24], OneThingOneClick [33], and ReDAL [55] on the SemanticKITTI [4] and nuScenes [5] validation sets. The ratio between labels used and all points is listed below each bar. Please note that all competing label-efficient methods mainly focus on indoor settings and are not specially designed for outdoor LiDAR segmentation.

a LiDAR sensor may perceive millions of points per second, exhaustively labeling all points is extremely laborious and time-consuming. Moreover, it may fail to scale when we extend the operational domain (*e.g.*, various cities and weather conditions) and seek to cover more rare cases. Therefore, to scale up the system, it is critical to have label-efficient approaches for LiDAR semantic segmentation, whose goal is to minimize the quantity of human annotations while still achieving high performance.

While there are some prior works studying label-efficient semantic segmentation, they mostly focus on indoor scenes [11,3] or 3D object parts [6], which are quite different in point cloud appearance and object type distribution, compared to the outdoor driving scenes (*e.g.*, significant variances in point density, extremely unbalanced point counts between common types, like ground and vehicles, and less common ones, such as cyclists and pedestrians). Besides, most prior explorations tend to address the problem from two independent perspectives, which may be less effective in our outdoor setting. Specifically, one perspective is improving labeling efficiency, where the methods resort to active learning [47,55,34], weak labels [44,54], and 2D supervision [53] to reduce labeling efforts. The other perspective focuses on training, where the efforts assume the partial labels are given and design semi/weakly supervised learning algorithms to exploit the limited labels and strive for better performance [33,60,44,61,20,34,66].

This paper proposes a novel framework, label-efficient semantic segmentation (LESS), for LiDAR point clouds captured by self-driving cars. Different from prior works, our method co-designs the labeling process and the model learning. Our co-design is based on two principles: 1) the labeling step is designed to provide bare minimum supervision, which is suitable for state-of-the-art semi/weakly supervised segmentation methods; 2) the model training step can tap into the labeling policy as a prior and deduce more learning targets. The proposed method can fit in a straightforward way with most state-of-the-

art LiDAR segmentation backbones without introducing any network architectural change or extra computational complexity when deployed onboard. Our approach is suitable for effectively labeling and learning from scratch. It is also highly compatible with mining long-tail instances, where, in practice, we mainly want to identify and annotate rare cases based on trained models.

Specifically, we leverage a philosophy that outdoor-scene objects are often well-separated when isolating ground points and design a heuristic approach to pre-segment an outdoor scene into a set of connected components. The component proposals are of high purity (*i.e.*, only contain one or a few classes) and cover most of the points. Then, instead of meticulously labeling all points, the annotators are only required to label one point per class for each component. In the model learning process, we train the backbone segmentation network with the sparse labels directly annotated by humans as well as the derived labels based on component proposals. To encourage a more descriptive embedding space, we employ contrastive prototype learning [18,29,48,63,33], which increases intra-class similarity and inter-class separation. We also leverage a multi-scan teacher model to exploit richer semantics within the temporally fused point clouds and distill the knowledge to boost the performance of the single-scan model.

We evaluate the proposed method on two large-scale autonomous driving datasets, SemanticKITTI [4] and nuScenes [5]. We show that our method significantly outperforms existing label-efficient methods (see Fig. 1). With extremely limited human annotations, such as 0.1% labeled points, the approach achieves highly competitive performance compared to the fully supervised counterpart, demonstrating the potential of practical deployment.

In summary, our contribution mainly includes:

- Analyze how label-efficient segmentation of outdoor LiDAR point clouds differs from the indoor settings, and show that the unbalanced category distribution is one of the main challenges.
- Leverage the unique geometric structure of LiDAR point clouds and design a heuristic algorithm to pre-segment input points into high-purity connected components. A customized labeling policy is then proposed to exploit the components with tailored labels and losses.
- Adapt beneficial components into label-efficient LiDAR segmentation and carefully design a network-agnostic pipeline that achieves on-par performance with the fully supervised counterpart.
- Evaluate the proposed pipeline on two large-scale autonomous driving datasets and extensively ablate each module.

2 Related work

2.1 Segmentation networks for LiDAR point clouds

In contrast to indoor-scene point clouds, outdoor LiDAR point clouds' large scale, varying density, and sparsity require the segmentation networks to be more efficient. Many works project the 3D point clouds from spherical view [43,27,10,12,58,2,30,39](*i.e.*, range images) or bird's-eye-view [45,65] onto 2D images, or try to fuse different views [31,19,1]. There are also some works directly

dataset	vegetation	road	building	car	motorcycle	person	bicycle
SemanticKITTI	1606	1197	799	257	2	2	1
nuScenes	867	2242	1261	270	3	16	1

Table 1: **Point distribution across the most common and rarest categories of SemanticKITTI [4] and nuScenes [5].** Numbers are normalized by the sample quantity of bicycles.

consuming point clouds [52,14,25,8]. They aim to structure the irregular data more efficiently. Zhu *et al*. [68] employ the voxel-based representation and alleviate the high computational burden by leveraging cylindrical partition and sparse asymmetrical convolution. Recent works also try to fuse the point and voxel representations [50,62,64,9], and even with range images [59]. All of these works can serve as the backbone network in our label-efficient framework.

2.2 Label-efficient 3D semantic segmentation

Label-efficient 3D semantic segmentation has recently received lots of attention [17]. Previous explorations are mainly two-fold: labeling and training.

As for labeling, several approaches seek active learning [47,55,34], which iteratively selects and requests points to be labeled during the network training. Hou *et al*. [23] utilize features from unsupervised pre-training to choose points for labeling. Wang *et al*. [53] project the point clouds to 2D and leverage 2D supervision signals. Some works utilize scene-level or sub-cloud-level weak labels [44,54]. There are also several approaches using rule-based heuristics or handcrafted features to help annotation [37,51,20].

As for training, Xie *et al*. [23,57] utilize contrastive learning for unsupervised pre-training. Some approaches employ self-training to generate pseudo-labels [33,60,44]. Lots of works use Conditional Random Fields (CRFs) [33,61,20,34] or random walk [66] to propagate labels. Moreover, there are also works that utilize prototype learning [33,66], siamese learning [61,44], temporal constraints [36], smoothness constraints [44,47], attention [54,66], cross task consistency [44], and synthetic data [56] to help training.

