
Annotator: A Generic Active Learning Baseline for LiDAR Semantic Segmentation

Binhui Xie

Beijing Institute of Technology
binhuixie@bit.edu.cn

Shuang Li[✉]

Beijing Institute of Technology
shuangli@bit.edu.cn

Qingju Guo

Beijing Institute of Technology
qingjuguo@bit.edu.cn

Chi Harold Liu

Beijing Institute of Technology
chiliu@bit.edu.cn

Xinjing Cheng

Tsinghua University & Inceptio Technology
cnorbot@gmail.com

Abstract

Active learning, a label-efficient paradigm, empowers models to interactively query an oracle for labeling new data. In the realm of LiDAR semantic segmentation, the challenges stem from the sheer volume of point clouds, rendering annotation labor-intensive and cost-prohibitive. This paper presents *Annotator*, a general and efficient active learning baseline, in which a voxel-centric online selection strategy is tailored to efficiently probe and annotate the salient and exemplar voxel girds within each LiDAR scan, even under distribution shift. Concretely, we first execute an in-depth analysis of several common selection strategies such as Random, Entropy, Margin, and then develop voxel confusion degree (VCD) to exploit the local topology relations and structures of point clouds. *Annotator* excels in diverse settings, with a particular focus on active learning (AL), active source-free domain adaptation (ASFDA), and active domain adaptation (ADA). It consistently delivers exceptional performance across LiDAR semantic segmentation benchmarks, spanning both simulation-to-real and real-to-real scenarios. Surprisingly, *Annotator* exhibits remarkable efficiency, requiring significantly fewer annotations, e.g., just labeling five voxels per scan in the SynLiDAR → SemanticKITTI task. This results in impressive performance, achieving 87.8% fully-supervised performance under AL, 88.5% under ASFDA, and 94.4% under ADA. We envision that *Annotator* will offer a simple, general, and efficient solution for label-efficient 3D applications.

1 Introduction

3D perception and understanding have become indispensable for machines to effectively interact with the real world. LiDAR (Light Detection And Ranging) [56, 58] is a widely-used methodology for capturing precise geometric information about the environment, spurring significant advancements in areas like autonomous vehicles and robotics [18, 29]. However, semantic segmentation of LiDAR presents an enormous challenge. The high-speed collection of millions of point clouds per second by on-board sensors sharply contrasts with the laborious and cost-prohibitive nature of annotating them. Consider, for instance, the vast number of outdoor scenes an autopilot can encounter, which is practically limitless. Yet, acquiring annotations for these large-scale point clouds entails intensive human labor. This underscores the urgency of establishing a label-efficient learning mechanism capable of boosting performance in the low-data regime [22, 82, 97, 104] or facilitating the adaptation of models to new domains [61, 70, 80, 92].

[✉] Corresponding author. Project page: <https://binhuixie.github.io/annotator-web/>

Extensive solutions encompass semi-supervised [9, 12, 33, 36, 90], weakly-supervised [25, 41, 82, 106] or self-supervised [6, 16, 69, 79, 108] learning. Semi- and weakly-supervised learning methods aim to alleviate the annotation burden by harnessing partially labeled or weakly labeled data. In contrast, self-supervised ones learn representations from point clouds via pretext tasks and then transfer to downstream tasks for weight initialization. Although these works offer scalability and practicality for real-world utility, they also confront new challenges, such as variations in LiDAR configurations, sensor biases, and environmental conditions. That is, the majority of prior works has endeavored to in-distribution scenarios, with limited consideration for label-efficient paradigms in out-of-distribution scenarios, especially for sparse outdoor point clouds. Recent efforts turn to large-scale auxiliary datasets [62, 91] and delve into domain adaptation (DA) algorithms [1, 32, 57, 99] to significantly reduce the annotation workload under a domain shift. Nevertheless, the performance of these methods still lags behind the fully-supervised approaches. In Figure 1, we provide an intuitive comparison of results across various paradigms. It becomes evident that there is ample room for improvement in the performance of these methods.

To surmount these obstacles and promote performance in the domain of interest, active learning (AL) is being an optimal paradigm [31, 40, 66, 89]. Given the limited annotation budget, a common scenario is that only an unlabeled target domain of large amounts of point clouds is available with the goal to interactively select a minimal subset of data to be annotated to maximally improve the segmentation performance. In reality, this setting faces a significant hurdle known as *cold start problem*: the lack of prior information to guide the initial selection of annotated data. A recent work has explored the impact of seeding strategies on the performance of AL methods [64]. Differently, we put forward a new path to access an auxiliary model via pre-training on the open-access auxiliary (source) dataset. This auxiliary model serves as a warm-up stage, allowing for smart target data selection for initial annotation. We formulate this new setting as active source-free domain adaptation, termed ASFDA. Take a further step, drawing inspiration from recent trends in 2D images [43, 50, 94, 95], we delve into the third setting, active domain adaptation (ADA) for semantic segmentation of 3D point clouds. In this setting, a labeled auxiliary dataset is available, and the objective is to select target instances for annotating and learn a model with higher segmentation performance on the target test set.

Overall, in this work, we benchmark three distinct active learning settings for LiDAR semantic segmentation and deliver a simple and general baseline, *Annotator*, as illustrated in Figure 2. Borrowing the idea of modeling and computational techniques in geometry processing, we introduce a voxel-centric selection strategy dedicated to point clouds. Specifically, an input LiDAR scene is first voxelized into voxel grids, with a large voxel size to expand the local areas during the selection process. After obtaining final network predictions, importance estimation is carried out for each voxel grid using several common strategies such as Random, the softmax entropy (Entropy), and the margin between highest softmax scores (Margin). But considering only uncertainty for selection would be suboptimal [3, 50, 93]. Therefore, we introduce the concept of voxel confusion degree (VCD), which takes into account nearby predictions, capturing diversity and redundancy within a voxel grid. VCD enables the exploitation of local topology relations and point cloud structures. As a result, VCD can represent both uncertainty and diversity of a voxel grid in the LiDAR scene. In each active round, we query the top one voxel grid within each scan for annotation until the budget is exhausted. Despite the simplicity of our *Annotator*, it achieves performance on par with the fully-supervised counterpart requiring $1000\times$ fewer annotations and significantly outperforms all prevailing acquisition strategies.

