Less is More: Reducing Task and Model Complexity for 3D Point Cloud Semantic Segmentation

Li Li[1] Hubert P. H. Shum[1] Toby P. Breckon[1,2]
Department of {Computer Science[1] | Engineering[2]}, Durham University, UK
{li.li4, hubert.shum, toby.breckon}@durham.ac.uk

Abstract

Whilst the availability of 3D LiDAR point cloud data has significantly grown in recent years, annotation remains expensive and time-consuming, leading to a demand for semi-supervised semantic segmentation methods with application domains such as autonomous driving. Existing work very often employs relatively large segmentation backbone networks to improve segmentation accuracy, at the expense of computational costs. In addition, many use uniform sampling to reduce ground truth data requirements for learning needed, often resulting in sub-optimal performance. To address these issues, we propose a new pipeline that employs a smaller architecture, requiring fewer ground-truth annotations to achieve superior segmentation accuracy compared to contemporary approaches. This is facilitated via a novel Sparse Depthwise Separable Convolution module that significantly reduces the network parameter count while retaining overall task performance. To effectively sub-sample our training data, we propose a new Spatio-Temporal Redundant Frame Downsampling (ST-RFD) method that leverages knowledge of sensor motion within the environment to extract a more diverse subset of training data frame samples. To leverage the use of limited annotated data samples, we further propose a soft pseudo-label method informed by LiDAR reflectivity. Our method outperforms contemporary semi-supervised work in terms of mIoU, using less labeled data, on the SemanticKITTI (59.5@5%) and ScribbleKITTI (58.1@5%) benchmark datasets, based on a 2.3× reduction in model parameters and 641× fewer multiply-add operations whilst also demonstrating significant performance improvement on limited training data (i.e., Less is More).

1. Introduction

3D semantic segmentation of LiDAR point clouds has played a key role in scene understanding, facilitating applications such as autonomous driving [6, 23, 26, 28, 46, 60, 63] and robotics [3, 38, 51, 52]. However, many contemporary methods require relatively large backbone architectures with millions of trainable parameters requiring many hundred gigabytes of annotated data for training at a significant computational cost. Considering the time-consuming and costly nature of 3D LiDAR annotation, such methods have become less feasible for practical deployment.

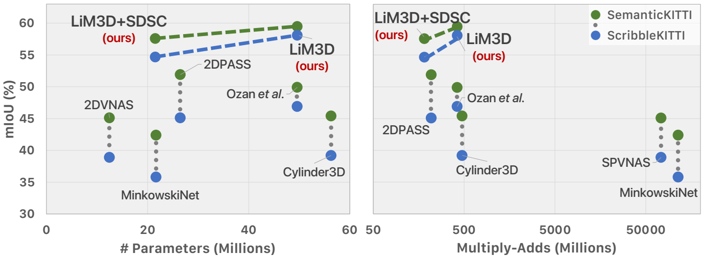

Figure 1. mIoU performance (%) against parameters and multiply-add operations on SemanticKITTI (fully annotated) and ScribbleKITTI (weakly annotated) under the 5% sampling protocol.

Existing supervised 3D semantic segmentation methods [13, 31, 38, 44, 51, 52, 56, 58, 63] primarily focus on designing network architectures for densely annotated data. To reduce the need for large-scale data annotation, and inspired by similar work in 2D [11, 47, 49], recent 3D work proposes efficient ways to learn from weak supervision [46]. However, such methods still suffer from high training costs and inferior on-task performance. To reduce computational costs, a 2D projection-based point cloud representation is often considered [3, 14, 31, 38, 51, 52, 56, 61], but again at the expense of significantly reduced on-task performance. As such, we observe a gap in the research literature for the design of semi or weakly supervised methodologies that employ a smaller-scale architectural backbone, hence facilitating improved training efficiency whilst also reducing their associated data annotation requirements.

In this paper, we propose a semi-supervised methodology for 3D LiDAR point cloud semantic segmentation. Facilitated by three novel design aspects, our *Less is More* (LiM) based methodologies require *less* training data and *less* training computation whilst offering (*more*) improved accuracy over contemporary state-of-the-art approaches (see Fig. 1).

Firstly, from an architectural perspective, we propose a novel **Sparse Depthwise Separable Convolution (SDSC)** module, which substitutes traditional sparse 3D convolution into existing 3D semantic segmentation architectures, resulting in a significant reduction in trainable parameters and numerical computation whilst maintaining on-task performance (see Fig. 1). Depthwise Separable Convolution has shown to be very effective within image classification tasks [12]. Here, we tailor a sparse variant of 3D Depthwise Separable Convolution for 3D sparse data by first applying a single submanifold sparse convolutional filter [20, 21] to each input channel with a subsequent pointwise convolution to create a linear combination of the sparse depthwise convolution outputs. This work is the first to attempt to introduce depthwise convolution into the 3D point cloud segmentation field as a conduit to reduce model size. Our SDSC module facilitates a 50% reduction in trainable network parameters without any loss in segmentation performance.

Secondly, from a training data perspective, we propose a novel **Spatio-Temporal Redundant Frame Downsampling (ST-RFD)** strategy that more effectively sub-samples a set of diverse frames from a continuously captured LiDAR sequence in order to maximize diversity within a minimal training set size. We observe that continuously captured LiDAR sequences often contain significant temporal redundancy, similar to that found in video [2], whereby temporally adjacent frames provide poor data variation. On this basis, we propose to compute the temporal correlation between adjacent frame pairs, and use this to select the most informative sub-set of LiDAR frames from a given sequence. Unlike passive sampling (*e.g*., uniform or random sampling), our active sampling approach samples frames from each sequence such that redundancy is minimized and hence training set diversity is maximal. When compared to commonplace passive random sampling approaches [29, 32, 46], ST-RFD explicitly focuses on extracting a diverse set of training frames that will hence maximize model generalization.

Finally, in order to employ semi-supervised learning, we propose a soft pseudo-label method informed by the LiDAR reflectivity response, thus maximizing the use of any annotated data samples. Whilst directly using unreliable soft pseudo-labels generally results in performance deterioration [5], the voxels corresponding to the unreliable predictions can instead be effectively leveraged as negative samples of unlikely categories. Therefore, we use cross-entropy to separate all voxels into two groups, *i.e*., a reliable and an unreliable group with low and high-entropy voxels respectively. We utilize predictions from the reliable group to derive positive pseudo-labels, while the remaining voxels from the unreliable group are pushed into a FIFO category-wise memory bank of negative samples [4]. To further assist semantic segmentation of varying materials in the situation where we have weak/unreliable/no labels, we append the reflectivity response features onto the point cloud features, which again improve segmentation results.

We evaluate our method on the SemanticKITTI [7] and ScribbleKITTI [46] *validation* set. Our method outperforms contemporary state-of-the-art semi- [29,32] and weakly- [46] supervised methods and offers *more* in terms of performance on limited training data, whilst using *less* trainable parameters and *less* numerical operations (*Less is More*).

Overall, our contributions can be summarized as follows:

- A novel methodology for semi-supervised 3D LiDAR semantic segmentation that uses significantly *less* parameters and offers (*more*) superior accuracy.[1]
- A novel Sparse Depthwise Separable Convolution (SDSC) module, to reduce trainable network parameters, and to both reduce the likelihood of over-fitting and facilitate a deeper network architecture.
- A novel Spatio-Temporal Redundant Frame Downsampling (ST-RFD) strategy, to extract a maximally diverse data subset for training by removing temporal redundancy and hence future annotation requirements.
- A novel soft pseudo-labeling method informed by LiDAR reflectivity as a proxy to in-scene object material properties, facilitating effective use of limited data annotation.

2. Related Work

Semi-supervised learning (SSL) LiDAR semantic segmentation is a special instance of weak supervision that combines a small amount of labeled, with a large amount of unlabeled point cloud during training. Numerous approaches have been explored for LiDAR semantic segmentation. Projection-based approaches [31, 32, 36, 38, 51, 52, 56] make full use of 2D-convolution kernels by using range or other 2D image-based spherical coordinate representations of point clouds. Conversely, voxel-based approaches [32, 44, 46, 63] transform irregular point clouds to regular 3D grids and then apply 3D convolutional neural networks with a better balance of the efficiency and effectiveness. Pseudo-labeling is generally applied to alleviate the side effect of intra-class negative pairs in feature learning from the teacher network [29, 32, 46, 57]. However, such methods only utilize samples with reliable predictions and thus ignore the valuable information that unreliable predictions carry. In our work, we combined a novel SSL framework with the mean teacher paradigm [45], demonstrating the utilization of unreliable pseudo-labels to improve segmentation performance.
Depthwise separable convolution [41] is a depthwise convolution followed by a pointwise convolution, to reduce both model size and complexity. Being a more computationally efficient alternative than standard convolution, it is used for mobile applications [24, 25, 40] and hardware accelerators [37]. Furthermore, it is a building block of Xception [12], a deep convolutional neural network architecture that achieves state-of-the-art performance on the ImageNet [15] classification task, via more efficient use of model parameterization. In this work, we propose a novel sparse variant of depthwise separable convolution, which has both the efficiency advantages of depthwise separable convolution and those of sparse convolution for processing spatially-sparse data [20].

[1] Full source code: `https://github.com/l1997i/lim3d/`.

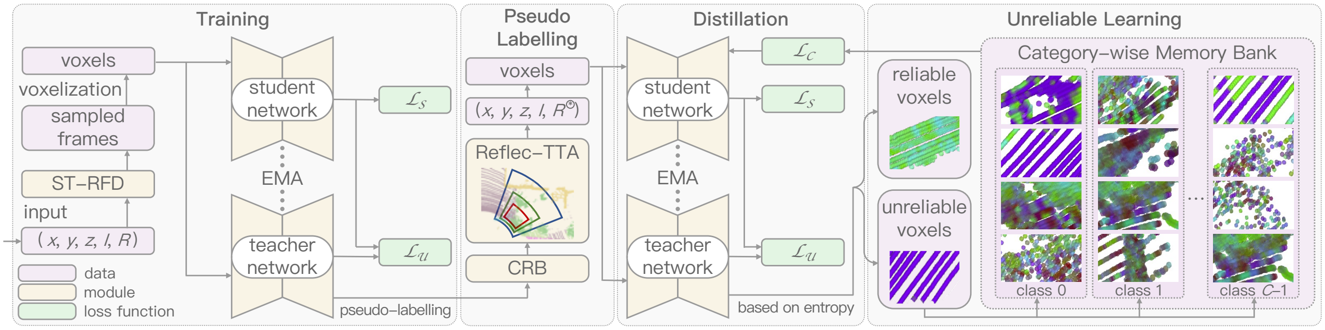

Figure 2. Our proposed architecture for unreliable pseudo-labels LiDAR semantic segmentation involves three stages: training, pseudo-labeling, and distillation with unreliable learning. We apply ST-RFD sampling before training the Mean Teacher on available annotations.