However, most recent works mainly focus on indoor scenes [11,3] or 3D object parts [6], while outdoor scenarios are largely under-explored.

3 Method

In this section, we present our LESS framework. Since existing label-efficient segmentation works typically address domains other than autonomous driving, we first conduct a pilot study to understand the challenges in this novel setting and introduce motivations behind LESS (Sec. 3.1). After briefly going over our LESS framework (Sec. 3.2), we dive into the details of each part (Secs. 3.3 to 3.6).

3.1 Pilot study: what should we pay attention to?

Previous works [33,23,47,53,44,54,61,66] on label-efficient 3D semantic segmentation mainly focused on indoor datasets, such as ScanNet-v2 [11] and S3DIS [3]. In these datasets, input points are sampled from high-quality reconstructed meshes and are thus densely and uniformly distributed. Also, objects in indoor scenarios typically share similar sizes and have a relatively balanced class distribution. However, in outdoor settings, input point clouds demonstrate substantially

higher complexity due to the varying point density and the ubiquitous occlusions throughout the scene. Moreover, in outdoor driving scenes, the sample distribution across different categories is highly unbalanced due to factors including occurring frequency and object size. Tab. 1 shows the point distribution over two autonomous driving datasets, where the numbers of road points are 1,197 and 2,242 times larger than that of bicycle points, respectively. The extremely unbalanced distribution adds extra difficulty for label-efficient segmentation, whose goal is to only label a tiny portion of points.

We conduct a pilot study to further examine this challenge. Specifically, we train a state-of-the-art semantic segmentation network, Cylinder3D [68], on the SemanticKITTI dataset with three intuitive setups: (a) 100% labels, (b) randomly annotating 0.1% points per scan, and (c) randomly selecting 0.1% scans and annotating all points for the selected scans. The results are shown in Appendix S.1. Without any special efforts, "0.1% random points" can already achieve a mean IoU of 48.0%, compared to 65.9% by the fully supervised version. On common categories, such as car, road, building, and vegetation, the performances of the "0.1% label" models are close to the fully supervised model. However, on the underrepresented categories, such as bicycle, person, and motorcycle, we observe substantial performance gaps compared to the fully supervised model. These categories tend to have small sizes, appear less frequently, and are thus more vulnerable when reducing the annotation budget. However, they are still critical for many applications such as autonomous driving. Moreover, we find that "0.1% random points" outperforms "0.1% random scans" by a large margin, mainly due to its label diversity.

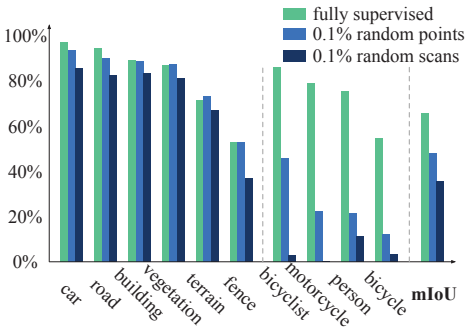


Fig. 2: **Pilot study: performances (IoU) of the most common and rarest categories.** Models are trained with 100% of labels and 0.1% of labels (in terms of points or scans) on SemanticKITTI [4].

These observations inspire us to rethink the existing paradigm of label-efficient segmentation. While prior works typically focus on either efficient labeling or improving training approaches, we argue that it can be more effective to address the problem by co-designing both. By integrating the two parts, we may cover more underrepresented instances with a limited labeling budget, and exploit the labeling efforts more effectively during network training.

3.2 Overview

Our LESS framework integrates pre-segmentation, labeling, and network training. It can work with most existing LiDAR segmentation backbones without changing their network architectures or inference latency. As shown in Fig. 3, our pipeline takes raw LiDAR sequences as input. It first employs a heuristic method to partition the point clouds into a set of high-purity components

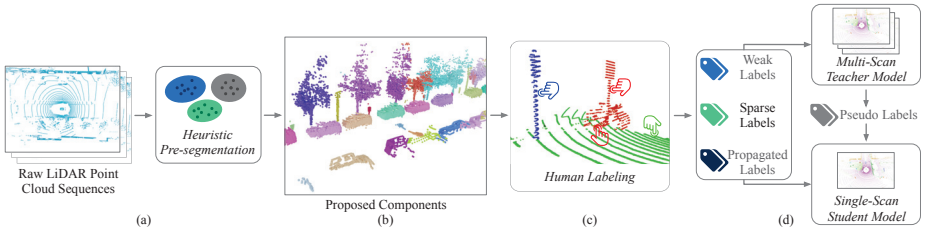


Fig. 3: **Overview of our LESS pipeline.** (a) We first utilize a heuristic algorithm to pre-segment each LiDAR sequence into a set of connected components. (b) Examples of the proposed components. Different colors indicate different components. For clear visualization, components of ground points are not shown. (c) Human annotators only need to coarsely label each component. Each color denotes a proposed component, and each click icon indicates a labeled point. Only sparse labels are directly annotated by humans. (d) We then train the network to digest various labels and utilize multi-scan distillation to exploit richer semantics in the temporally fused point clouds.

(Sec. 3.3). Instead of exhaustively labeling all points, annotators only need to quickly label a few points for each component proposal (*e.g.*, one point label for each class that appears). Besides the human-annotated sparse labels, we derive other types of labels so as to train the network with more context information (Sec. 3.4). During the network training, we employ contrastive prototype learning to realize a more descriptive embedding space (Sec. 3.5). We also boost the single-scan model by distilling the knowledge from a multi-scan teacher, which exploits richer semantics within the temporally fused point clouds (Sec. 3.6).

3.3 Pre-segmentation

We design a heuristic pre-segmentation to subdivide the point cloud into a collection of components. Each resulting component proposal is of high purity, containing only one or a few categories, which facilitates annotators to coarsely label all the proposals, *i.e.*, one point label per class (Sec. 3.4). In this way, we can derive dense supervision by disseminating the sparse point-wise annotations to the whole components. Since modern networks can learn the semantics of homogeneous neighborhoods from sparse annotations, spending lots of annotation budgets on large objects may be futile. Our component-wise coarse annotation is agnostic to the object size, which benefits underrepresented small objects.

For indoor scenarios, many prior arts [33,20,47] leverage the surface normal and color information to generate super voxels and assume that the points within each super voxel share the same category. These approaches, however, might not generalize to outdoor LiDAR point clouds, where the surface can be noisy and color information is not available. Since the homogeneity assumption is hard to hold, we instead propose to lift this constraint and allow each component to contain more than one category.