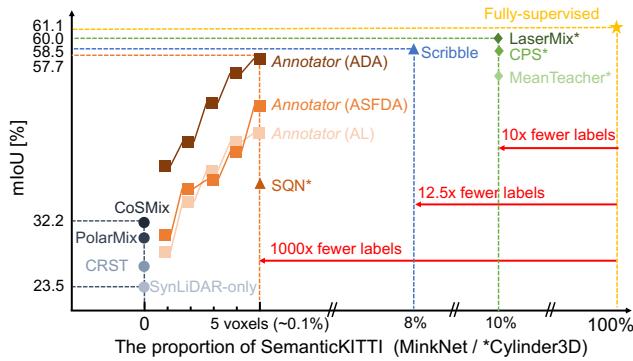


Figure 1: **Performance vs. annotated proportion** on SemanticKITTI val [4] of existing label-efficient LiDAR segmentation paradigms including domain adaptation (●) [61, 90, 91, 112], weakly- (▲) [25, 82] and semi-supervised (◆) [9, 33, 78] learning. As a reference, fully supervised counterpart (★) is reported as well. *Annotator* (■) attains excellent balance between performance and annotation cost.

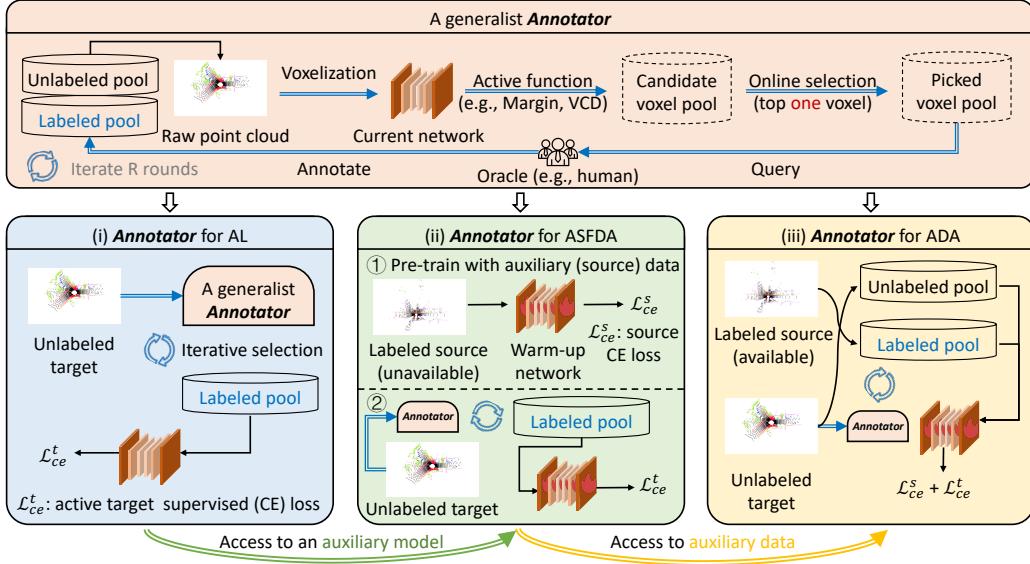


Figure 2: **An illustration of Annotator.** *Annotator* is a new active learning baseline with broad applicability, capable of interactively querying a tiny subset of the most informative new (target) data points based on available inputs without task-specific designs. This includes (i) only unlabeled new (target) data being available (active learning, AL); (ii) access to an auxiliary (source) pre-trained model (active source-free domain adaptation, ASFDA); and (iii) availability of labeled source data and unlabeled target data (active domain adaptation, ADA). Remarkably, *Annotator* attains excellent results not only in in-domain settings but also manifests adaptive transfer to out-of-domain settings.

The contribution of this paper can be summarized in three aspects. *First*, we present a voxel-centric active learning baseline that significantly reduces the labeling cost and effectively facilitates learning with a limited budget, achieving near performance to that of fully-supervised methods with $1000 \times$ fewer annotations. *Second*, we introduce a label acquisition strategy, the voxel confusion degree (VCD), which is more robust and diverse to select point clouds under a domain shift. *Third*, *Annotator* is generally applicable for various network architectures (voxel-, range- and bev-views), settings (in-distribution and out-of-distribution), and scenarios (simulation-to-real and real-to-real) with consistent gains. We hope this work could lay a solid foundation for label-efficient 3D applications.

2 A Generic Baseline

2.1 Preliminaries and overview

Problem setup. In the context of LiDAR semantic segmentation, a LiDAR scan is made of a set of point clouds and let $X \in \mathbb{R}^{N \times 4}$, $Y \in \mathbb{K}^N$ respectively denote N points and the corresponding labels. \mathbb{K} is a predefined semantic class vocabulary $\mathbb{K} = \{1, \dots, K\}$ of K categorical labels. Each point x_i in X is a 1×4 vector with a 3D Cartesian coordinate relative to the scanner (a_i, b_i, c_i) and an intensity value of returning laser beam. Our baseline works in the following settings: active learning (AL), active source-free domain adaptation (ASFDA), and active domain adaptation (ADA). First, we are given an unlabeled target domain $\mathcal{D}^t = \{X^t \cup X^a\}$, where X^t denotes unlabeled target point clouds and X^a denotes the selected points to be annotated and is initialized as empty set, i.e., $X^a = \emptyset$. Next, for ASFDA and ADA, a labeled source domain $\mathcal{D}^s = \{X^s, Y^s\}$ can be utilized only in pre-training stage and anytime respectively. Ultimately, given a limited budget, our goal is to iteratively select a subset of data points from \mathcal{D}^t to annotate until the budget is exhausted, all the while catching up with the performance of the fully-supervised model.