Temporal redundancy is highly prevalent within video [48, 62] and radar [35] sequences alike. Existing semi-supervised 3D LiDAR segmentation methods [32, 46] utilize a passive uniform sampling strategy to filter unlabeled points from a fully-labeled point cloud dataset. Active learning frameworks handle the redundancy to reduce annotation or training efforts by selecting informative and diverse sub-scenes for label acquisition [16, 27, 53]. We propose a novel temporal-redundancy-based sampling strategy with comparable time cost to uniform sampling, to reduce the inter-frame spatio-temporal redundancy and maximize data diversity.

3. Methodology

We first present an overview of the mean teacher framework we employ (Sec. 3.1) and then explain our use of unreliable pseudo-labels informed by LiDAR reflectivity for semi-supervised learning (Sec. 3.2). Subsequently, we detail our ST-RFD strategy for dataset diversity (Sec. 3.4) and finally our parameter-reducing SDSC module (Sec. 3.5).

Formally, given a LiDAR point cloud $P = \{\mathbf{p} \mid \mathbf{p} = (x, y, z, I, R) \in \mathbb{R}^5\}$ where (x, y, z) is a 3D coordinate, I is intensity and R is reflectivity, our goal is to train a semantic segmentation model by leveraging both a large amount of unlabeled $U = \{\mathbf{p}_i^u\}_{i=1}^{N_u} \subsetneq P$ and a smaller set of labeled data $V = \{(\mathbf{p}_i^v, \mathbf{y}_i^v)\}_{i=1}^{N_v} \subsetneq P$.

Our overall architecture involves three stages (Fig. 2): (1) **Training:** we utilize reflectivity-prior descriptors and adapt the Mean Teacher framework to generate high-quality pseudo-labels; (2) **Pseudo-labeling:** we fix the trained teacher model prediction in a class-range-balanced [46] manner, expanding dataset with Reflectivity-based Test Time Augmentation (Reflec-TTA) during test time; (3) **Distillation with unreliable predictions:** we train on the generated pseudo-labels, and utilize unreliable pseudo-labels in a category-wise memory bank for improved discrimination.

3.1. Mean Teacher Framework

We introduce weak supervision using the Mean Teacher framework [45], which avoids the prominent slow training issues associated with Temporal Ensembling [33]. This framework consists of two models of the same architecture known as the student and teacher respectively, for which we utilize a Cylinder3D [63]-based segmentation head f. The weights of the student model θ are updated via standard backpropagation, while the weights of the teacher model θ^* are updated by the student model through Exponential Moving Averaging (EMA):

$$\theta^*_{t+1} = \kappa\theta^*_t + (1-\kappa)\theta_{t+1}, \quad t \in \{0, 1, \cdots T-1\}, \quad (1)$$

where κ denotes a smoothing coefficient to determine update speed, and T is the maximum time step.

During training, we train a set of weakly-labeled point cloud frames with voxel-wise inputs generated via asymmetrical 3D convolution networks [63]. For every point cloud, our optimization target is to minimize the overall loss:

$$\mathcal{L} = \mathcal{L}_S + \lambda_U \mathcal{L}_U + g\lambda_C \mathcal{L}_C, \quad (2)$$

where $\mathcal{L}_S$ and $\mathcal{L}_U$ denote the losses applied to the supervised and unsupervised set of points respectively, $\mathcal{L}_C$ denotes the contrastive loss to make full use of unreliable pseudo-labels, λ_U is the weight coefficient of $\mathcal{L}_U$ to balance the losses, and g is the gated coefficient of $\mathcal{L}_C$. g equals λ_C if and only if it is in the distillation stage. We use the consistency loss (implemented as a Kullback-Leibler divergence loss [23]), lovasz softmax loss [8], and the voxel-level InfoNCE [47] as $\mathcal{L}_U$, $\mathcal{L}_S$ and $\mathcal{L}_C$ respectively.

We first generate our pseudo-labels for the unlabeled points via the teacher model. Subsequently, we generate reliable pseudo-labels in a class-range-balanced (CRB) [46] manner, and utilize the qualified unreliable pseudo-labels as negative samples in the distillation stage. Finally, we train the model with both reliable and qualified unreliable pseudo-labels to maximize the quality of the pseudo-labels.

3.2. Learning from Unreliable Pseudo-Labels

Unreliable pseudo-labels are frequently eliminated from semi-supervised learning tasks or have their weights decreased to minimize performance loss [29, 39, 46, 55, 59, 65].

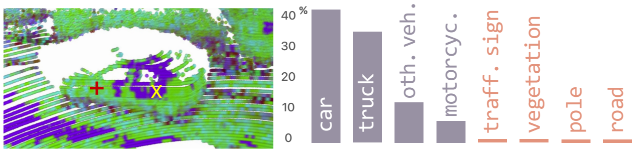

Figure 3. Illustration on unreliable pseudo-labels. Left: entropy predicted from an unlabeled point cloud, with lower entropy corresponding to greener color. Right: Category-wise probability of an unreliable prediction X, only top-4 and last-4 probabilities shown.

In line with this idea, we utilize CRB method [46] to first mask off unreliable pseudo-labels and then subsequently generate high-quality reliable pseudo-labels.

However, such a simplistic discarding of unreliable pseudo-labels may lead to valuable information loss as it is clear that unreliable pseudo-labels (*i.e.*, the corresponding voxels with high entropy) can offer information that helps in discrimination. Voxels that correlate to unreliable predictions can alternatively be thought as negative samples for improbable categories [49], although performance would suffer if such unreliable predictions are used as pseudo-labels directly [5]. As shown in Fig. 3, the unreliable pseudo predictions show a similar level of confidence on `car` and `truck` classes, whilst being sure the voxel cannot be `pole` or `road`. Thus, together with the use of CRB for high-quality reliable pseudo-labels, we also ideally want to make full use of these remaining unreliable pseudo-labels rather than simply discarding them. Following [49], we propose a method to leverage such unreliable pseudo-labels for 3D voxels as negative samples. However, to maintain a stable amount of negative samples, we utilize a category-wise memory bank $\mathcal{Q}_c$ (FIFO queue, [54]) to store all the negative samples for a given class c. As negative candidates in some specific categories are severely limited in a mini-batch due to the long-tailed class distribution of many tasks (*e.g.* autonomous driving), without such an approach in place we may instead see the gradual dominance of large and simple-to-learn classes within our generated pseudo-labels.

Following [22, 47], our method has three prerequisites, *i.e.*, anchor voxels, positive candidates, and negative candidates. They are obtained by sampling from a particular subset, constructed via Eq. (3) and Eq. (4), in order to reduce overall computation. In particular, the set of features of all candidate anchor voxels for class c is denoted as:

$$\mathcal{A}_c = \left\{ \mathbf{E}_{a,b} \mid y^*_{a,b} = c, p_{a,b}(c) > \delta_p \right\}, \tag{3}$$

where $\mathbf{E}_{a,b}$ is the feature embedding for the a-th point cloud frame at voxel b, δ_p is the positive threshold of all classes, $p_{a,b}(c)$ is the softmax probability by the segmentation head at c-th dimension. $y^*_{a,b}$ is set to the ground truth label $y^*_{a,b}$ if the ground truth is available, otherwise, $y^*_{a,b}$ is set to the pseudo label $\hat{y}_{a,b}$, due to the absence of ground truth.

The positive sample is the common embedding center of all possible anchors, which is the same for all anchors from the same category, shown in Eq. (4).

$$\mathbf{E}^+_\mathbf{c} = \frac{1}{|\mathcal{A}_c|} \sum_{\mathbf{E}_c \in \mathcal{A}_c} \mathbf{E}_c. \tag{4}$$

Following [49], we similarly construct multiple negative samples $\mathbf{E}^-_c$ for each anchor voxel.

Finally, for each anchor voxel containing one positive sample and $N-1$ negative samples, we propose the voxel-level InfoNCE loss [47] (a variant of contrastive loss) $\mathcal{L}_C$ in Eq. (5) to encourage maximal similarity between the anchor voxel and the positive sample, and the minimal similarity between the anchor voxel and multiple negative samples.

$$\begin{aligned} \mathcal{L}_C &= -\frac{1}{C} \sum_{c=0}^{C-1} \mathop{\mathbb{E}}_{\mathbf{E}_c} \left[\log \frac{f\left(\mathbf{e}_c, \mathbf{e}^+_c, \tau\right)}{\sum_{\mathbf{e}^-_{c,j} \in \mathbf{E}^-_c} f\left(\mathbf{e}_c, \mathbf{e}^-_{c,j}, \tau\right)} \right] \\ &= -\frac{1}{C} \sum_{c=0}^{C-1} \mathop{\mathbb{E}}_{\mathbf{E}_c} \left[\log \frac{\exp\left(\left\langle \mathbf{e}_c, \mathbf{e}^+_c \right\rangle / \tau\right)}{\exp\left(\left\langle \mathbf{e}_c, \mathbf{e}^+_c \right\rangle / \tau\right) + \sum_{j=1}^{N-1} \exp\left(\left\langle \mathbf{e}_c, \mathbf{e}^-_{c,j} \right\rangle / \tau\right)} \right] \end{aligned} \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity. $\mathbf{e_c}$, $\mathbf{e^+_c}$ and $\mathbf{e}^-_{c,j}$ denote the embedding, positive sample of the current anchor voxel, and embedding of the j-th negative sample of class c.

