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

Li Li\textsuperscript{1} Hubert P. H. Shum\textsuperscript{1} Toby P. Breckon\textsuperscript{1,2} \\
Department of \{Computer Science\textsuperscript{1} | Engineering\textsuperscript{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.

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/.
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 subsamples 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 \( P = \{p \mid p = (x, y, z, I, R) \in \mathbb{R}^5\} \) 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 \( U = \{p^u_i\}_{i=1}^{N_u} \not\subseteq P \) and a smaller set of labeled data \( V = \{(p^l_i, y^l_i)\}_{i=1}^{N_l} \subseteq P \).

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 \( \theta \) are updated via standard backpropagation, while the weights of the teacher model \( \theta^* \) are updated by the student model through Exponential Moving Averaging (EMA):

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

where \( \kappa \) 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:

\[
\mathcal{L} = \mathcal{L}_S + \lambda_U \mathcal{L}_U + g \lambda_C \mathcal{L}_C,
\]

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, \( \lambda_U \) is the weight coefficient of \( \mathcal{L}_U \) to balance the losses, and \( g \) is the gated coefficient of \( \mathcal{L}_C \); \( g \) equals \( \lambda_C \) 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].
Thus, together with the use of CRB for high-quality reliable voxels with high entropy, can offer information that helps if the ground truth is available, otherwise, at frame at voxel $c$, according to Eq. (3) and Eq. (4), in order to reduce dominance of large and simple-to-learn classes within our distribution of many tasks. 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:

$$R = \frac{S}{4\pi r^2}, \quad r^2 \propto S,$$

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 (Reflect-TTA) technique, as a substitute for label-dependent Pyramid Local Semantic-context
(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, \( h_i \):

\[
\begin{align*}
    h_i & = \left\{ h_i^{(k)} \mid k \in [1, N_b] \right\} \subseteq \mathbb{R}^{N_b}, \quad i \in [1, s], \\
    h_i^{(k)} & = \left\{ R_j \in r_k, \forall j \mid p_j \in b_i \right\}, \\
    r_k & = \left\{ (k-1)/N_b, k/N_b \right\}, \quad k \in [1, N_b].
\end{align*}
\]  

(7)

The Reflec-TTA features \( R^\circ \) of all points \( p_j \in b_j \) is further computed as the concatenation of the coarse histogram \( h_i \) of the normalized histogram:

\[
R^\circ = \{ h_i / \max(h_i) \mid i \in [1, s] \} \subseteq \mathbb{R}^{sN_b}
\]  

(8)

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

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.

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 \([p/q]\) 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 \( \upsilon \), 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 \( \psi \) to measure the similarity between the RGB images associated with two adjacent point clouds, we define the empirical supervisor function \( \upsilon \) with decay property \( \upsilon(x) = \exp(-\beta x) \), where \( \beta \in (0, +\infty) \) is the decay coefficient, and \( x \) is the redundancy calculated from \( \psi \). 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
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 $\nu$.

1. Divide $P$ evenly into \( \lceil p/q \rceil \) subsets $Q$.
2. $D \leftarrow$ empty dictionary.
3. forall $e \leftarrow 0 : n_e - 1$ do
   // loop for all sequences
4.   $C_e \leftarrow \emptyset$ // chosen point cloud frames
5.   forall $i \leftarrow 0 : \lceil p/q \rceil - 1$ do
6.     $Q_{i,j} \leftarrow j$-th frame in subset $Q_i$.
7.     $M_i \leftarrow \sum_{j=1}^{\lceil p/q \rceil} \psi(Q_{i,j}).$ // redundancy
8.     $k_i \leftarrow \left\lfloor \nu(M_i) \cdot q \right\rfloor$.
9.     $T_i \leftarrow$ select $k_i$ frames in $Q_i$ with the smallest $M_i$.
10. $C_e \leftarrow C_e \cup T_i$.
11. Append key-value pair $(e, C_e)$ into $D$.

**Return:** Dictionary $D$.

tens of millions of trainable parameters [23,26,28,60,63] 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.

<table>
<thead>
<tr>
<th>Repr.</th>
<th>Samp.</th>
<th>Method</th>
<th>Training mIoU (%)</th>
<th>Validation mIoU (%)</th>
<th>#Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1% 5% 10% 20% 40% 50% 100%</td>
<td>1% 5% 10% 20% 40% 50% 100%</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>U</td>
<td>LaserMix [32] (2022)</td>
<td>43.4 58.8 59.4 – 61.4</td>
<td>38.3 54.4 55.6 – 58.7</td>
<td></td>
</tr>
<tr>
<td>Voxel</td>
<td>U</td>
<td>Cylinder3D [63] (CVPR’21)</td>
<td>– 45.4 56.1 57.8 58.7 – 67.8</td>
<td>– 39.2 48.0 52.1 53.8 – 56.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>LaserMix [32] (2022)</td>
<td>50.6 – 60.0 61.9</td>
<td>44.2 – 53.7 55.1 – 56.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>Jiang et al. [50] (ICCV’21)</td>
<td>– 41.8 49.9 58.8 59.9 – 65.8</td>
<td>– 46.9* 54.2* 56.5* 58.6* – 61.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>Unal et al. [46] (CVPR’22)</td>
<td>– 49.9* 58.7* 59.1* 60.9</td>
<td>– 46.9* 54.2* 56.5* 58.6*</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>LiM3D+SDSC (ours)</td>
<td>57.2 57.6 61.0 61.7</td>
<td>55.8 56.1 56.9 57.2</td>
<td>58.9 59.3 60.7</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>LiM3D (ours)</td>
<td>58.4 59.5 62.2 63.1</td>
<td>57.0 58.1 61.0</td>
<td>61.2 62.0 62.1</td>
</tr>
</tbody>
</table>