Overview. Figure 2 displays an overview of *Annotator*, which is a label-efficient baseline for LiDAR semantic segmentation. It is composed of two parts: 1) a generalist *Annotator* which contains a voxelization process to get voxel grids and an active function with online selection for picking the most valuable voxel grid of each input scan in each active round; 2) the pipelines of distinct active

learning settings are described. For AL, we interactively select a subset of voxels from the current scan to be annotated and train the network with these sparse annotated voxel grids. In the case of ASFDA, we begin by pre-training a network on the source domain through standard supervised learning. This warm-up network then serves as a strong initialization to aid the initial selection. As for ADA, except for the pre-training stage, we also make use of annotated source domain to promote the selection in each round and facilitate domain alignment. In the following, we will detail why we select salient and exemplar data points from a voxel-centric perspective and how to address *cold start problem* via an auxiliary model. After that, overall objectives for all three settings are elaborated.

2.2 A generalist *Annotator*

In this section, we proposed a general active learning baseline called *Annotator*. The core idea is to select salient and exemplar voxel grids from each LiDAR scan. It's important to note that previous researches have proposed frame-based [15, 101], region-based [89], and point-based [40] selection strategies. The first two usually require an offline stage, which may be infeasible at large scales. The last one is costly due to the sparsity of outdoor point clouds. By contrast, our voxel-centric selection focuses on querying salient and exemplar areas and annotating all points within those areas. This approach is more efficient and flexible. Moreover, it can be seamlessly applied to various network architectures, including voxel-, range- and bev-views, as demonstrated in the experiment section.

To implement it, we begin with the voxelization process as introduced in [10, 109]. Each input LiDAR scan X is transformed into a 3D voxel grid set V . This process involves sampling the continuous 3D input space into discrete voxel grids, where points falling into the same grid are merged. Each voxel grid serves as a selection unit. Mathematically, for a point $x_i \in X$, the corresponding voxel grid coordinate is $(a_i^v, b_i^v, c_i^v) = \lfloor (a_i, b_i, c_i)/\Delta \rfloor$, with Δ denoting predefined voxel size. In our experiments, we have found that using a large voxel grid is more robust against noise and sparsity. Unless otherwise specified, we use $\Delta_1 = 0.05$ for training and $\Delta_2 = 0.25$ for the selection process.

Selection strategies. For each voxel grid $v_j \in V$, we assess its importance and select the best voxel grid per LiDAR scan in each active round. Initially, we employ a Random selection strategy. Subsequently, we explore softmax entropy (Entropy) and the margin between highest softmax scores (Margin). It's essential to note that while these common selection strategies are not technical contributions, they are necessary to build our baseline. Detailed calculations are provided below.

- **Random:** randomly select a target voxel grid v_j from V to be annotated in each round.
- **Entropy:** first calculate the softmax entropy of each point $x_i \in v_j$ and then adopt the maximum value as the Entropy score of this grid, i.e., $\text{Entropy}(v_j) = \max_{x_i \in v_j} -p_i \log p_i$, where p_i is the softmax score of point x_i . The voxel grid with the highest Entropy score is selected in each scan.
- **Margin:** first calculate the margin between highest softmax score of each point $x_i \in v_j$ and then adopt the maximum value as the Margin score of this grid, i.e., $\text{Margin}(v_j) = \max_{x_i \in v_j} (\max(p_i) - \max_2(p_i))$, where $\max_2(\cdot)$ is the second-largest value operator. In each scan, the voxel grid with the lowest Margin score is chosen.

The VCD strategy. Our voxel confusion degree (VCD) is motivated by an important observation: the previously mentioned selection strategies become less effective when models are applied in new domains due to mis-calibrated uncertainty estimation. Therefore, the VCD is designed to estimate category diversity within a voxel grid rather than uncertainty, making it more robust under domain shift. Here's how it works: we begin by obtaining pseudo label \hat{y}_i for each point x_i . Next, we divide points within v_j into K clusters: $v_j^{<k>} = \{x_i^{<k>} | x_i \in v_j, \hat{y}_i = k\}$. This allows us to collect statistical information about the categories present in the voxel grid. With this information, we calculate VCD to assess the significance of voxel grids as follows:

$$\text{VCD}(v_j) = - \sum_{k=1}^K \frac{|v_j^{<k>}|}{|v_j|} \log \frac{|v_j^{<k>}|}{|v_j|},$$

where $|\cdot|$ denotes the number of points in a set. Finally, voxel grid with the highest VCD score is selected in each scan. The insight is that a higher score indicates a greater category diversity within a voxel, which would be beneficial for model training once being annotated. In all experiments, *Annotator* is equipped with VCD by default, and the results indicate the superiority of VCD strategy.

Making a good first impression. To avoid the *cold start problem* mentioned before, we introduce a warm start mechanism that pre-trains an auxiliary model with an auxiliary (source) dataset, and then it is used to select voxel grids in the first round. This warm start stage is applied in ASFDA and ADA.

Discussion: balancing annotation cost and computation cost. Our primary focus is on reducing annotation cost while maintaining performance comparable to fully-supervised approaches. Let’s consider simulation-to-real tasks as an example. The simplest setup involves active learning within the real dataset. However, this setup yields less satisfactory results due to the *cold-start problem*: the lack of prior information for selecting an initial annotated set. To address this, we utilize a synthetic dataset to train an auxiliary model in a brief warm-up stage, enabling smarter data selection in the first round. Importantly, this warm-up process is short, conducted only once, and results in minimal costs (both annotation and computation). For a detailed analysis, please refer to Appendix B.1.