3.3. Reflectivity-Based Test Time Augmentation

To obtain minimal accuracy degradation despite very few weak labels, *e.g.*, 1% weakly-labeled ScribbleKITTI [46] dataset, we propose a Test Time Augmentation (TTA) that does not depend on any label, but only relies on a feature of original LiDAR points themselves. Also included in almost every LiDAR benchmark dataset for autonomous driving [7, 10, 18, 30, 34, 46], is the intensity of light reflected from the surface of an object at each point. In the presence of limited data labels in the semi-supervised learning case, this property of the material surface, normalized by distance to obtain the surface reflectivity in Eq. (6), could readily act as auxiliary information to identify different semantic classes.

Our intuition is that reflectivity R, as a point-wise distance-normalized intensity feature, offers consistency across lighting conditions and range as:

$$R = Ir^2 = \frac{S}{4\pi r^2} \cdot r^2 \propto S, \tag{6}$$

where S is the return strength of the LiDAR laser pulse, I is the intensity and r is the point distance from the source on the basis that scene objects with similar surface material, coating, and color characteristics will share similar S returns.

On this basis, we define our novel reflectivity-based Test Time Augmentation (Reflec-TTA) technique, as a substitute for label-dependent Pyramid Local Semantic-context

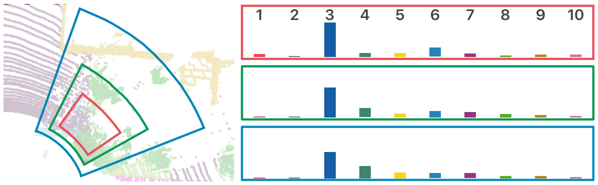

Figure 4. Coarse histograms of Reflec-TTA bins (not to scale).

(PLS) augmentation [46] during test-time as ground truth is not available. We append our point-wise reflectivity to the existing point features in order to enhance performance in presence of false or non-existent pseudo-labels at the distillation stage. As shown in Fig. 4, and following [46], we apply various sizes s of bins in cylindrical coordinates to analyze the intrinsic point distribution of the LiDAR sensor at varying resolutions (shown in red, green and blue in Fig. 4). For each bin b_i, we compute a coarse histogram, $\mathbf{h}_i$:

$$\begin{aligned} \mathbf{h}_i \quad &= \left\{ h_i^{(k)} \mid k \in [1, N_b] \right\} \in \mathbb{R}^{N_b}, \quad i \in [1, s], \\ h_i^{(k)} &= \# \left\{ R_j \in r_k, \ \forall j \mid p_j \in b_i \right\}, \\ r_k \quad &= [\ (k-1)/N_b, \ k/N_b \ ), \quad k \in [\ 1, N_b\ ]. \end{aligned} \tag{7}$$

The Reflec-TTA features $R^{\circledast}$ of all points $p_j \in b_j$ is further computed as the concatenation of the coarse histogram $\mathbf{h}_i$ of the normalized histogram:

$$R^{\circledast} = \{ \mathbf{h}_i / \max(\mathbf{h}_i) \mid i \in [1, s] \} \in \mathbb{R}^{sN_b} \tag{8}$$

In the distillation stage, we append $R^{\circledast}$ to the input features and redefine the input LiDAR point cloud as the augmented set of points $P^{\circledast} = \left\{ p \mid (x, y, z, I, R^{\circledast}) \in \mathbb{R}^{sN_b+4} \right\}$.

3.4. Spatio-Temporal Redundant Frame Downsampling

Due to the spatio-temporal correlation of LiDAR point cloud sequences often captured from vehicles in metropolitan locales, many large-scale point cloud datasets demonstrate significant redundancy. Common datasets employ a frame rate of 10Hz [7, 9, 10, 18, 30, 34, 42], and a number of concurrent laser channels (beams) of 32 [10], 64 [7, 9, 18, 30, 42] or 128 [34]. Faced with such large-scale, massively redundant training datasets, the popular practice of semi-supervised semantic segmentation approaches [11, 17, 32, 45, 64] is to uniformly sample 1%, 10%, 20%, or 50% of the available annotated training frames, without considering any redundancy attributable to temporary periods of stationary capture (*e.g.* due to traffic, Fig. 5) or multi-pass repetition (*e.g.* due to loop closure).

To extract a diverse set of frames, we propose a novel algorithm called Spatio-Temporal Redundant Frame Downsampling (ST-RFD, Algorithm 1) that determines spatio-temporal redundancy by analyzing the spatial-overlap within time-continuous LiDAR frame sequences. The key idea is that if spatial-overlap among some continuous frame sequence is high due to spatio-temporal redundancy, multiple representative frames can be sub-sampled for training, significantly reducing both training dataset size and redundant training computation.

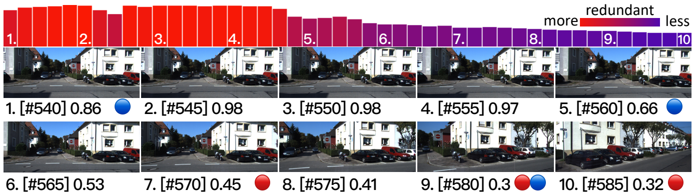

Figure 5. Illustration of LiDAR frame temporal correlation as *[# frame ID] redundancy* with 5% sampling on SemanticKITTI [7] (sequence `00`) using uniform sampling (selected frames in ●) and ST-RFD strategy (●).

Fig. 6 shows an overview of ST-RFD. It is conducted inside each temporal continuous LiDAR sequence e. First, we evenly divide p point cloud frames in each sequence into $\lceil p/q \rceil$ subsets (containing q frames). For each frame at time t inside the subset, we find its corresponding RGB camera image in the dataset at time t and $t+1$. To detect the spatio-temporal redundancy at time t, the similarity $\psi(t, t+1)$ between temporally adjacent frames are then computed via the Structural Similarity Index Measure (SSIM, [50]). We utilize the mean value of similarity scores between all adjacent frames in the current subset as a proxy to estimate the spatio-temporal redundancy present. A sampling rate is then determined according to this mean similarity for frame selection within this subset. This is repeated for all subsets in every sequence to construct our final set of sub-sampled LiDAR frames for training.

Concretely, as shown in Algorithm 1, we implement a ST-RFD supervisor that determines the most informative assignments (*i.e.*, the key point cloud frames) that the teacher and student networks should train on respectively. The ST-RFD supervisor has an empirical supervisor function υ, which decides the amount of assignments, *i.e.*, the sampling rate corresponding to the extent of spatio-temporal redundancy. Using SSIM [50] as the redundancy function ψ to measure the similarity between the RGB images associated with two adjacent point clouds, we define the empirical supervisor function υ with decay property $\upsilon(x) = \exp(-\beta x)$, where $\beta \in (0, +\infty)$ is the decay coefficient, and x is the redundancy calculated from ψ. In this way, the higher the degree of spatio-temporal redundancy (as $\psi \to 1$), the lower the sampling rate our ST-RFD supervisor will allocate, hence reducing the training set requirements for teacher and student alike.

3.5. Sparse Depthwise Separable Convolution

Existing LiDAR point cloud semantic segmentation methods generally rely on a large-scale backbone architecture with

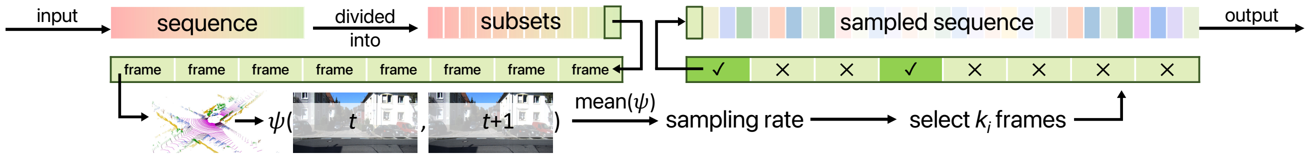

Figure 6. Overview of our proposed Spatio-Temporal Redundant Frame Downsampling approach.

Algorithm 1: Spatio-Temporal Redundant Frame Downsampling.

```
Input: Point cloud frames pool $P$ (size of $p$), subset size $q$,
       redundancy function $\psi \in [0, 1]$ and empirical supervisor
       function $\upsilon$.
Divide $P$ evenly into $\lceil p/q \rceil$ subsets $Q$.
$D \leftarrow$ empty dictionary.
forall $e \leftarrow 0 : n_e - 1$ do
    // loop for all sequences
    $C_e \leftarrow \varnothing$          // chosen point cloud frames
    forall $i \leftarrow 0 : \lceil p/q \rceil - 1$ do
        // loop for subsets $Q$
        $Q_{i,j} \leftarrow j$-th frame in subset $Q_i$.
        $\overline{M_i} \leftarrow \frac{1}{q} \sum_{j=0}^{q-1} \psi(Q_{i,j})$.          // redundancy
        $k_i \leftarrow \lceil \upsilon(\overline{M_i}) \cdot q \rceil$.
        $T_i \leftarrow$ select $k_i$ frames in $Q_i$ with the smallest $M_i$.
        $C_e \leftarrow C_e \cup T_i$.
    Append key-value pair $(e, C_e)$ into $D$.
Return: Dictionary $D$.
```

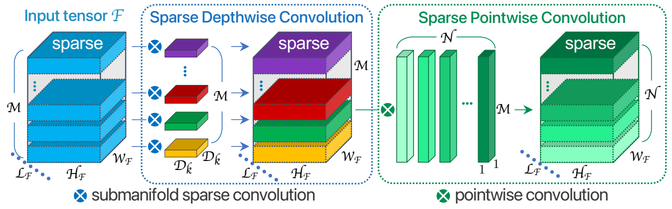
Input tensor $\mathcal{F}$
Sparse Depthwise Convolution
Sparse Pointwise Convolution
sparse
sparse
sparse
$\mathcal{M}$
$\mathcal{M}$
$\mathcal{M}$
$\mathcal{M}$
$\mathcal{N}$
$\mathcal{N}$
$\mathcal{W}_\mathcal{F}$
$\mathcal{W}_\mathcal{F}$
$\mathcal{W}_\mathcal{F}$
$\mathcal{L}_\mathcal{F}$
$\mathcal{L}_\mathcal{F}$
$\mathcal{L}_\mathcal{F}$
$\mathcal{H}_\mathcal{F}$
$\mathcal{H}_\mathcal{F}$
$\mathcal{H}_\mathcal{F}$
$\mathcal{D}_k$
$\mathcal{D}_k$
1
1
submanifold sparse convolution
pointwise convolution

Figure 7. Illustration of the SDSC convolution module.

tens of millions of trainable parameters [23,26,28,46,60,63] due to the requirement for 3D (voxel-based) convolution operations, to operate on the voxelized topology of the otherwise unstructured LiDAR point cloud representation, which suffer from both high computational training demands and the risk of overfitting. Based on the observation that depthwise separable convolution has shown results comparable with regular convolution in tasks such as image classification but with significantly fewer trainable parameters [12,24,25,37,40,43], here we pursue the use of such an approach within 3D point cloud semantic segmentation.