Unlike indoor scenarios, objects in outdoor scans are often well-separated after detecting and isolating the ground points. Inspired by this philosophy, we design an intuitive approach to pre-segment each LiDAR sequence, which includes

four steps: **(a) Fuse overlapping scans.** We first split a LiDAR sequence into sub-sequences, each containing t consecutive scans. We then fuse the scans of each sub-sequence based on the provided ego-poses. In this way, we can label the same instance across overlapping scans at one click. **(b) Detect ground points.** While the ground surface may not be flat at the full-scene scale, we assume for each local region (*e.g.*, 5m $\times$ 5m), the ground points can be fitted by a plane. We thus partition the whole scene into a uniform grid according to the xy coordinates, and then employ the RANSAC algorithm [15] to detect the ground points for each local cell. Since the ground points may belong to different categories (*e.g.*, parking zone, sidewalk, and road), we regard the ground points from each local cell as a single component instead of merging all of them. We allow a single ground component to contain multiple classes, and one point per class will be labeled later. **(c) Construct connected components.** After detecting and isolating the ground points, the remaining objects are often well-separated. We build a graph G, where each node represents a point. We connect every pair of points (u, v) in the graph, whose Euclidean distance is smaller than a threshold τ. We then divide the points into groups by calculating the connected components for the graph G. Due to the non-uniform point density distribution of the LiDAR point clouds, it is hard to use a fixed threshold across different ranges. We thus propose an adaptive threshold $\tau(u, v) = \max(r_u, r_v) \times d$ to compensate for the varying density, where r_u and r_v are the distances between the points and the sensor centers, and d is a pre-defined hyper-parameter. **(d) Subdivide large components.** After step (c), there usually exist some connected components covering an enormous area (*e.g.*, buildings and vegetation), which are prone to include some small objects. To keep each component of high purity and facilitate network training, we subdivide oversized components to ensure each component is bounded within a fixed size. Also, we ignore small components with only a few points, which tend to be noisy and can lead to excessive component proposals.

In practice, we find our pre-segmentation generates a small number of components for each sequence. The component proposals cover most of the points, and each component tends to have high purity. These open up the possibility of quickly bootstrapping the labeling from scratch. Moreover, unlike other methods [33,20,47] relying on various handcrafted features, our method only utilizes the simple geometrical connectivity, allowing it to generalize to various scenarios without tuning lots of hyper-parameters. Please refer to Sec. 4.4 for statistics of the pre-segmentation results and the supplementary material for more details.

3.4 Annotation policy & training labels

Instead of meticulously labeling every point, we propose to coarsely annotate the component proposals. Specifically, for each component proposal, an annotator needs to first skim through the component and then label only one point for each identified category. Fig. 3 (c) illustrates an example where the pre-segmentation yields three components colored in red, blue, and green, respectively. Because the blue component only has traffic-sign points, the annotator only needs to randomly select one point to label. The green component is similar, as it only contains road points. In the red component, there is a bicycle lying against a

traffic sign, and the annotator needs to select one point for each class to label. By coarsely labeling all components, we are unlikely to miss any underrepresented instances, as the proposed components cover the majority of points. Moreover, since the number of components is orders of magnitude smaller than that of points and our coarse annotation policy frees annotators from carefully labeling instance boundaries (required in the labeling process to build SemanticKITTI [4] dataset), we are thus able to reduce manual labeling costs.

Based on the component proposals, we can obtain three types of labels. **Sparse labels:** points directly labeled by annotators. Although only a tiny subset of points are labeled, sparse labels provide the most accurate and diverse supervision. **Weak labels:** classes that appear in each component. Weak labels are derived based on human-annotated sparse labels within each component. In the example of Fig. 3 (c), all red points can only be either bicycles or traffic signs. We disseminate weak labels from each component to the points therein. The multi-category weak labels provide weak but dense supervision and cover most points. **Propagated labels:** for the pure components (*i.e.*, only one category appears), we can propagate the label to the entire component. Given the effectiveness of our pre-segmentation approach, the propagated labels also cover a wide range of points. However, since some categories may be easier to be separated and prone to form pure components, the distribution of the propagated labels may be biased and less diverse than the sparse labels.

We formulate a joint loss function by exploiting the three types of labels: $\mathcal{L} = \mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{propagated}} + \mathcal{L}_{\text{weak}}$, where $\mathcal{L}_{\text{sparse}}$ and $\mathcal{L}_{\text{propagated}}$ are weighted cross-entropy loss with respect to the sparse labels and propagated labels, respectively. We utilize inverse square root of label frequency [38,69,35] as category weights to emphasize underrepresented categories. Here, we calculate a cross-entropy loss for each label type separately, because propagated labels significantly outnumber sparse labels while sparse labels provide more diverse supervision.

Denote the weak labels as binary masks l_{ij} for point i and category j. $l_{ij} = 1$ when point i belongs to a component that contains category j. We exploit the multi-category weak labels by penalizing the impossible predictions:

$$\mathcal{L}_{\text{weak}} = -\frac{1}{n}\sum_{i=1}^{n}\log(1 - \sum_{l_{ij}=0} p_{ij}) \tag{1}$$

where p_{ij} is the predicted probability of point i, and n is the number of points. Prior approaches [54,44] aggregate per-point predictions into component-level predictions and then utilize the multiple-instance learning loss (MIL) [41,42] to supervise the learning. Here, we only penalize the negative predictions without encouraging the positive ones. This is because our network takes a single-scan point cloud as input, but the labels are collected and derived over the temporally fused point clouds. Hence, a positive instance may not always appear in each individual scan, due to occlusions or limited sensor coverage.