2.3 Optimization

The overall loss function is the standard cross-entropy loss, which is defined as:

$$\mathcal{L}_{ce}(X) = \frac{1}{|X|} \sum_{x_i \in X} \sum_{k=1}^K -y_i^k \log p_i^k,$$

where K is the number of categories, y_i is the one-hot label of point x_i and p_i^k is the predicted probability of point x_i belonging to category k . Hereafter, for AL, the objective is $\min_{\theta} \mathcal{L}_{ce}(X^a)$; for ASFDA, the objective is $\min_{\theta_s} \mathcal{L}_{ce}(X^a)$; for ADA, the objective is $\min_{\theta_s} \mathcal{L}_{ce}(X^s) + \mathcal{L}_{ce}(X^a)$. Here, θ and θ_s denote training from scratch and training from the source pre-trained model, respectively.

3 Experiments

In this section, we conduct extensive experiments on several public benchmarks under three active learning scenarios: (i) AL setting where all available data points are from unlabeled target domain; (ii) ASFDA setting where we can only access a pre-trained model from the source domain; (iii) ADA setting where all data points from source domain can be utilized and a portion of unlabeled target data is selected to be annotated. We first introduce the dataset used in this work and experimental setup and then present experimental results of baseline methods and extensive analyses of *Annotator*.

3.1 Experiment setup

Datasets. We build all benchmarks upon SynLiDAR [91], SemanticKITTI [4], SemanticPOSS [48], and nuScenes [5], constructing two simulation-to-real and two real-to-real adaptation scenarios. SynLiDAR [91] is a large-scale synthetic dataset, which has 198,396 LiDAR scans with point-level segmentation annotations over 32 semantic classes. Following [91], we use 19,840 point clouds as the training data. SemanticKITTI (KITTI) [4] is a popular LiDAR segmentation dataset, including 2,9130 training scans and 6,019 validation scans with 19 categories. SemanticPOSS (POSS) [48] consists of 2,988 real-world scans with point-level annotations over 14 semantic classes. As suggested in [48], we use the sequence 03 for validation and the remaining sequences for training. nuScenes [5] contains 19,130 training scans and 4,071 validation scans with 16 object classes.

Class mapping. To ensure compatibility between source and target labels across datasets, we perform class mapping. Specifically, we map SynLiDAR labels into 19 common categories for SynLiDAR \rightarrow KITTI and 13 classes for SynLiDAR \rightarrow POSS. Similarly, we map labels into 7 classes for KITTI \rightarrow nuScenes and nuScenes \rightarrow KITTI. We refer readers to Appendix A.1 for detailed class mappings.

Implementation details. We primarily adopt MinkNet [10] and SPVCNN [77] as the segmentation backbones. Note that, all experiments share the same backbones and are within the same codebase, which are implemented using PyTorch [49] on a single NVIDIA Tesla A100 GPU. We use the SGD optimizer and adopt a cosine learning rate decay schedule with initial learning rate of 0.01. And the batch size for both source and target data is 16. For additional details, please consult Appendix A.2. Finally, we evaluate the segmentation performance before and after adaptation, following the typical evaluation protocol [53] in LiDAR domain adaptive semantic segmentation [33, 36, 61, 89, 90].

3.2 Experimental results

Quantitative results summary. We initially evaluate *Annotator* on four benchmarks and two backbones while adhering to a fixed budget of selecting and annotating five voxel grids in each scan.

Method	Simulation-to-Real						Real-to-Real					
	SynLiDAR $\xrightarrow{19}$ KITTI			SynLiDAR $\xrightarrow{13}$ POSS			KITTI $\xrightarrow{7}$ nuScenes			nuScenes $\xrightarrow{7}$ KITTI		
Source-/Target-Only	22.0 / 61.1			30.4 / 56.7			28.4 / 82.5			34.6 / 83.3		
Random	35.3 / 36.3 / 45.3			27.4 / 30.9 / 43.4			66.0 / 67.5 / 71.9			70.9 / 69.7 / 74.7		
Entropy [84]	39.8 / 49.6 / 50.1			42.8 / 45.5 / 49.9			59.7 / 60.3 / 73.1			70.7 / 69.1 / 74.0		
Margin [30]	46.9 / 44.3 / 49.0			41.6 / 44.1 / 46.9			60.2 / 59.2 / 71.4			73.1 / 70.3 / 76.7		
Annotator	53.7 / 54.1 / 57.7			44.9 / 48.2 / 52.0			70.4 / 72.4 / 75.9			76.8 / 75.3 / 81.8		

Table 1: Quantitative summary of all baselines’ performance based on MinkNet [10] over various LiDAR semantic segmentation benchmarks using only 5 voxel grids. Source-/Target-Only correspond to the model trained on the annotated source/target dataset which are considered as lower/upper bound. Note that results are reported following the order of AL / ASFDA / ADA in each cell.

Method	Simulation-to-Real						Real-to-Real					
	SynLiDAR $\xrightarrow{19}$ KITTI			SynLiDAR $\xrightarrow{13}$ POSS			KITTI $\xrightarrow{7}$ nuScenes			nuScenes $\xrightarrow{7}$ KITTI		
Source-/Target-Only	24.2 / 63.7			37.0 / 51.9			21.3 / 81.3			47.1 / 85.0		
Random	40.9 / 41.7 / 51.0			35.5 / 37.8 / 42.3			65.0 / 66.9 / 64.3			70.4 / 68.1 / 75.8		
Entropy [84]	52.7 / 52.1 / 52.8			35.2 / 40.5 / 46.8			61.3 / 66.0 / 66.3			69.5 / 67.4 / 72.6		
Margin [30]	47.1 / 49.9 / 50.7			42.9 / 44.8 / 47.1			57.8 / 60.3 / 63.2			72.3 / 73.0 / 75.3		
Annotator	52.8 / 54.6 / 55.6			44.9 / 47.5 / 50.9			71.4 / 72.1 / 72.3			79.5 / 80.5 / 78.4		

Table 2: Quantitative summary of all baselines’ performance based on SPVCNN [77] over various LiDAR semantic segmentation benchmarks using only 5 voxel grids.