As such we propose the first formulation of sparse variant depthwise separable convolution [25] applied to 3D point clouds, namely Sparse Depthwise Separable Convolution (SDSC). SDSC combines the established computational advantages of sparse convolution for point cloud segmentation [20], with the significant trainable parameter reduction offered by depthwise separable convolution [12].

Our SDSC module, as outlined in Fig. 7, initially takes a tensor $F \in \mathbb{R}^{H_F \times W_F \times L_F \times M}$ as input, where H_F, W_F, L_F and M denote radius, azimuth, height in the cylinder coordinate [63] and channels respectively. Firstly, a sparse depthwise convolution SDC$(M, M, D_k, s = 1)$ is applied, with M input and output feature planes, a kernel size of D_k and stride s in order to output a tensor $T \in \mathbb{R}^{H_F \times W_F \times L_F \times M}$. Inside our sparse depthwise convolution, M sparse spatial convolutions are performed independently over each input channel using submanifold sparse convolution [20] due to its tensor shape preserving property at no computational or memory overhead. Secondly, the sparse pointwise convolution SPC$(M, N, 1, s = 1)$ projects the channels output T by the sparse depthwise convolution onto a new channel space, to mix the information across different channels. As a result, the sparse depthwise separable convolution SDSC$(M, N, D_k, s = 1)$ is the compound of the sparse depthwise convolution and the sparse pointwise convolution, namely SDSC$(M, N, D_k, s = 1) =$ SDC $\circ$ SPC.

Using a sparse voxelized input representation similar to [19], and a series of such SDSC sub-modules we construct the popular Cylinder3D [63] sub-architectures within our overall Mean Teacher architectural design (Fig. 2).

4. Evaluation

We evaluate our proposed *Less is More 3D* (LiM3D) approach against state-of-the-art 3D point cloud semantic segmentation approaches using the SemanticKITTI [7] and ScribbleKITTI [46] benchmark datasets.

4.1. Experimental Setup

SemanticKITTI [7] is a large-scale 3D point cloud dataset for semantic scene understanding with 20 semantic classes consisting of 22 sequences - [`00` to `10` as *training*-split (of which `08` as *validation*-split) + `11` to `21` as *test*-split].
ScribbleKITTI [46] is the first scribble (*i.e.* sparsely) annotated dataset for LiDAR semantic segmentation providing sparse annotations for the *training* split of SemanticKITTI for 19 classes, with only 8.06% of points from the full SemanticKITTI dataset annotated.
Evaluation Protocol: Following previous work [29,32,46,63], we report performance on the SemanticKITTI and ScribbleKITTI *training* set for intermediate training steps, as this metric provides an indication of the pseudo-labeling quality, and on the *validation* set to assess the performance benefits of each individual component. Performance is reported using the mean Intersection over Union (mIoU, as %) metric.

Table 1. Comparative mIoU for Range- and Voxel-based methods using uniform sampling (U), sequential partition (P) and ST-RFD sampling (S): **bold**/underlined = **best**/2nd best; * denotes reproduced result; – denotes missing result due to unavailability from original authors.

Repr.	Samp.	Method		SemanticKITTI [7] 1%	5%	10%	20%	40%	50%	100%	ScribbleKITTI [46] 1%	5%	10%	20%	40%	50%	100%
Range	U	LaserMix [32]	(2022)	43.4	–	58.8	59.4	–	61.4	–	38.3	–	54.4	55.6	–	58.7	–
Voxel	U	Cylinder3D [63]	(CVPR'21)	–	45.4	56.1	57.8	58.7	–	67.8	–	39.2	48.0	52.1	53.8	–	56.3
	U	LaserMix [32]	(2022)	50.6	–	60.0	61.9	–	62.3	–	44.2	–	53.7	55.1	–	56.8	–
	P	Jiang *et al.* [29]	(ICCV'21)	–	41.8	49.9	58.8	59.9	–	65.8	–	–	–	–	–	–	–
	U	Unal *et al.* [46]	(CVPR'22)	–	49.9*	58.7*	59.1*	60.9	–	68.2*	–	46.9*	54.2*	56.5*	58.6*	–	61.3
	S	LiM3D+SDSC	(ours)	57.2	57.6	61.0	61.7	62.1	62.7	67.5	55.8	56.1	56.9	57.2	58.9	59.3	60.7
	S	LiM3D	(ours)	**58.4**	**59.5**	**62.2**	**63.1**	**63.3**	**63.6**	**69.5**	**57.0**	**58.1**	**61.0**	**61.2**	**62.0**	**62.1**	**62.4**

For semi-supervised training, we report over both the benchmarks using the SemanticKITTI and ScribbleKITTI *validation* set under 5%, 10%, 20%, and 40% partitioning. We further report the relative performance of semi-supervised or scribble-supervised for ScribbleKITTI (SS) training to the fully supervised upper-bound (FS) in percentages (SS/FS) to further analyze semi-supervised performance and report the results for the fully-supervised training on both *validation* sets for reference. The trainable parameter count and number of multiply-adds (multi-adds) are additionally provided as a metric of computational cost.

Implementation Details: Training is performed using 4× NVIDIA A100 80GB GPU without pre-trained weights with a DDP shared training strategy [1] to maintain GPU scaling efficiency, whilst reducing memory overhead significantly. Specific hyper-parameters are set as follows - Mean Teacher: $\kappa = 0.99$; unreliable pseudo-labeling: $\lambda_C = 0.3$, $\tau = 0.5$; ST-RFD: $\beta = \{7.45, 5.72, 4.00, 2.28, 0\}$ for sampling $\{5\%$, $10\%, 20\%, 40\%, 100\%\}$ labeled training frames, assuming the remainder as unlabeled; Reflec-TTA: $N_b = 10$, $s = 3$ various Reflec-TTA bin sizes, following [46], we set each bin $b_i = (\rho, \phi) \in \{(20, 40), (40, 80), (80, 120)\}$.

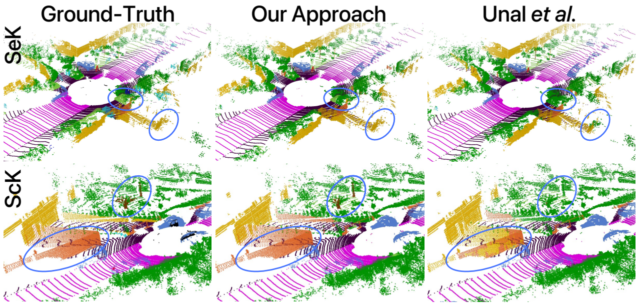

Figure 8. Comparing the 10% sampling split of SemanticKITTI (SeK, first row) and ScribbleKITTI (ScK, second row) *validation* set with ground-truth (left), our approach (middle) and Unal *et al.* [46] (right) with areas of improvement highlighted.

4.2. Experimental Results

In Tab. 1, we present the performance of our *Less is More* 3D (LiM3D) point cloud semantic segmentation approach both with (LiM3D+SDSC) and without (LiM3D) SDSC in a side-by-side comparison with leading contemporary state-of-the-art approaches on the SemanticKITTI and ScribbleKITTI benchmark *validation* sets to illustrate our approach offers superior or comparable (within 1% mIoU) performance across all sampling ratios. Furthermore, we present supporting qualitative results in Fig. 8.

Table 2. Component-wise ablation of LiM3D (mIoU as %, and #parameters in millions, M) on SemanticKITTI [7] *training* and *validation* sets where UP, RF, RT, ST, SD denote Unreliable Pseudo-labeling, Reflectivity Feature, Reflec-TTA, ST-RFD, and SDSC module respectively.

UP	RF	RT	ST	SD	Training mIoU (%) 5%	10%	20%	40%	Validation mIoU (%) 5%	10%	20%	40%	#Params (M)
					82.8	87.5	87.8	88.2	54.8	58.1	59.3	60.8	49.6
✓					–	–	–	–	55.9	58.8	59.9	61.2	49.6
✓	✓				83.6	88.3	88.7	89.1	56.8	59.6	60.5	61.4	49.6
✓		✓			–	–	–	–	57.5	59.8	61.2	62.6	49.6
✓	✓	✓			–	–	–	–	58.7	61.3	62.4	62.8	49.6
✓	✓	✓	✓		**85.2**	**89.1**	**89.5**	**89.7**	**59.5**	**62.2**	**63.1**	**63.3**	49.6
✓	✓	✓	✓	✓	83.8	88.6	89.0	89.2	57.6	61.0	61.7	62.1	**21.5**

On SemanticKITTI, with a lack of available supervision, LiM3D shows a relative performance (SS/FS) from 85.6% (5%-fully-supervised) to 91.1% (40%-fully-supervised), and LiM3D+SDSC from 85.3% to 92.0%, compared to their respective fully supervised upper-bound. LiM3D/LiM3D+SDSC performance is also less sensitive to reduced labeled data sampling compared with other methods.

Our model significantly outperforms on small ratio sampling splits, *e.g.*, 5% and 10%. LiM3D shows up to 19.8% and 18.9% mIoU improvements whilst, with a smaller model size LiM3D+SDSC again shows significant mIoU improvements by up to 16.4% and 15.5% when compared with other range and voxel-based methods respectively.

4.3. Ablation Studies

Effectiveness of Components. In Tab. 2 we ablate each component of LiM3D step by step and report the performance on the SemanticKITTI *training* set at the end of training as an overall indicator of pseudo-labeling quality in addition to the corresponding *validation* set.

As shown in Tab. 2, adding unreliable pseudo-labeling (UP) in the distillation stage, we can increase the $valid$ mIoU by $+0.7\%$ on average in *validation* set. Appending reflectivity features (RF) in the training stage, we further improve the mIoU on the *training* set by $+0.7\%$ on average. Due to the improvements in training, the model generates a higher quality of pseudo-labels, which results to a $+0.5\%$ increase in mIoU in the *validation* set. If we disable reflectivity features in the training stage, applying Reflec-TTA in the distillation stage alone, we then get an average improvement of $+1.3\%$ compared with pseudo-labeling only. On the whole, enabling all reflectivity-based components (RF+RT) shows great improvements of up to $+2.8\%$ in $validation$ mIoU.