3.5 Contrastive prototype learning

Besides the great success in self-supervised representation learning [7,21,40], contrastive learning has also shown effectiveness in supervised learning and few-shot

learning [26,18,46,49]. It can overcome shortcomings of the cross-entropy loss, such as poor margins [13,32,26,63], and construct a more descriptive embedding space. Following [18,29,48,63,33], we exploit the limited annotations by learning distinctive class prototypes (*i.e.*, class centroids in the feature space). Without pre-training, a contrastive prototype loss $\mathcal{L}_{\text{proto}}$ is added to Sec. 3.4 as an auxiliary loss. Due to the limited annotations and unbalanced label distribution, only using samples within each batch to determine class prototypes may lead to unstable results. Inspired by the idea of momentum contrast [21], we instead learn the class prototypes $\mathbf{P}_c$ by using a moving average over iterations:

$$\mathbf{P}_c \leftarrow m\mathbf{P}_c + (1-m)\frac{1}{n_c}\sum_{y_i=c}\text{stopgrad}(h(f(x_i))) \tag{2}$$

where $f(x_i)$ is the embedding of point x_i, h is a linear projection head with vector normalization, stopgrad denotes the stop gradient operation, y_i is the label of x_i, n_c is the number of points with label c in a batch, and m is a momentum coefficient. In the beginning, $\mathbf{P}_c$ are initialized randomly.

The prototype loss $\mathcal{L}_{\text{proto}}$ is calculated for the points with sparse labels and propagated labels within each batch:

$$\mathcal{L}_{\text{proto}} = \frac{1}{n}\sum_{i}^{n} -w_{y_i}\log\frac{\exp(h(f(x_i))\cdot\mathbf{P}_{y_i}/\tau)}{\sum_c \exp(h(f(x_i))\cdot\mathbf{P}_c/\tau)} \tag{3}$$

where $h(f(x_i))\cdot\mathbf{P}_{y_i}$ indicates the cosine similarity between the projected embedding and the prototype, τ is a temperature hyper-parameter, n is the number of points, and w_{y_i} is the inverse square root weight of category y_i. $\mathcal{L}_{\text{proto}}$ aims to learn a better embedding space by increasing intra-class compactness and inter-class separability.

3.6 Multi-scan distillation

We aim to learn a segmentation network that takes a single LiDAR scan as input and can be deployed in real-time onboard applications. During our label-efficient training, we can train a multi-scan network as a teacher model. It applies temporal fusion of multiple scans and takes the densified point cloud as input, compensating for the sparsity and incompleteness within a single scan. The teacher model is thus expected to exploit the richer semantics and perform better than a single-scan model. Especially, it may improve the performance for those underrepresented categories, which tend to be small and sparse. After that, we distill the knowledge from the multi-scan teacher model to boost the performance of the single-scan student model.

Specifically, for a scan at time t, we fuse the point clouds of neighboring scans at time $\{t + i\Delta; i \in [-2, 2]\}$ (Δ is a time interval) using the ego-poses of the LiDAR sensor. To enable a large batch size, we use voxel subsampling [67] to normalize the fused point cloud to a fixed size. Labels are then fused accordingly. Besides the spatial coordinates, we also concatenate an additional channel indicating the time index i of each point. The teacher model is trained using the loss functions introduced in Secs. 3.4 and 3.5.

The student model shares the same backbone network and is first trained from scratch in the same way as the teacher model except for the single-scan

input. We then fine-tune it by incorporating an additional distillation loss $\mathcal{L}_{\text{dis}}$. Specifically, following [22], we match student predictions with the soft pseudo-labels generated by the teacher model via a cross-entropy loss:

$$\mathcal{L}_{\text{dis}} = -\frac{T^2}{n}\sum_{i}^{n}\sum_{c}\frac{\exp(u_{ic}/T)}{\sum_{c'}\exp(u_{ic'}/T)}\log\left(\frac{\exp(v_{ic}/T)}{\sum_{c'}\exp(v_{ic'}/T)}\right) \tag{4}$$

where u_{ic} and v_{ic} are the predicted logits for point i and category c by the teacher and student models respectively, and T is a temperature hyper-parameter. A higher temperature is typically used so that the probability distribution across classes is smoother, and the distillation is thus encouraged to match the negative logits, which also contain rich information. The cross-entropy is multiplied by T^2 to align the magnitudes of the gradients with existing other losses [22].

Please note that the idea of multi-scan distillation may only be beneficial for our label-efficient LiDAR segmentation setting. For the fully supervised setting, all labels are already available and accurate, and there is no need to leverage the pseudo labels. For the indoor setting, all points are sampled from high-quality reconstructed meshes, and there is no need for a multi-scan teacher model.

4 Experiments

We employ Cylinder3D [68], a recent state-of-the-art method for LiDAR semantic segmentation, as our backbone network. We utilize ground truth labels to mimic the obtained human annotations, and no extra noise is added. Please refer to the supplementary material for more implementation and training details.

We evaluate the proposed method on two large-scale autonomous driving datasets, SemanticKITTI [4] and nuScenes [5]. **SemanticKITTI** [4] is collected in Germany with 64-beam LiDAR sensors. The (sensor) capture and annotation frequency is 10 Hz. It contains 10 training sequences (19k scans), 1 validation sequence (4k scans), and 11 testing sequences (20k scans). 19 classes are used for segmentation. **nuScenes** [5] is collected in Boston and Singapore with 32-beam LiDAR sensors. Although the (sensor) capture frequency is 20Hz, the annotation frequency is only 2Hz. It contains 700 training sequences (28k scans), 150 validation sequences (6k scans), and 150 testing sequences (6k scans). 16 classes are used for segmentation. For both datasets, we follow the official guidance [4,5] to use mean intersection-over-union (mIoU) as the evaluation metric.

4.1 Comparison on SemanticKITTI

We compare the proposed method with both label-efficient [55,24,33,23] and fully supervised [58,65,1,16,12,27,50,31,68] methods. Please note that all competing label-efficient methods mainly focus on indoor settings and are not specially designed for outdoor LiDAR segmentation. Among them, ContrastiveSC [23] employs contrastive learning as unsupervised pre-training and uses the learned features for active labeling, ReDAL [55] also employs active labeling, OneThingOneClick [33] proposes a self-training approach and iteratively propagate the labels, and SQN [24] presents a network by leveraging the similarity between neighboring points. We report the results on the validation set. Since ContrastiveSC [23] and OneThingOneClick [33] are only tested on indoor datasets in