Model	car	bicle	mtcle	truck	oth-v.	pers.	bclst	mclst	road	park.	sidew.	oth-g.	build.	fence	vegit.	trunk	terra.	pole	traff.	mIoU	
Source-Only	59.4	6.2	27.2	0.6	5.8	18.4	37.9	5.4	9.3	8.8	31.0	0.1	24.5	22.6	62.7	27.7	43.4	22.8	3.6	22.0	
DA	ADDA [81]	52.5	4.5	11.9	0.3	3.9	9.4	27.9	0.5	52.8	4.9	27.4	0.0	61.0	17.0	57.4	34.5	42.9	23.2	4.5	
	AdvEnt [83]	58.3	5.1	14.3	0.3	1.8	14.3	44.5	0.5	50.4	4.3	34.8	0.0	48.3	19.7	67.5	34.8	52.0	33.0	6.1	
	CRST [112]	62.0	5.0	12.4	1.3	9.2	16.7	44.2	0.4	53.0	2.5	28.4	0.0	57.1	18.7	69.8	35.0	48.7	32.5	6.9	
	ST-PCT [91]	70.8	7.3	13.1	1.9	8.4	12.6	44.0	0.6	56.4	4.5	31.8	0.0	66.7	23.7	73.3	34.6	48.4	39.4	11.7	
	CoSMix [61]	75.1	6.8	29.4	27.1	11.1	22.1	25.0	24.7	79.3	14.9	46.7	0.1	53.4	13.0	67.7	31.4	32.1	37.9	13.4	
	PolarMix [90]	76.3	8.4	17.8	3.9	6.0	26.6	40.8	15.9	70.3	0.0	44.4	0.0	68.4	14.7	69.6	38.1	37.1	40.6	10.6	
AL	Random	90.6	0.0	0.0	4.5	11.1	0.0	0.0	0.0	84.5	19.1	68.3	0.0	84.4	45.5	85.8	53.9	73.3	47.8	2.0	35.3
	Entropy [84]	94.2	0.0	19.8	23.4	24.7	6.4	0.0	0.2	79.0	19.5	62.4	2.4	85.1	50.4	86.9	56.5	74.2	52.9	18.6	39.8
	Margin [30]	92.0	0.0	35.9	45.3	34.0	40.7	61.0	0.0	80.5	19.8	67.0	0.1	80.6	47.3	83.4	55.5	67.2	51.6	30.0	46.9
	Annotator	94.5	0.3	40.3	56.3	46.8	63.1	76.9	0.2	84.0	23.4	69.2	2.0	87.4	51.9	85.8	62.6	70.6	61.6	43.6	53.7
ASFDA	Random	90.5	0.0	0.0	4.7	16.5	0.0	0.0	0.0	84.4	20.8	68.9	0.1	84.7	45.9	85.8	55.0	72.8	53.7	5.4	36.3
	Entropy [84]	94.1	0.0	40.7	42.6	36.1	54.1	59.9	0.3	81.1	19.3	66.3	3.3	84.6	47.8	86.3	59.6	74.4	61.4	31.0	49.6
	Margin [30]	90.1	0.0	34.5	32.5	31.1	39.6	55.6	0.0	79.2	17.8	65.1	0.0	79.0	43.5	83.1	54.7	65.8	49.6	20.0	44.3
	Annotator	94.4	0.3	34.5	78.1	47.8	59.8	60.9	1.7	84.4	21.5	70.2	3.2	87.2	54.4	86.4	65.2	73.6	60.6	44.0	54.1
ADA	Random	93.0	0.0	30.0	23.0	25.0	37.9	32.5	0.2	84.2	25.7	71.6	0.1	81.0	54.0	83.7	56.9	72.0	53.7	35.8	45.3
	Entropy [84]	94.1	16.9	50.2	47.1	31.4	60.2	81.2	6.6	62.9	12.6	58.1	0.1	80.4	52.7	83.0	53.2	64.7	57.5	39.6	50.1
	Margin [30]	92.5	0.0	39.3	58.2	30.2	51.0	76.1	0.0	82.4	22.8	68.6	0.8	69.7	51.2	77.8	55.5	61.6	57.4	36.0	49.0
	Annotator	95.2	22.0	59.7	69.0	49.4	63.4	82.1	3.6	84.1	28.9	71.4	1.7	85.4	58.8	85.6	60.1	73.2	60.3	41.6	57.7
Target-Only	95.7	20.4	63.9	70.3	45.5	65.0	78.5	0.0	93.5	49.6	81.0	0.2	91.1	63.8	87.2	68.5	72.3	64.4	49.1	61.1	

Table 3: Per-class results on task of SynLiDAR $\xrightarrow{19}$ KITTI (MinkNet [10]) using only 5 voxel budgets. Domain adaptation (DA) results are reported from [61, 90].

The results in Table 1 and Table 2 paint a clear picture overall: all baseline methods achieve significant improvements over the Source-Only model, especially for *Annotator* with VCD strategy, underscoring the success of the proposed voxel-centric online selection strategy. In particular, *Annotator* achieves the best results across all simulation-to-real and real-to-real tasks. For SynLiDAR \rightarrow KITTI task, *Annotator* achieves 87.8% / 88.5% / 94.4% fully-supervised performance under AL / ASFDA / ADA settings respectively. For SynLiDAR \rightarrow POSS task, they are 79.0% / 85.0% / 91.7% respectively. On the task of KITTI \rightarrow nuScenes, they are 85.3% / 87.8% / 92.0% respectively. And on the task of nuScenes \rightarrow KITTI, they are 92.2% / 90.3% / 98.2%, respectively. It is also clear that domain shift between simulation and real-world is more significant than those between real-world datasets. Therefore, simulation-to-real tasks show poorer performance. Further, we compare *Annotator* with additional AL algorithms and extend it to indoor semantic segmentation in Appendix B.3 and B.4.