Table 3. The computation cost and mIoU (in percentage) under 5%-labeled training results on SemanticKITTI (SeK) and ScribbleKITTI (ScK) *validation* set.

Method	# Parameters	# Mult-Adds	SeK [7]	ScK [46]
Cylider3D [63]	56.3	476.9M	45.4	39.2
Unal *et al.* [46]	49.6	420.2M	49.9	46.9
2DPASS [58]	26.5	217.4M	51.7	45.1
MinkowskiNet [13]	21.7	114.0G	42.4	35.8
SPVNAS [44]	**12.5**	73.8G	45.1	38.9
LiM3D+SDSC (ours)	21.5	**182.0M**	57.6	54.7
LiM3D (ours)	49.6	420.2M	**59.5**	**58.1**

Table 4. Effects of ST-RFD sampling on SemanticKITTI and ScribbleKITTI *validation* set (mIoU as %).

Sampling	SemanticKITTI [7]				ScribbleKITTI [46]			
	5%	10%	20%	40%	5%	10%	20%	40%
Random	58.5	61.6	62.6	62.7	57.1	60.3	60.5	60.9
Uniform	58.7	61.3	62.4	62.8	56.9	60.6	60.3	61.0
ST-RFD-R	59.1	**62.4**	62.9	**63.4**	58.0	60.7	**61.2**	61.8
ST-RFD	**59.5**	62.2	**63.1**	63.3	**58.1**	**61.0**	**61.2**	**62.0**

Table 5. Effects of differing reliability using pseudo voxels on SemanticKITTI *validation* set, measured by the entropy of voxel-wise prediction. *Unreliable* and *Reliable*: selecting negative candidates with top 20% highest entropy scores and bottom 20% counterpart respectively. *Random*: sampling randomly regardless of entropy.

Ratio	Unreliable		Reliable		Random	
	mIoU	SS/FF	mIoU	SS/FF	mIoU	SS/FF
5%	**59.5**	**85.6**	57.2	82.3	56.4	81.2
10%	**62.2**	**89.5**	60.8	87.5	59.7	85.9
20%	**63.1**	**90.8**	61.4	88.3	60.5	87.1
40%	**63.3**	**91.1**	62.8	90.4	61.3	88.2

Table 6. Reflectivity (Reflec-TTA) vs. Intensity (intensity-based TTA) on SemanticKITTI and ScribbleKITTI *validation* set (mIoU, %).

TTA	SemanticKITTI [7]				ScribbleKITTI [46]			
	5%	10%	20%	40%	5%	10%	20%	40%
Intensity	56.2	59.1	59.8	60.9	55.7	57.5	57.9	59.2
Reflectivity	**59.5**	**62.2**	**63.1**	**63.3**	**58.1**	**61.0**	**61.2**	**62.0**

Substituting the uniform sampling with our ST-RFD strategy, we observe further average improvements of $+1.0\%$ and $+0.8\%$ on $training$ and $validation$ respectively (Tab. 2).

Our SDSC module reduces the trainable parameters of our model by 57%, with a performance cost of -0.7% and -1.4% mIoU on $training$ and $validation$ respectively (Tab. 2). Finally, we provide two models, one without SDSC (LiM3D) and one with (LiM3D+SDSC), corresponding to the bottom two rows of Tab. 2.

Effectiveness of SDSC module. In Tab. 3, we compare our LiM3D and LiM3D+SDSC with recent state-of-the-art methods under 5%-labeled semi-supervised training on the SemanticKITTI and ScribbleKITTI *validation* sets. LiM3D+SDSC outperforms the voxel-based methods [46, 63] with at least a **2.3**× reduction in model size. Similarly, with comparable model size [13, 44, 58], LiM3D+SDSC has higher mIoU in both datasets and up to **641**× fewer multiply-add operations.

Effectiveness of ST-RFD strategy. In Tab. 4, we illustrate the effectiveness of our ST-RFD strategy by comparing LiM3D with two widely-used strategies in semi-supervised training, *i.e.*, random sampling and uniform sampling on SemanticKITTI [7] and ScribbleKITTI [46] *validation* set. Whilst uniform and random sampling have comparable results on both *validation* sets, simply applying our ST-RFD strategy improves the baseline by $+0.90\%$, $+0.75\%$, $+0.60\%$ and $+0.55\%$ on SemanticKITTI under 5%, 10%, 20% and 40% sampling protocol respectively. Furthermore, using corresponding range images of point cloud, rather than RGB images to compute the spatio-temporal redundancy within ST-RFD (see ST-RFD-R in Tab. 4), has no significant difference on the performance.

Effectiveness of Unreliable Pseudo-Labeling. In Tab. 5, we evaluate selecting negative candidates with different reliability to illustrate the improvements of using unreliable pseudo-labels in semi-supervised semantic segmentation. The *"Unreliable"* selecting of negative candidates outperforms other alternative methodologies, showing the positive performance impact of unreliable pseudo-labels.

Effectiveness of Reflec-TTA. In Tab. 2, we compare LiM3D performance with and without Reflec-TTA and further experiment on the SemanticKITTI and ScribbleKITTI *validation* set in Tab. 6. This demonstrates that the LiDAR point-wise intensity feature $I^{\circledast}$, in place of the distance-normalized reflectivity feature $R^{\circledast}$, offers inferior on-task performance.

5. Conclusion

This paper presents an efficient semi-supervised architecture for 3D point cloud semantic segmentation, which achieves *more* in terms of performance with *less* computational costs, *less* annotations, and *less* trainable model parameters (*i.e.*, *Less is More*, LiM3D). Our architecture consists of three novel contributions: the SDSC convolution module, the ST-RFD sampling strategy, and the pseudo-labeling method informed by LiDAR reflectivity. These individual components can be applied to any 3D semantic segmentation architecture to reduce the gap between semi or weakly-supervised and fully-supervised learning on task performance, whilst managing model complexity and computation costs.

References

[1] FairScale . FairScale: A general purpose modular PyTorch library for high performance and large scale training, 2021. 7

[2] Shahriar Akramullah. *Digital Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis*. Apress, Berkeley, CA, 2014. 2

[3] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C. Murillo. 3D-MiniNet: Learning a 2D Representation From Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation. *IEEE Robot. Autom. Lett.*, 5(4):5432–5439, Oct. 2020. 1

[4] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C. Murillo. Semi-Supervised Semantic Segmentation with Pixel-Level Contrastive Learning from a Class-wise Memory Bank. In *Int. Conf. Comput. Vis.*, pages 8199–8208, 2021. 2

[5] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. In *Int. Jt. Conf. Neural Netw.*, 2020. 2, 4

[6] Amir Atapour-Abarghouei and Toby P. Breckon. Veritatem Dies Aperit - Temporally Consistent Depth Prediction Enabled by a Multi-Task Geometric and Semantic Scene Understanding Approach. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3368–3379, 2019. 1

[7] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Int. Conf. Comput. Vis.*, pages 9296–9306, Seoul, Korea (South), Oct. 2019. 2, 4, 5, 6, 7, 8

[8] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovasz-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[9] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11679–11689, 2020. 5

[10] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11618–11628, 2020. 4, 5

[11] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2613–2622, June 2021. 1, 5

[12] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, Apr. 2017. 2, 6

[13] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3070–3079, 2019. 1, 8

[14] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds. In *Adv. Vis. Comput.*, pages 207–222, 2020. 1

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009. 2

[16] Ngoc Phuong Anh Duong, Alexandre Almin, Léo Lemarié, and B Ravi Kiran. Reducing redundancy in Semantic-KITTI: Study on data augmentations within Active Learning. In *Adv. Neural Inform. Process. Syst. Worksh.*, page 8, 2017. 3

[17] Geoff French. Semi-supervised semantic segmentation needs strong, varied perturbations. In *Brit. Mach. Vis. Conf.*, page 21, 2020. 5

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3354–3361, 2012. 4, 5

[19] Ben Graham. Sparse 3D convolutional neural networks. In *Brit. Mach. Vis. Conf.*, pages 150.1–150.9, Swansea, 2015. 6

[20] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 6

[21] Benjamin Graham and Laurens van der Maaten. Submanifold Sparse Convolutional Networks, June 2017. 2

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9726–9735, 2020. 4

[23] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-Voxel Knowledge Distillation for LiDAR Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, page 10, 2022. 1, 3, 6

[24] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for MobileNetV3. In *Int. Conf. Comput. Vis.*, pages 1314–1324, 2019. 2, 6

[25] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Apr. 2017. 2, 6

[26] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 6

[27] Zeyu Hu, Xuyang Bai, Runze Zhang, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. LiDAL: Inter-frame Uncertainty Based Active Learning for 3D LiDAR Semantic Segmentation. In *Eur. Conf. Comput. Vis.* Springer. 3

[28] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Émilie Wirbel, and Patrick Pérez. xMUDA: Cross-Modal Unsupervised Domain Adaptation for 3D Semantic Segmentation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 6

[29] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided Point Contrastive Learning for Semi-supervised Point Cloud Semantic Segmentation. In *Int. Conf. Comput. Vis.*, 2021. 2, 3, 6, 7

[30] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 perception dataset 2020, 2019. 4, 5

[31] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. KPRNet: Improving projection-based LiDAR semantic segmentation, Aug. 2020. 1, 2

[32] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. LaserMix for Semi-Supervised LiDAR Semantic Segmentation, June 2022. 2, 3, 5, 6, 7

[33] Samuli Laine and Timo Aila. Temporal Ensembling for Semi-Supervised Learning. In *Int. Conf. Learn. Represent.*, 2017. 3

[34] Li Li, Khalid N. Ismail, Hubert P. H. Shum, and Toby P. Breckon. DurLAR: A High-Fidelity 128-Channel LiDAR Dataset with Panoramic Ambient and Reflectivity Imagery for Multi-Modal Autonomous Driving Applications. In *Int. Conf. 3D Vis.*, pages 1227–1237, London, United Kingdom, Dec. 2021. 4, 5