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 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: \( N_i = 10, s = 3 \) various Reflec-TTA bin sizes, following [46], we set each bin \( b_i = (\rho, \phi) \in \{(20, 40), (40, 80), (80, 120)\} \).

![Ground-Truth](image1.png) ![Our Approach](image2.png) ![Unal et al.](image3.png)

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.

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.

<table>
<thead>
<tr>
<th>UP</th>
<th>RF</th>
<th>RT</th>
<th>ST</th>
<th>SD</th>
<th>Training mIoU (%)</th>
<th>Validation mIoU (%)</th>
<th>#Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>82.8 87.5 87.8 88.2</td>
<td>54.8 58.1 59.3 60.8</td>
<td>49.6</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>83.6 88.3 88.7 89.1</td>
<td>56.8 59.6 60.5 61.4</td>
<td>49.6</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>83.7 88.6 88.5 89.2</td>
<td>57.5 59.8 61.2 62.6</td>
<td>49.6</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>85.2 89.1 89.5 89.7</td>
<td>59.5 62.2 63.1 63.3</td>
<td>49.6</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>83.8 88.6 89.0 89.2</td>
<td>57.6 61.0 61.7 62.1</td>
<td>21.5</td>
</tr>
</tbody>
</table>

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.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th># Parameters</th>
<th># Mult-Adds</th>
<th>SeK [7]</th>
<th>ScK [46]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylinder3D [63]</td>
<td>56.3</td>
<td>476.9M</td>
<td>45.4</td>
<td>39.2</td>
</tr>
<tr>
<td>Unal et al. [46]</td>
<td>49.6</td>
<td>420.2M</td>
<td>49.9</td>
<td>46.9</td>
</tr>
<tr>
<td>2D PASS [58]</td>
<td>26.5</td>
<td>217.4M</td>
<td>51.7</td>
<td>45.1</td>
</tr>
<tr>
<td>MinkowskiNet [13]</td>
<td>21.7</td>
<td>114.0G</td>
<td>42.4</td>
<td>35.8</td>
</tr>
<tr>
<td>SPVNAS [44]</td>
<td>12.5</td>
<td>73.8G</td>
<td>45.1</td>
<td>38.9</td>
</tr>
<tr>
<td>LiM3D+SDSC (ours)</td>
<td>21.5</td>
<td>182.0m</td>
<td>57.6</td>
<td>54.7</td>
</tr>
<tr>
<td>LiM3D (ours)</td>
<td>49.6</td>
<td>420.2M</td>
<td>59.5</td>
<td>58.1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>10%</td>
</tr>
<tr>
<td>Random</td>
<td>58.5</td>
<td>61.6</td>
</tr>
<tr>
<td>Uniform</td>
<td>58.7</td>
<td>61.3</td>
</tr>
<tr>
<td>ST-RFD-R</td>
<td>59.1</td>
<td>62.4</td>
</tr>
<tr>
<td>ST-RFD</td>
<td>59.5</td>
<td>62.2</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Unreliable mIoU</th>
<th>Reliable mIoU</th>
<th>Random mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SS/FF</td>
<td>SS/FF</td>
<td>SS/FF</td>
</tr>
<tr>
<td>5%</td>
<td>59.5</td>
<td>85.6</td>
<td>57.2</td>
</tr>
<tr>
<td>10%</td>
<td>62.2</td>
<td>89.5</td>
<td>60.8</td>
</tr>
<tr>
<td>20%</td>
<td>63.1</td>
<td>90.8</td>
<td>61.4</td>
</tr>
<tr>
<td>40%</td>
<td>63.3</td>
<td>91.1</td>
<td>62.8</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>10%</td>
</tr>
<tr>
<td></td>
<td>Intensity</td>
<td>Reflectivity</td>
</tr>
<tr>
<td></td>
<td>mIoU</td>
<td>mIoU</td>
</tr>
<tr>
<td>5%</td>
<td>56.2</td>
<td>59.1</td>
</tr>
<tr>
<td>10%</td>
<td>59.5</td>
<td>62.2</td>
</tr>
</tbody>
</table>

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 Iσ, in place of the distance-normalized reflectivity feature Rσ, 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

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


[32] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. LaserSeg: Reduced Resolution Training, Aug. 2021. 2, 6


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


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