Per-class performance. To sufficiently realize the capacity of our *Annotator*, we also provide the class-wise IoU scores on two simulation-to-real tasks (Table 3 and Table 4) for different algorithms and comparison results with state-of-the-art DA methods [61, 90]. Other results of the remainder

Model	car	bike	pers.	rider	grou.	buil.	fence	plants	trunk	pole	traf.	garb.	cone.	mIoU	
Source-Only	44.7	1.9	33.5	38.3	77.0	54.2	30.3	63.8	22.0	12.9	0.4	11.2	4.7	30.4	
DA	CRST [112]	22.0	6.8	23.5	31.8	60.3	58.2	9.1	63.2	18.9	41.6	1.9	13.5	1.0	27.1
	ST-PCT [91]	27.8	6.6	28.9	34.8	63.9	64.1	12.1	63.7	18.6	41.0	4.9	16.6	1.6	29.6
	CoSMix [61]	36.2	10.6	55.8	51.4	78.7	66.2	24.9	71.3	23.5	34.2	22.5	28.9	20.4	40.4
	PolarMix [90]	25.0	10.7	32.6	39.1	79.0	44.8	23.8	64.2	11.9	29.6	5.8	15.3	13.3	30.4
AL	Random	24.0	47.8	28.9	0.1	79.3	66.7	27.7	76.4	0.1	5.5	0.2	0.0	0.0	27.4
	Entropy [84]	37.8	39.6	58.9	45.2	75.2	56.3	38.6	69.7	39.3	23.7	36.1	1.6	34.2	42.8
	Margin [30]	30.1	44.2	55.3	46.8	79.2	63.7	44.0	74.3	34.1	23.4	34.5	9.7	1.2	41.6
	Annotator	41.0	50.1	49.3	52.0	78.5	66.4	56.4	73.4	31.1	29.6	34.5	15.7	6.2	44.9
ASFDA	Random	32.2	46.4	37.9	0.4	79.2	69.8	33.0	77.7	17.4	5.5	2.8	0.0	0.0	30.9
	Entropy [84]	32.1	46.5	65.7	58.0	74.4	62.9	45.5	69.6	41.5	34.5	33.7	13.2	14.3	45.5
	Margin [30]	30.2	47.6	59.5	44.5	79.7	66.8	51.7	73.5	28.6	30.1	35.2	25.2	0.1	44.1
	Annotator	56.2	54.2	63.6	58.7	80.9	64.6	58.1	73.4	37.8	26.3	34.0	6.3	11.9	48.2
ADA	Random	65.0	10.9	59.3	54.3	58.6	70.0	54.2	63.9	39.6	39.8	20.8	27.8	0.0	43.4
	Entropy [84]	53.3	29.1	62.9	52.7	80.3	71.9	48.2	72.3	38.9	30.0	27.6	44.2	37.8	49.9
	Margin [30]	61.3	25.2	60.6	56.2	79.6	54.2	46.6	66.7	38.1	29.2	30.8	40.8	20.4	46.9
	Annotator	67.4	18.0	64.0	52.0	78.5	61.5	1.5	68.6	48.5	32.7	37.9	50.8	43.8	52.0
Target-Only	73.7	60.4	68.6	62.2	81.7	79.2	60.8	78.9	36.5	31.2	44.1	12.9	46.6	56.7	

Table 4: Per-class results on task of SynLiDAR $\xrightarrow{13}$ POSS (MinkNet [10]) using only 5 voxel budgets.

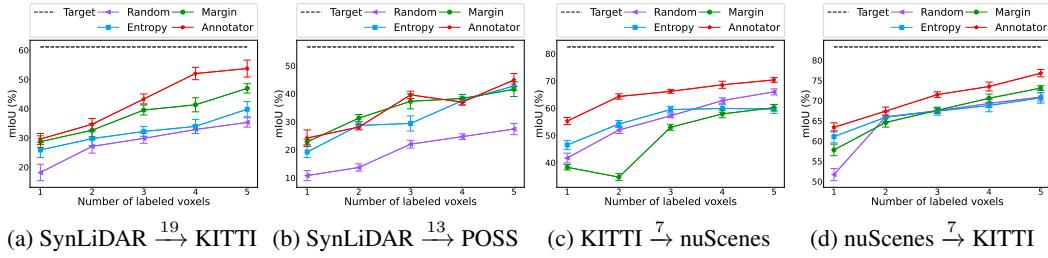


Figure 3: Active learning results on various benchmarks varying active budget.

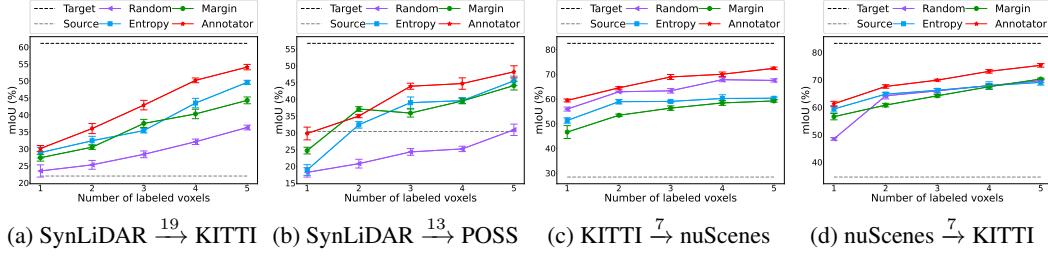


Figure 4: Active source-free domain adaptation results on various benchmarks varying active budget.

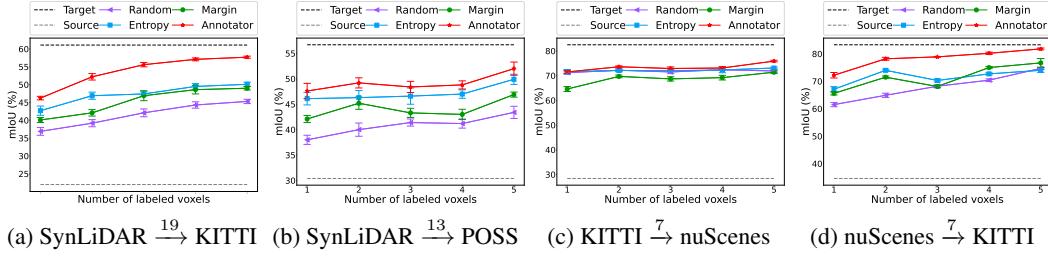


Figure 5: Active domain adaptation results on various benchmarks varying active budget.

tasks and backbones are listed in Appendix B.5. It is noteworthy that *Annotator* under any active learning settings significantly outperform DA methods with respect to some specific categories such as “traf.”, “pole”, “garb.” and “cone” etc. These results also showcase the class-balanced selection of the proposed *Annotator*, which is testified in Figure 6 as well.