Method	Annot.	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
SqueezeSegV3 [58]	100%	52.7	86	31	48	51	42	52	52	0	95	47	82	0	80	47	83	53	72	42	38
PolarNet [65]		53.6	92	31	39	46	24	54	62	0	92	47	78	2	89	46	85	60	72	58	42
MPF [1]		57.0	94	28	55	62	36	57	74	0	95	47	81	1	88	53	86	54	73	57	42
S-BKI [16]		57.4	94	34	57	45	27	53	72	0	94	50	84	0	89	60	87	63	75	64	45
TemporalLidarSeg [12]		61.3	92	43	54	84	61	64	68	0	95	44	83	1	89	60	85	64	71	59	47
KPRNet [27]		63.1	95	43	60	76	51	75	81	0	96	51	84	0	90	60	88	66	76	63	43
SPVNAS [50]		64.7	97	35	72	81	66	71	86	0	94	48	81	0	92	67	88	65	74	64	49
AMVNet [31]		65.2	96	49	65	89	55	71	86	0	96	54	83	0	91	62	88	67	74	65	49
Cylinder3D [68]		65.9	97	55	79	80	67	75	86	1	95	46	82	1	89	53	87	71	71	66	53
Cylinder3D*		66.2	97	48	72	94	67	74	91	0	93	44	79	3	91	60	88	70	72	63	53
ReDAL [55]	5%	59.8	95	30	59	63	50	63	84	1	92	39	78	1	89	54	87	62	74	64	50
OneThingOneClick [33]	0.1%	26.0	77	0	0	2	1	0	2	0	63	0	38	0	73	44	78	39	53	25	0
ContrastiveSC [23]	0.1%	46.0	93	0	0	62	45	28	0	0	90	39	71	6	90	42	89	57	75	54	34
SQN [24]	0.1%	52.0	93	8	35	59	46	41	59	0	91	37	76	1	89	51	85	61	73	53	35
LESS (Ours)	0.1%	**66.0**	97	50	73	94	67	76	92	0	93	40	79	3	91	60	87	68	71	62	51
SQN [24]	0.01%	38.3	83	0	22	12	17	15	47	0	85	21	65	0	79	37	77	46	67	44	12
LESS (Ours)	0.01%	**61.0**	96	33	61	73	59	68	87	0	92	38	76	5	89	52	87	67	71	59	46

Table 2: **Comparison on the SemanticKITTI validation set.** Cylinder3D [68] is our fully supervised counterpart. Cylinder3D* is our re-trained version with our proposed prototype learning and multi-scan distillation.

the original paper, we adapt the source code published by the authors and train their models on SemanticKITTI [4]. For other methods, the results are either obtained from the literature or correspondences with the authors.

Tab. 2 lists the results, where our method outperforms existing label-efficient methods by a large margin. With only 0.1% sparse labels (as defined in Sec. 3.4), it even completely match the performance of the fully supervised baseline Cylinder3D [68], which demonstrates the potential of deployment into real applications. By checking the breakdown results, we find that the differences between methods mainly come from the underrepresented categories, such as bicycle, motorcycle, person, and bicyclist. Existing label-efficient methods, which are mainly designed for indoor settings, suffer a lot from the highly unbalanced sample distribution, while our method is remarkably competitive in those underrepresented classes. See Fig. 4 for further demonstration. OneThingOneClick [33] fails to produce decent results, which is partially due to its pure super-voxel assumption that does not always hold in outdoor scenes. As for the 0.01% annotations setting, the performance of SQN [24] drops drastically to 38.3%, whereas our proposed method can still achieve a high mIoU of 61.0%. For completeness, we also re-train Cylinder3D [68] with our proposed prototype learning and multi-scan distillation. We find that the two strategies provide marginal gain in the fully-supervised setting, where all labels are available and accurate.

4.2 Comparison on nuScenes

We also compare the proposed method with existing approaches on the nuScenes [5] dataset and report the results on the validation set. Since the author-released model of Cylinder3D [68] utilizes SemanticKITTI for pre-training, here, we report its result based on training the model from scratch for a fair comparison.

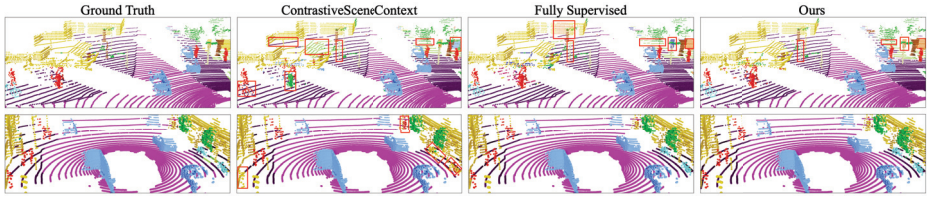


Fig. 4: **Qualitative examples on the SemanticKITTI [4] (first row) and nuScenes [5] (second row) validation sets.** Please zoom in for the details. Red rectangles highlight the wrong predictions. Our results are similar to the fully supervised counterpart, while ContrastiveSceneContext [23] produces worse results on underrepresented categories (see persons and bicycles). Please noet that, points in two datasets (with different density) are visualized in different point size for better visualization.

Method	Anno.	mIOU(%)
(AF)2-S3Net [9]	100%	62.2
SPVNAS [50]		74.8
Cylinder3D [68]		75.4
AMVNet [31]		77.2
RPVNet [59]		77.6
ContrastiveSC [23]	0.2%	63.5
LESS (Ours)	0.2%	**73.5**
ContrastiveSC [23]	0.9%	65.5
LESS (Ours)	0.9%	**74.8**

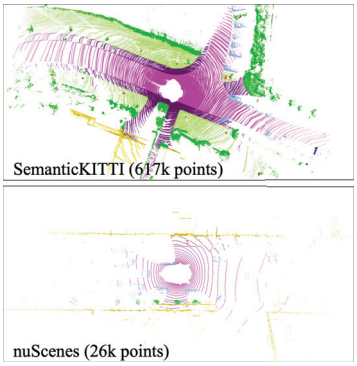
Table 3: **Comparison on nuScenes validation set.** Cylinder3D [68] is our fully supervised counterpart.

pre-seg.	weak labels	propa. labels	proto. learning	multi-scan distillation	mIoU (%)
✗	✗	✗	✗	✗	48.1
✓	✗	✗	✗	✗	59.3
✓	✓	✗	✗	✗	61.6
✓	✗	✓	✗	✗	62.2
✓	✓	✓	✗	✗	63.5
✓	✓	✓	✓	✗	64.9
✓	✓	✓	✓	✓	66.0

Table 4: **Ablation study on the SemanticKITTI validation set.** All variants use 0.1% sparse labels.