[35] Peizhao Li, Pu Wang, Karl Berntorp, and Hongfu Liu. Exploiting Temporal Relations on Radar Perception for Autonomous Driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17071–17080, 2022. 3

[36] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. AMVNet: Assertion-based Multi-View Fusion Network for LiDAR Semantic Segmentation, Dec. 2020. 2

[37] Dominic Masters, Antoine Labatie, Zach Eaton-Rosen, and Carlo Luschi. Making EfficientNet More Efficient: Exploring Batch-Independent Normalization, Group Convolutions and Reduced Resolution Training, Aug. 2021. 2, 6

[38] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In *Int. Conf. Intell. Robots Syst.*, pages 4213–4220, 2019. 1, 2

[39] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In *Adv. Neural Inform. Process. Syst.*, volume 29, 2016. 3

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4510–4520, Salt Lake City, UT, 2018. IEEE. 2, 6

[41] Laurent Sifre. *Rigid-Motion Scattering For Image Classification*. PhD thesis, Ecole Polytechnique, 2014. 2

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2443–2451, Seattle, WA, USA, June 2020. 5

[43] Mingxing Tan and Quoc Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proc. 36th Int. Conf. Mach. Learn.*, pages 6105–6114. PMLR, May 2019. 6

[44] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2, 8

[45] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Adv. Neural Inform. Process. Syst.*, 2017. 2, 3, 5

[46] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised LiDAR semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[47] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, Jan. 2019. 1, 3, 4

[48] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-Short Temporal Contrastive Learning of Video Transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13990–14000, 2022. 3

[49] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-Supervised Semantic Segmentation Using Unreliable Pseudo-Labels. *IEEE Conf. Comput. Vis. Pattern Recog.*, Mar. 2022. 1, 4

[50] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, Apr. 2004. 5

[51] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *Int. Conf. Robot. Autom.*, pages 1887–1893, 2018. 1, 2

[52] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In *Int. Conf. Robot. Autom.*, pages 4376–4382, 2019. 1, 2

[53] Tsung-Han Wu, Yueh-Cheng Liu, Yu-Kai Huang, Hsin-Ying Lee, Hung-Ting Su, Ping-Chia Huang, and Winston H. Hsu. ReDAL: Region-based and Diversity-aware Active Learning for Point Cloud Semantic Segmentation. In *Int. Conf. Comput. Vis.* 3

[54] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3733–3742, 2018. 4

[55] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-Training With Noisy Student Improves ImageNet Classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10687–10698, 2020. 3

[56] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3:

Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[57] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. In *Conf. AAAI Artif. Intell.*, 2021. 2

[58] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2DPASS: 2D priors assisted semantic segmentation on LiDAR point clouds. In *Eur. Conf. Comput. Vis.*, 2022. 1, 8

[59] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. ST++: Make Self-Training Work Better for Semi-Supervised Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4268–4277, 2022. 3

[60] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & Label: A Domain Adaptation Approach to Semantic Segmentation of LiDAR Point Clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15358–15368, Nashville, TN, USA, June 2021. 1, 6

[61] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[62] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep Feature Flow for Video Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4141–4150, Honolulu, HI, 2017. 3

[63] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 2, 3, 6, 7, 8

[64] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-training. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Eur. Conf. Comput. Vis.*, volume 11207, pages 297–313, 2018. 5

[65] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. PseudoSeg: Designing Pseudo Labels for Semantic Segmentation. In *Int. Conf. Learn. Represent.*, 2021. 3

Less is More: Reducing Task and Model Complexity for 3D Point Cloud Semantic Segmentation - Supplementary

Li Li[1] Hubert P. H. Shum[1] Toby P. Breckon[1,2]
Department of {Computer Science[1] | Engineering[2]}, Durham University, UK
{li.li4, hubert.shum, toby.breckon}@durham.ac.uk

A. Performance with Different Backbones

In Tab. A1, alongside Cylinder3D [8], we also implement our architecture with popular backbone networks [2, 5] widely-used in 3D semantic segmentation.

B. Runtime Comparison

We conduct a runtime comparison between LiM3D and SOTAs on NVIDIA A100 GPU and Graphcore IPU-POD16 for Intelligence Processing Unit (IPU) acceleration. Summary inference times per item: LiM3D (ours, GPU) - 0.51s, Cylinder3D [8] (GPU) - 0.48s, Unal *et al.* [6] (GPU) - 0.53s, LiM3D+SDSC (ours, IPU) - 0.12s. SDSC could be slower on GPU due to its lower arithmetic intensity (ratio of compute to memory operation) [3, 7], but the high-bandwidth on-chip memory in IPU accelerator significantly improves the efficiency.

C. More Quantitative Results on Semi-supervised Segmentation

Besides {5%, 10%, 20%, 40%} labeled frames training, we also report our results with less than 5% label frames shown in Tab. A2. By applying our proposed architecture for semi-supervised and scribble-supervised 3D semantic segmentation, LiM3D and LiM3D+SDSC achieve higher than 80% relative performances (SS/FS) comparing with fully-supervised methods with less than only 1% labeled, *i.e.*, 191 frames (Tab. A2).

D. More Qualitative Results

Figs. A1 and A2 show a higher-resolution version of qualitative results that our method has superior performance. Fig. A3 compares {5%, 10%, 20%, 40%} sampling splits of SemanticKITTI [1] using LiM3D (ours). Note that using our semi-supervised methodology, the results training with very few ground-truth labels (*e.g.*, 5% and 10%) can achieve comparable performance to the training with a large number of labels (see Fig. A3, the 1-st and 2-nd rows *v.s.* 4-th row), with only subtle differences shown in the green and red circle. In Fig. A4, the magnification of regional details shows that our method can achieve better segmentation results than other methods, especially in the category of `vegetation`, `fence`, `sidewalk`, *etc*.

E. Parameters and Computation Costs Analysis on SDSC Sub-module

Given a Tensor $F \in \mathbb{R}^{H_F \times W_F \times L_F \times M}$, where H_F, W_F, L_F and M denote the radius, azimuth, height in the cylinder coordinate [8] and channels respectively. Applying convolution operation only for the active site of the sparse 3D point cloud, the computational cost (in FLOPs) of submanifold sparse convolution (SSC, [4]) is $a \times M \times N$ for the active site, where M is the number of input channels as defined previously, and N is the number of output channels. a is the number of active inputs to the spatial location defined in [4]. The computational cost for the inactive site is 0.

Since our SDSC sub-module consists of a sparse depthwise convolution (SDC) and a sparse pointwise convolution (SPC), the computational cost for SDSC is the sum cost of those two parts. SDC has a computational cost of

$$a \times M \times H_F \times W_F \times L_F. \tag{A1}$$

SPC computes a linear combination of the SDC output via a 1×1 convolution, which has the computational cost of

$$M \times N \times H_F \times W_F \times L_F. \tag{A2}$$

As a result, the computational cost of SDSC is the sum of Eqs. (A1) and (A2), *i.e.*,

$$a \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F. \tag{A3}$$

The ratio of computational cost of SDSC to SSC [4] for active site, *i.e.*, cost$(SDSC)$: cost(SSC), is:

$$\begin{aligned}&\frac{a \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F}{a \times M \times N \times H_F \times W_F \times L_F}\\ =&\frac{1}{N} + \frac{1}{a} \approx \frac{1}{N} \quad (a \gg N).\end{aligned} \tag{A4}$$

Similar to the computational cost analysis, the parameters of SDSC is also the sum of SDC and SPC. The ratio of model

Table A1. 3D semantic segmentation results of LiM3D (ours) evaluated on SemanticKITTI [1] and ScribbleKITTI [6] *valid*-set with 10% labeled data, using different backbones. Alongside the per-class metrics, we show the relative performance of the semi-supervised approach against the fully supervised (SS/FS). S: with SDSC sub-module (✓) or without SDSC sub-module, *i.e.*, with normal sparse convolution.

Model	Dataset	S	mIoU	SS/FF	car	bicycle	motorcycle	truck	other vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
LiM3D (ours) + Cylinder3D [8]	Semantic		**62.2**	**89.5**	95.5	47.6	**65.2**	60.7	42.4	**75.2**	84.3	0.0	**94.2**	42.4	**80.7**	**5.4**	91.0	61.1	86.6	65.7	70.7	64.0	49.2
	Semantic	✓	61.0	87.8	95.3	43.9	59.2	46.2	47.0	71.4	79.8	1.6	93.8	44.0	80.0	4.4	90.8	60.4	87.7	64.5	73.5	63.8	51.4
	Scribble		61.0	87.8	95.0	34.5	52.9	61.5	41.3	71.0	**85.6**	0.0	93.7	44.4	79.9	0.4	90.1	58.1	87.9	61.5	74.8	65.3	44.3
	Scribble	✓	56.7	81.6	**95.6**	**48.9**	45.2	16.0	43.1	66.9	81.8	0.0	91.8	30.9	75.7	1.8	90.0	59.2	86.6	62.6	69.8	63.4	46.8
LiM3D (ours) + MinkowskiNet [2]	Semantic		60.4	86.9	94.6	44.3	47.1	70.2	29.5	68.7	80.8	0.0	93.6	38.4	79.6	0.1	90.2	58.7	88.2	66.1	75.6	**65.5**	**58.9**
	Semantic	✓	59.4	85.5	94.5	43.3	47.1	70.2	29.5	66.7	76.0	0.0	93.5	38.1	79.2	0.1	90.1	58.4	87.8	65.1	74.3	64.1	50.1
	Scribble		56.2	80.9	93.8	42.7	37.6	68.4	33.7	54.2	63.5	0.0	91.9	39.9	76.7	0.1	88.9	61.9	86.9	67.3	74.3	59.1	27.7
	Scribble	✓	50.3	72.4	92.8	08.8	30.3	68.2	36.3	27.2	56.6	**1.7**	88.0	47.9	71.8	2.7	85.3	63.9	86.7	71.2	74.7	56.6	13.6
LiM3D (ours) + SPVCNN [5]	Semantic		62.1	89.4	94.6	48.3	59.1	72.2	45.9	62.2	68.1	0.0	91.7	**59.2**	79.0	1.7	**91.6**	65.0	86.6	70.9	71.4	64.2	42.8
	Semantic	✓	60.8	87.5	92.7	29.6	64.8	72.9	**51.2**	54.0	47.1	0.0	89.0	58.2	75.2	1.9	88.8	**68.6**	**88.9**	**73.4**	**77.8**	64.4	30.8
	Scribble		59.6	85.8	92.5	29.4	63.7	72.8	50.8	52.2	43.2	0.0	89.8	56.3	75.1	0.6	88.6	66.7	87.6	71.0	74.3	62.3	29.6
	Scribble	✓	54.7	78.7	91.0	23.1	61.8	**73.1**	45.5	30.0	36.3	1.6	87.5	56.2	72.7	0.7	86.7	68.3	87.7	72.8	75.8	61.3	19.3

Table A2. 3D semantic segmentation results of LiM3D (ours) evaluated on SemanticKITTI [1] and ScribbleKITTI [6] *valid*-set with %1 and 2% labeled data. Alongside the per-class metrics, we show the relative performance of the semi-supervised approach against the fully supervised (SS/FS). S: with SDSC sub-module (✓) or without SDSC sub-module, *i.e.*, with normal sparse convolution.