Results with varying budgets. We investigate the impact of varying budgets and compare the performance with baseline methods, as illustrated in Figure 3, Figure 4 and Figure 5. A consistent

Model	car	bike	pers.	rider	grou.	buil.	fence	plants	trunk	pole	traf.	garb.	cone.	mIoU	
SalsaNet	Random	30.9	40.6	22.8	10.4	74.7	57.2	26.6	66.6	15.6	5.5	8.3	0.0	10.6	28.4
	Entropy	32.8	45.2	33.1	18.6	76.8	52.6	40.2	64.7	20.1	5.5	11.2	12.7	4.3	32.1
	Margin	29.8	38.2	33.6	28.0	71.1	48.0	27.7	61.3	24.5	12.5	20.6	18.4	0.0	31.8
	Annotator	32.0	45.1	39.7	31.8	76.5	53.9	40.9	64.7	26.2	11.8	17.7	13.7	13.1	35.9
	Target-Only	39.2	51.0	52.7	40.2	79.3	66.1	50.1	71.5	28.1	18.7	28.3	8.0	16.7	42.3
PolarNet	Random	36.6	50.5	40.8	0.1	76.1	69.3	50.3	74.0	3.1	17.8	1.3	0.0	0.0	32.3
	Entropy	44.5	48.8	50.3	11.8	77.9	63.6	45.4	71.0	10.0	13.3	19.1	0.0	0.0	35.0
	Margin	22.0	35.4	42.8	24.3	64.1	54.7	33.0	64.2	19.4	20.1	17.5	4.0	0.0	30.9
	Annotator	44.4	51.7	55.9	39.2	76.2	64.3	51.9	70.3	22.4	18.6	28.7	6.9	21.7	42.5
	Target-Only	66.3	57.2	62.3	51.8	80.8	74.9	61.3	75.5	22.8	21.8	29.4	4.8	46.1	50.4

Table 5: Per-class results on the SemanticPOSS val (range-view: SalsaNet [11] and bev-view: PolarNet [107]) under active learning setting using only 10 voxel budgets.

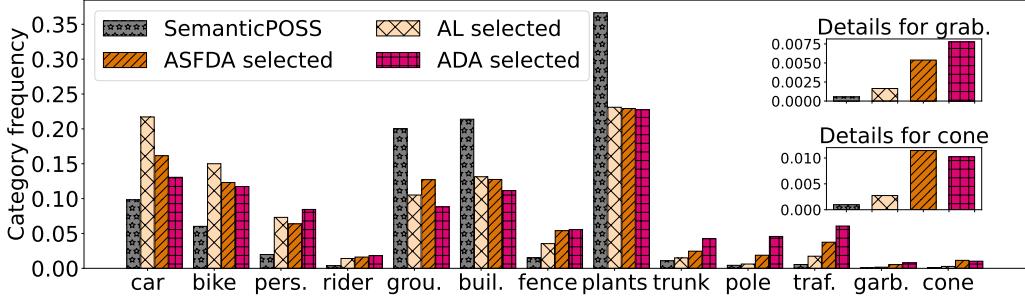


Figure 6: **Category frequencies** on SemanticPOSS train [48] of Annotator selected 5 voxel grids under AL, ASFDA, ADA scenarios, with the model trained on SynLiDAR $\xrightarrow{13}$ POSS (MinkNet [10]).

observation across these experiments is that *Annotator* consistently outperforms the baseline methods regardless of the budget allocation. In particular, *Annotator* achieves the best performance with about five voxel grids of each LiDAR scan, highlighting the effectiveness of our method in selecting informative areas for active learning. Additionally, we notice that the performance of *Annotator* tends to saturate when the budget exceeds four voxel grids, particularly in the nuScenes \rightarrow KITTI adaptation task. This phenomenon can be attributed to the fact that the selected voxels at this point provide a sufficient foundation for training a highly competent segmentation model.

3.3 Analysis

More network architectures. As a general baseline, *Annotator* can be easily applied to other non-voxelization based backbones. Here, we conduct experiments on both SalsaNext [11] (range-view) and PolarNet [107] (bev-view) and per-class results are presented in Table 5. The findings reveal that *Annotator* continues to yield significant gains, even when applied to range- or bev-view backbones, with a limited budget. However, the performance gains in these cases are somewhat less pronounced compared to the voxel-view counterparts. Also, to achieve a fully-supervised performance of 85%, a budget twice as large is required. We suspect that some annotations derived from voxel-centric selection may not be entirely applicable to other non-voxelization based methods.

Effect of voxel size Δ_2 . We conduct experiments on different Δ_2 while keeping the same budget for selection process and results are listed in Table 6. We can observe that active rounds (# round) decreases as Δ_2 increases since the number of voxels (# voxel) will be small when Δ_2 is large. Notably, the performance of the large voxel grid (≥ 0.2) is more adequate and robust.

Δ	0.05	0.1	0.15	0.2	0.25	0.3	0.35
# voxel	64973	54543	43795	36091	30414	25992	22539
# round	11	9	7	6	5	4	4
AL	39.6	42.9	43.9	44.2	44.9	45.1	44.8
ASFDA	40.0	46.2	46.0	48.0	48.2	48.3	48.0
ADA	44.5	44.1	49.3	52.4	52.0	52.1	51.4

Table 6: Experiments on different values of Δ_2 (from 0.05 to 0.35) for selection process, conducted on SynLiDAR $\xrightarrow{13}$ POSS (MinkNet [10]).