For other fully-supervised methods [9,50,31,59], the results are either obtained from the literature or correspondences with the authors. Since no prior label-efficient work is tested on the nuScenes [5] dataset, we adapt the source code published by the authors to train ContrastiveSceneContext [23] from scratch.

We want to point out that points in the nuScenes dataset are much sparser than those in SemanticKITTI. In nuScenes, only 2 scans per second are labeled, while in SemanticKITTI, 10 scans per second are labeled. Due to the difference of sensors (32-beam vs. 64-beam), the number of points per scan in nuScenes is also much smaller (26k vs. 120k). See the right inset for the comparison of two datasets (fused points for 0.5 seconds). Considering the sparsity of the original ground truth labels, here we report the 0.2% and 0.9% annotation settings.



Tab. 3 shows the results, where our proposed method outperforms ContrastiveSceneContext [23] by a large margin. With only 0.2% sparse labels, our result is also highly competitive with the fully-supervised counterpart [68].

Statistics	SemanticKITTI	nuScenes
one-category components	68.6%	80.6%
two-category components	23.8%	14.9%
components with more than two categories	7.6%	4.5%
average number of categories per component	1.40	1.25
coverage of sparse labels	0.1%	0.2%
coverage of propagated labels	42.0%	53.6%
coverage of weak labels	95.5%	99.0%

Table 5: **Statistics of the pre-segmentation and labeling.** Only sparse labels are directly annotated by humans.

annotation policy	pre-segmentation	mIoU (%)	#labels for motorcycle	IoU (%) of motorcycle
randomly sample points	✗	48.1	943	22.2
randomly sample scans	✗	35.6	548	0.0
active labeling [23]	✗	54.2	456	36.7
uniform grid partition	✓	61.4	1024 (76k)	61.3
geometric partition [28,33]	✓	61.9	1190 (294k)	64.0
LESS (Ours)	✓	64.9	1146 (933k)	72.3

Table 6: **Comparison of various annotation policies on SemanticKITTI.** All methods utilize 0.1% annotations and the same backbone network [68]. The fourth column indicates the number of sparse labels (and propagated labels) for an underrepresented category (*i.e.*, motorcycle). Multi-scan distillation is not utilized here. The IoU results are calculated on the validation set.

4.3 Ablation study

Tab. 4 shows the ablation study of each component. The first row is the result of training with 0.1% random point labels. By incorporating the pre-segmentation, we spend the limited annotation budget on more underrepresented instances, thereby significantly increasing mIoU from 48.1% to 59.3%. Derived from the component proposals, weak labels and propagated labels complement the human-annotated sparse labels and provide dense supervision. Compared to multi-category weak labels, propagated labels provide more accurate supervision and thus lead to a slightly higher gain. Both contrastive prototype learning and multi-scan distillation further boost the performance and finally close the gap between LESS and the fully-supervised counterpart in terms of mIoU.

4.4 Analysis of pre-segmentation & labeling

By leveraging the unique geometric structure and a careful design, our pre-segmentation works well for outdoor LiDAR point clouds. Tab. 5 summarizes some statistics of the pre-segmentation and labeling results. For both datasets, only less than 10% of the components contain more than two categories, which validates that our pre-segmentation generates high-purity components. The high "coverage of propagated labels" indicates that we thus deduce a good amount of "free" supervision from the pure components. The low "coverage of sparse labels" shows that annotators indeed only need to label a tiny portion of points, thus reducing human effort. The "coverage of weak labels" confirms that the proposed components can faithfully cover most points. Furthermore, the consistent results across two distinct datasets verify that our method generalizes well in practice.

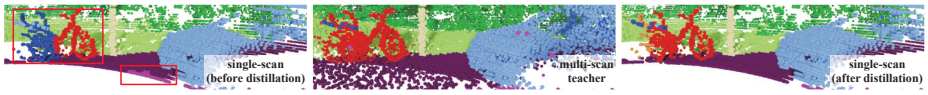


Fig. 5: **Improving segmentation with multi-scan distillation.** The multi-scan teacher leverages the richer semantics via temporal fusion to accurately segment the bicycle and ground, which provides high-quality supervision to enhance the single-scan model.

single-scan (before)		multi-scan teacher		single-scan (after)	
mIoU	bicycle	mIoU	bicycle	mIoU	bicycle
64.9%	45.6%	66.8%	51.5%	66.0%	49.9%

Table 7: **Results of the multi-scan distillation on the SemanticKITTI validation set.** 0.1% annotations are used.

Tab. 6 shows the comparison of different annotation policies (*i.e.*, how to use the labeling budget). The first two baselines are introduced in Sec. 3.1, "active labeling" utilizes the features from contrastive pre-training to actively select points [23], "uniform grid partition" uniformly divides the fused point clouds into a grid according to the xy coordinates and treats each cell as a component, "geometric partition" extracts handcrafted geometric features and solves a minimal partition problem [28,68]. All of them are trained with the same backbone Cylinder3D [68]. The first three methods employ no pre-segmentation and are trained with $\mathcal{L}_{\text{sparse}}$ only. The other approaches utilize our labeling policy (*i.e.*, one label per class for each component) and are trained with additional $\mathcal{L}_{\text{propagated}}$, $\mathcal{L}_{\text{weak}}$, and $\mathcal{L}_{\text{proto}}$. As a result, their performances are much higher than the first three methods. We also report the number of labels and the IoU for an underrepresented category. We see that our policy leads to more useful supervisions and higher IoUs for underrepresented categories.

4.5 Analysis of multi-scan distillation

Tab. 7 and Fig. 5 show the results of multi-scan distillation. The teacher model exploits the densified point clouds via temporal fusion and thus performs better than the single-scan model (even compared to the fully supervised single-scan model). Through knowledge distillation from the teacher model, the student model improves a lot in the underrepresented classes and completely matches the fully supervised model in mIoU.

5 Conclusion and future work

We study label-efficient LiDAR point cloud semantic segmentation and propose a pipeline that co-designs the labeling and the model learning and can work with most 3D segmentation backbones. We show that our method can utilize bare minimum human annotations to achieve highly competitive performance.