% labeled	Dataset	S	mIoU	SS/FF	car	bicycle	motorcycle	truck	other vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
LiM3D (ours) 2% / 383 frames	Semantic		**59.3**	**85.3**	**95.6**	**37.6**	50.1	54.4	46.0	**68.8**	77.1	0.0	**87.9**	32.8	**76.6**	2.0	**91.4**	**54.8**	**89.5**	**69.5**	**77.1**	**66.2**	**49.4**
	Semantic	✓	58.7	84.5	94.8	37.1	**55.0**	56.2	45.2	66.1	75.4	0.0	87.0	32.5	75.9	2.1	89.1	49.7	89.3	68.0	76.0	65.8	45.9
	Scribble		58.2	83.7	92.5	35.6	52.0	**57.1**	**49.4**	66.8	78.5	0.0	85.6	30.2	74.4	2.4	89.7	54.0	88.2	66.9	74.4	63.7	47.7
	Scribble	✓	56.8	81.7	93.1	34.9	47.0	50.9	43.8	64.1	75.6	0.0	85.2	29.5	73.9	2.0	88.8	49.5	88.2	66.9	74.8	64.4	47.4
LiM3D (ours) 1% / 191 frames	Semantic		58.4	84.0	92.6	37.5	51.2	50.4	47.9	68.6	**80.3**	0.0	86.3	**33.5**	74.7	**3.9**	89.4	51.4	88.3	67.4	75.1	64.8	45.8
	Semantic	✓	57.2	82.3	92.6	34.5	47.2	54.5	44.3	65.5	76.6	0.0	85.5	29.2	74.3	2.5	88.9	49.7	88.1	67.1	74.9	63.9	47.0
	Scribble		57.0	82.0	93.1	31.7	46.8	55.4	45.2	65.2	71.8	0.0	85.3	29.8	74.0	2.7	89.1	50.9	88.3	67.8	75.4	64.2	45.9
	Scribble	✓	55.8	80.3	92.7	27.6	43.6	50.6	42.3	60.6	73.9	0.0	85.3	29.1	74.2	2.6	87.3	49.7	87.2	68.5	70.2	64.5	43.6

parameters of SDSC to SSC [4] is:

$$\begin{aligned}&\frac{\mathbf{D_K} \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F}{\mathbf{D_K} \times M \times N \times H_F \times W_F \times L_F}\\ =&\frac{1}{N} + \frac{1}{\mathbf{D_K}} \approxeq \frac{1}{\mathbf{D_K}} \quad (N \gg \mathbf{D_K}),\end{aligned} \tag{A5}$$

where $\mathbf{D_K}$ is the dimension of convolution kernel K of size $D_{K,1} \times D_{K,2} \times D_{K,3}$, *i.e.*, $\mathbf{D_K} = D_{K,1} \times D_{K,2} \times D_{K,3}$.

LiM3D+SDSC uses SDSC sub-module as the basic building block for constructing other convolution-based modules in Cylinder3D (*e.g.*, residual block, upsample block, and downsample block). Take the residual block as an example, SDSC uses approximately 32x, 64x, $\cdots$, 512x less computation than SSC for active sites when $N = \{32, 64, \cdots, 512\}$ (Eq. (A4)). SDSC-based residual block with a kernel size of $1 \times 3 \times 1$ has 3x fewer parameters than the SSC-based residual block with the same kernel size, and 9x fewer parameters with $3 \times 1 \times 3$ kernel size (Eq. (A5)).