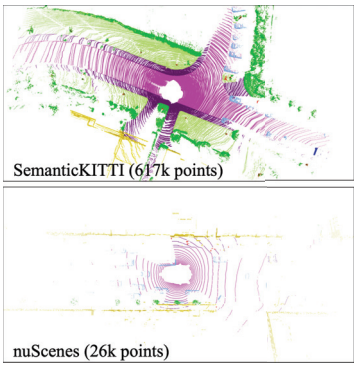
We have shown LESS is an effective approach for bootstrapping labeling and learning from scratch. In addition, LESS is also highly compatible for efficiently improving a performant model. With the predictions of an existing model, the proposed pipeline can be used for annotators to pick and label component proposals of high-values, such as underrepresented classes, long-tail instances, classes with most failures, *etc*. We leave this for future exploration.

In this supplementary material, we first present the implementation and training details of our proposed method and baseline methods (Appendix S.1). We then show the visual examples of our pre-segmentation results (Appendix S.2), the full results on the nuScenes dataset (Appendix S.3), and the multi-scan distillation results on the SemanticKITTI dataset (Appendix S.4). Finally, we analyze the generated label distribution (Appendix S.5) and the robustness to label noise (Appendix 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 [4] and nuScenes [5] datasets, respectively. In our experiments, choosing hyper-parameters with visual inspection is convenient and sufficient to achieve satisfactory results.



For the SemanticKITTI [4] 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 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.

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 [9]	100%	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++ [39]		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 [65]		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 [50]		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 [68]		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 [31]		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 [59]		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 [23]	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 [23]	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 S8: **Comparison of different methods on the nuScenes validation set.** Cylinder3D [68] is our fully supervised counterpart.

For the nuScenes [5] 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 [5] 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 τ 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 [68].

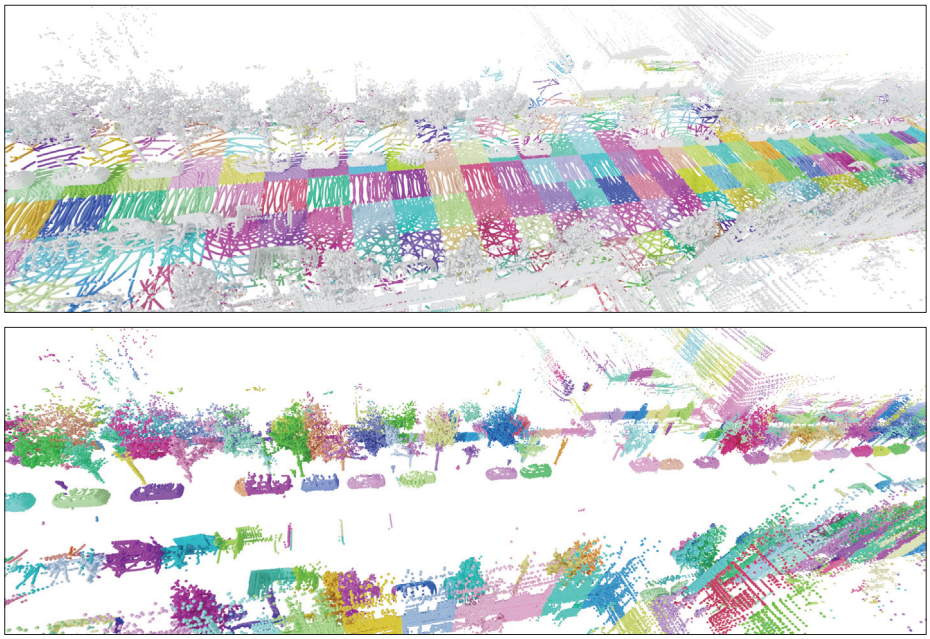


Fig. S6: **Examples of the pre-segmentation results.** First row: detected ground points of each cell. Non-ground points are colored in gray. Each other color indicates a proposed ground component. Second row: connected components of the non-ground points. Each color indicates a connected component. The example is from the nuScenes dataset, where 40 scans are fused.

Baseline Methods We adopt the author released code to train OneThingOneClick [33] and ContrastiveSceneContext [23] on SemanticKITTI and nuScenes. For other methods, the results are either obtained from the literature or correspondences with the authors.

For **ContrastiveSceneContext** [23], 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 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** [33], we first apply the geometrical partition described in [28] 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

	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 S9: **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.

	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 S10: **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.

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. S6 shows the examples of our pre-segmentation results.

S.3 Full results on nuScenes

Tab. S8 shows the full results on the nuScenes validation set.

S.4 Full table of multi-scan distillation

Tab. S11 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.

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 S11: **Results of the multi-scan distillation on the SemanticKITTI validation set.** 0.1% annotations are used.

S.5 Label distribution

Tab. S9 and Tab. S10 summarize the distributions of the generated sparse labels and the propagated labels. By leveraging our proposed pre-segmentation and 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. Alnaggar, Y.A., Afifi, M., Amer, K., ElHelw, M.: Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1800–1809 (2021) 3, 10, 11
2. Alonso, I., Riazuelo, L., Montesano, L., Murillo, A.C.: 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. IEEE Robotics and Automation Letters **5**(4), 5432–5439 (2020) 3
3. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017) 2, 4

4. 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, 3, 4, 5, 8, 10, 11, 12, 15
5. 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, 4, 10, 11, 12, 15, 16
6. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 2, 4
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 1597–1607. PMLR (2020) 8
8. Cheng, M., Hui, L., Xie, J., Yang, J., Kong, H.: Cascaded non-local neural network for point cloud semantic segmentation. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8447–8452. IEEE (2020) 4
9. 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) 1, 4, 12, 16
10. Cortinhal, T., Tzelepis, G., Aksoy, E.E.: Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In: International Symposium on Visual Computing. pp. 207–222. Springer (2020) 3
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5828–5839 (2017) 2, 4
12. Duerr, F., Pfaller, M., Weigel, H., Beyerer, J.: Lidar-based recurrent 3d semantic segmentation with temporal memory alignment. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 781–790. IEEE (2020) 3, 10, 11
13. Elsayed, G.F., Krishnan, D., Mobahi, H., Regan, K., Bengio, S.: Large margin deep networks for classification. In: Advances in Neural Information Processing Systems (NeurIPS) (2018) 9
14. Fang, Y., Xu, C., Cui, Z., Zong, Y., Yang, J.: Spatial transformer point convolution. arXiv preprint arXiv:2009.01427 (2020) 4
15. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981) 7
16. Gan, L., Zhang, R., Grizzle, J.W., Eustice, R.M., Ghaffari, M.: Bayesian spatial kernel smoothing for scalable dense semantic mapping. IEEE Robotics and Automation Letters **5**(2), 790–797 (2020) 10, 11
17. Gao, B., Pan, Y., Li, C., Geng, S., Zhao, H.: Are we hungry for 3d lidar data for semantic segmentation? ArXiv abs/2006.04307 **3**, 20 (2020) 4
18. Gao, Y., Fei, N., Liu, G., Lu, Z., Xiang, T., Huang, S.: Contrastive prototype learning with augmented embeddings for few-shot learning. arXiv preprint arXiv:2101.09499 (2021) 3, 9
19. Gerdzhev, M., Razani, R., Taghavi, E., Bingbing, L.: Tornado-net: multiview total variation semantic segmentation with diamond inception module. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 9543–9549. IEEE (2021) 3

20. Guinard, S., Landrieu, L.: Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds. In: ISPRS Workshop 2017 (2017) 2, 4, 6, 7
21. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9729–9738 (2020) 8, 9
22. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Advances in Neural Information Processing Systems (NeurIPS) (2015) 10
23. 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, 10, 11, 12, 13, 14, 16, 17
24. Hu, Q., Yang, B., Fang, G., Guo, Y., Leonardis, A., Trigoni, N., Markham, A.: Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds with 1000x fewer labels. arXiv preprint arXiv:2104.04891 (2021) 2, 10, 11
25. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11108–11117 (2020) 4
26. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2020) 9
27. Kochanov, D., Nejadasl, F.K., Booij, O.: Kprnet: Improving projection-based lidar semantic segmentation. arXiv preprint arXiv:2007.12668 (2020) 3, 10, 11
28. 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) 13, 14, 17
29. Li, J., Zhou, P., Xiong, C., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020) 3, 9
30. Li, S., Chen, X., Liu, Y., Dai, D., Stachniss, C., Gall, J.: Multi-scale interaction for real-time lidar data segmentation on an embedded platform. arXiv preprint arXiv:2008.09162 (2020) 3
31. 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) 3, 10, 11, 12, 16
32. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 2, p. 7 (2016) 9
33. 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, 4, 6, 7, 9, 10, 11, 13, 17
34. Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J.: Semantic labeling of mobile lidar point clouds via active learning and higher order mrf. IEEE Transactions on Geoscience and Remote Sensing **56**(7), 3631–3644 (2018) 2, 4
35. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., Van Der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 181–196 (2018) 8

36. Mei, J., Gao, B., Xu, D., Yao, W., Zhao, X., Zhao, H.: Semantic segmentation of 3d lidar data in dynamic scene using semi-supervised learning. IEEE Transactions on Intelligent Transportation Systems **21**(6), 2496–2509 (2019) 4
37. Mei, J., Zhao, H.: Incorporating human domain knowledge in 3-d lidar-based semantic segmentation. IEEE Transactions on Intelligent Vehicles **5**(2), 178–187 (2019) 4
38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 3111–3119 (2013) 8
39. 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) 3, 16
40. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 8
41. Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. arXiv preprint arXiv:1412.7144 (2014) 8
42. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1713–1721 (2015) 8
43. Razani, R., Cheng, R., Taghavi, E., Bingbing, L.: Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. arXiv preprint arXiv:2103.08852 (2021) 3
44. Ren, Z., Misra, I., Schwing, A.G., Girdhar, R.: 3d spatial recognition without spatially labeled 3d. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13204–13213 (2021) 2, 4, 8
45. Rist, C.B., Schmidt, D., Enzweiler, M., Gavrila, D.M.: Scssnet: Learning spatially-conditioned scene segmentation on lidar point clouds. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 1086–1093. IEEE (2020) 3
46. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015) 9
47. Shi, X., Xu, X., Chen, K., Cai, L., Foo, C.S., Jia, K.: Label-efficient point cloud semantic segmentation: An active learning approach. arXiv preprint arXiv:2101.06931 (2021) 2, 4, 6, 7
48. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2017) 3, 9
49. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1857–1865 (2016) 9
50. 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) 1, 4, 10, 11, 12, 16
51. Thomas, H., Agro, B., Gridseth, M., Zhang, J., Barfoot, T.D.: Self-supervised learning of lidar segmentation for autonomous indoor navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 14047–14053. IEEE (2021) 4
52. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 6411–6420 (2019) 4

53. Wang, H., Rong, X., Yang, L., Feng, J., Xiao, J., Tian, Y.: Weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes. arXiv preprint arXiv:2004.12498 (2020) 2, 4
54. Wei, J., Lin, G., Yap, K.H., Hung, T.Y., Xie, L.: Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4384–4393 (2020) 2, 4, 8
55. 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 International Conference on Computer Vision (ICCV). pp. 15510–15519 (2021) 2, 4, 10, 11
56. Xiao, A., Huang, J., Guan, D., Zhan, F., Lu, S.: Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation. arXiv preprint arXiv:2107.05399 (2021) 4
57. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 574–591. Springer (2020) 4
58. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 1–19. Springer (2020) 3, 10, 11
59. 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) 1, 4, 12, 16
60. Xu, K., Yao, Y., Murasaki, K., Ando, S., Sagata, A.: Semantic segmentation of sparsely annotated 3d point clouds by pseudo-labelling. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 463–471. IEEE (2019) 2, 4
61. Xu, X., Lee, G.H.: Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13706–13715 (2020) 2, 4
62. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2020) 1, 4
63. Yang, H.M., Zhang, X.Y., Yin, F., Liu, C.L.: Robust classification with convolutional prototype learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3474–3482 (2018) 3, 9
64. Zhang, F., Fang, J., Wah, B., Torr, P.: Deep fusionnet for point cloud semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 644–663. Springer (2020) 4
65. 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) 3, 10, 11, 16
66. Zhao, N., Chua, T.S., Lee, G.H.: Few-shot 3d point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8873–8882 (2021) 2, 4
67. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 9

68. 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) 1, 2, 4, 5, 10, 11, 12, 13, 14, 16
69. Zou, Y., Weinacker, H., Koch, B.: Towards urban scene semantic segmentation with deep learning from lidar point clouds: A case study in baden-württemberg, germany. Remote Sensing **13**(16), 3220 (2021) 8