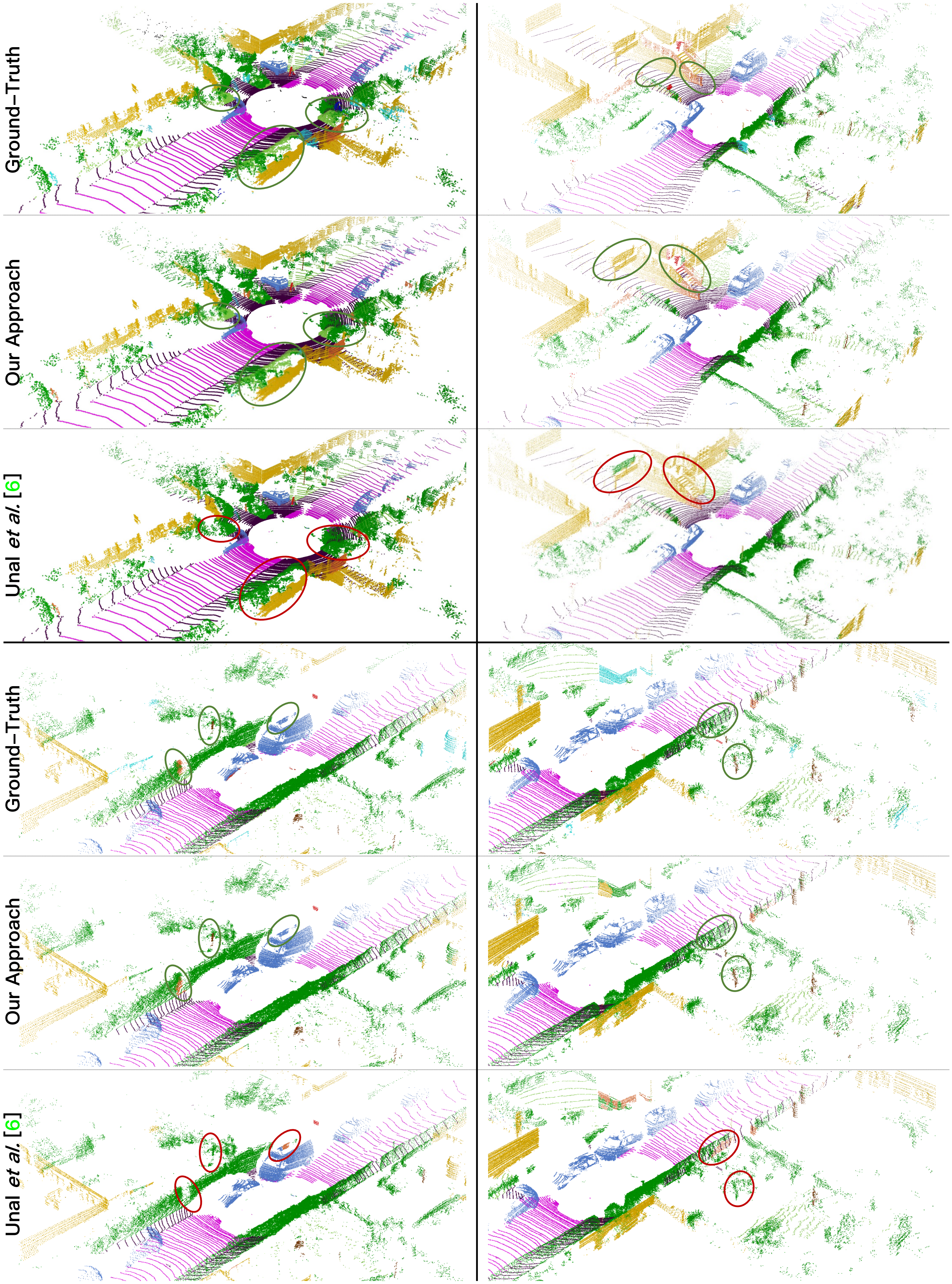

Figure A1. Comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

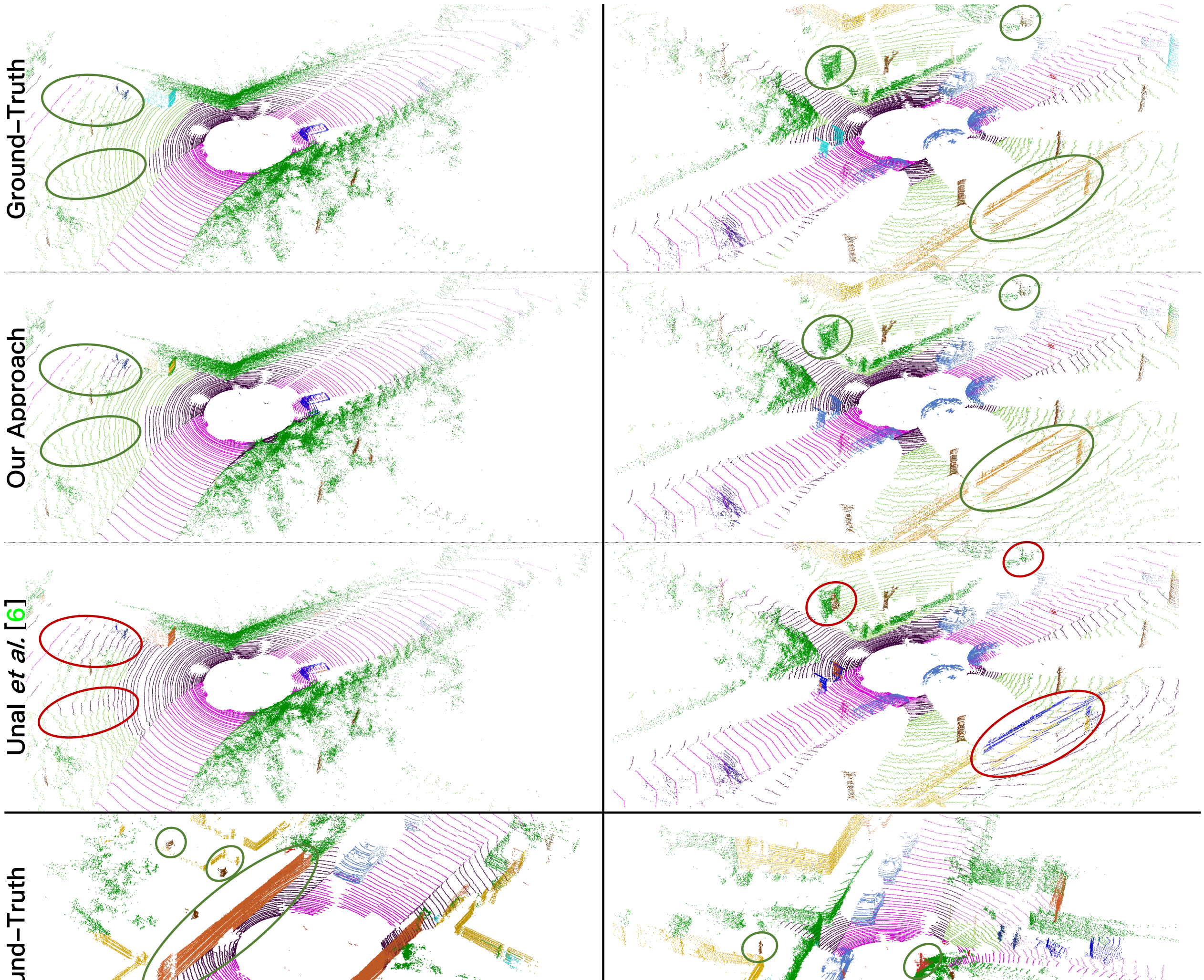

Figure A2. Comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

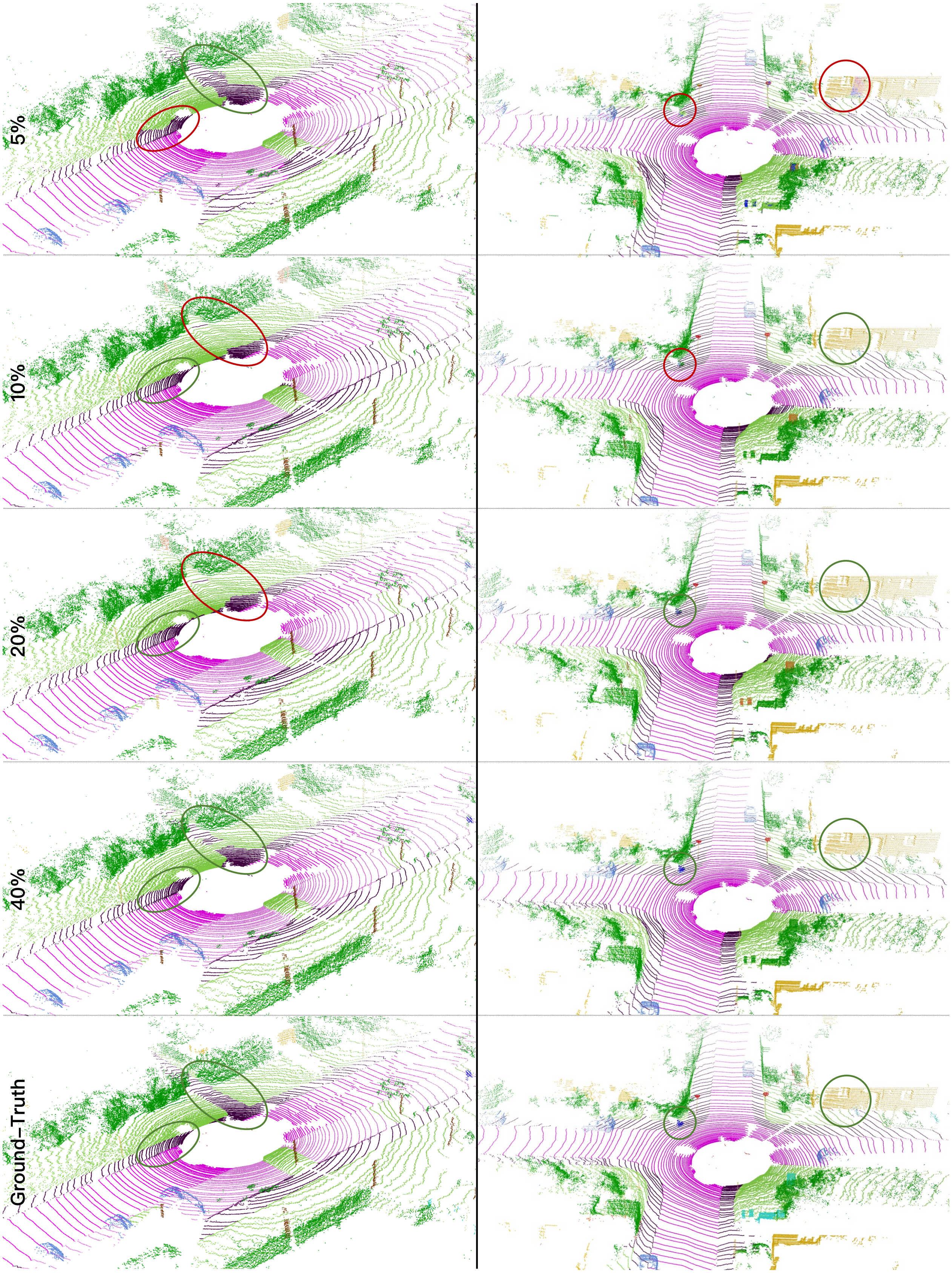

Figure A3. Comparing the 5%, 10%, 20%, 40% sampling split of SemanticKITTI [1] *validation* set with ground-truth (bottom) with areas of improvement highlighted in green, and areas of underperformance in red.

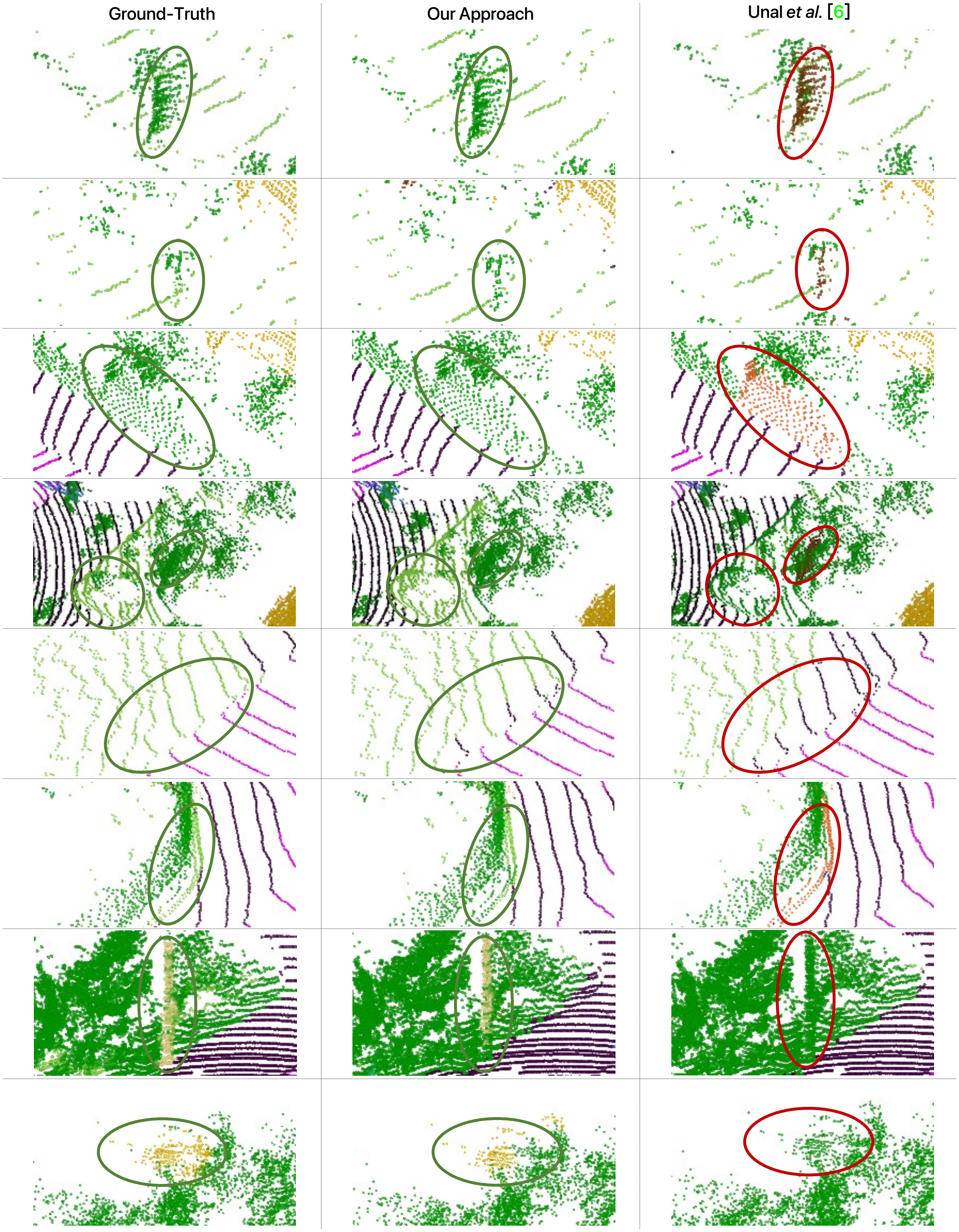

Figure A4. Magnification of regional details: comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Int. Conf. Comput. Vis.*, pages 9296–9306, Seoul, Korea (South), Oct. 2019. 1, 2, 3, 4, 5, 6

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3070–3079, 2019. 1, 2

[3] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer. Squeezenext: Hardware-aware neural network design. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2018. 1

[4] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 2

[5] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[6] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised LiDAR semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3, 4, 6

[7] S. Williams, A. Waterman, and D. A. Patterson. Roofline: an insightful visual performance model for multicore architectures. *Communications of the Acm*, 52(4):65–76, 2009. 1

[8] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 2