Category frequency. To obtain deep insight into *Annotator*, we also visualize a detailed plot of the class frequencies of 5 voxel grids selected by *Annotator* (AL, ASFDA, ADA) and true class

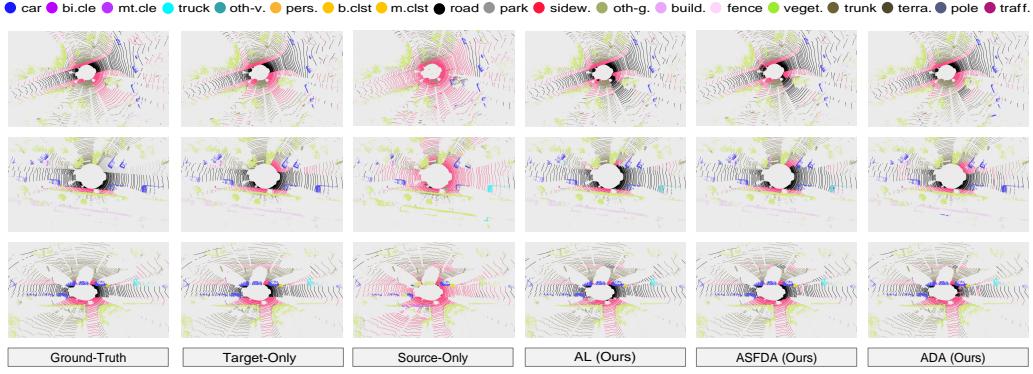


Figure 7: **Visualization of segmentation results** for the task $\text{SynLiDAR} \xrightarrow{19} \text{KITTI}$ using MinkNet [10]. Each row shows results of Ground-Truth, Target-Only, Source-Only, our *Annotator* under AL, ASFDA, and ADA scenarios one by one. Best viewed in color.

frequencies of the SemanticPOSS train in Figure 6. As expected, we clearly see that the true distribution is exactly a long-tail distribution while *Annotator* is able to pick out more voxels that contain rare classes. Particularly, it asks labels for more annotations of “rider”, “pole”, “trunk”, “traf.”, “grab.” and “cone”. This, along with the apparent gains in these classes in Table 4, confirms the diversity and balance of voxel grids selected by *Annotator*.

Qualitative results. Figure 7 visualizes segmentation results for Source-/Target-Only, our *Annotator* under AL, ASFDA, and ADA approaches on SemanticKITTI val. The illustration demonstrates the ability of *Annotator* to enhance predictions not only in distant regions but also to effectively eliminate false predictions across a wide range of directions around the center. By employing voxel-centric selection, *Annotator* succeeds in enhancing segmentation accuracy even when faced with extremely limited annotation availability. More qualitative results are shown in Appendix B.6.

3.4 Limitations

Currently, *Annotator* has two main limitations. First, high annotation cost and potential biases. *Annotator* has made substantial strides in enhancing LiDAR semantic segmentation with human involvement. Nonetheless, the annotation cost remains a challenge. It’s imperative to acknowledge the existence of label and sensor biases, which can be a safety concern in real-world deployments. Second, expansion beyond semantic segmentation. *Annotator* current focus on LiDAR semantic segmentation represents a significant limitation in fully realizing its potential. In the future work, we plan to extend *Annotator* to other 3D tasks, such as LiDAR object detection. This may involve two key changes: i) shifting to frame-level selection; ii) reformulating the VCD strategy to consider the diversity for each box annotation.

4 Related Work

LiDAR perception. Deep learning has made LiDAR perception tasks such as classification [20, 55, 76, 105] and detection [8, 34, 71, 100] easy to solve, allowing deployment in outdoor scenarios. Differently, LiDAR semantic segmentation [4, 27, 37, 38, 45, 51, 52], receiving a class label for each point, is an indispensable technology to understand a scene that is beyond the scope of modern object detectors [40]. There exist various techniques to segment the 3D LiDAR point clouds, e.g., point [37, 51, 86], voxel [21, 44, 110], range [87, 88], bird’s eye [107], and multiple view [74, 96, 103] methods. As the best approaches for LiDAR perception are typically trained under full supervision, which can be costly more than capturing data itself, several methods resort to more frugal learning techniques [16], such as semi- [33, 36], weak- [25, 39] and self-supervision [79, 108], zero-shot [24] and few-shot [69] learning and, as studied here, active learning [72] and domain adaptation [80].

Active learning for LiDAR point clouds. To avoid the burden of complete point cloud annotation, these methods iteratively select and request the most exemplar scans [101], regions [89], points [40], or

boxes [42] to be labeled during the network training. Most selection strategies lean on uncertainty [28, 95] or diversity [66, 98] criteria. Uncertainty sampling can be measured over each point prediction scores of the model, e.g., softmax entropy [84] or the margin between the two highest scores [59], to select the most confusing of the current model. For example, Hu *et al.* [28] estimate the inconsistency across frames to exploit the inter-frame uncertainty embedded in LiDAR sequences. On the other side, diversity sampling has been ensured by selecting core sets [65]. Leveraging the unique geometric structure of LiDAR point clouds, Liu *et al.* [40] partition the point could into a collection of components then annotate a few points for each component. Recently, the need for an initially annotated fraction of the data to bootstrap an active learning method has been investigated [7, 23, 64, 102, 111], which is termed as cold start problem. In this work, we show that a smart selection of the first set of data with the aid of an auxiliary model can boost all baseline methods drastically.

Domain adaptation for LiDAR point clouds. To tackle the sensor-bias problem encountered in LiDAR deployment, a large body of literature on domain adaptation (DA) [13, 32, 60–63, 80, 99] has been developed. These methods aim to overcome the challenges posed by variations in data collection, sensor characteristics, and environmental conditions, enabling machines to perceive the real world more accurately and reliably. To name a few, Kong *et al.* [32] explore cross-city adaptation for uni-modal LiDAR segmentation. Rochan *et al.* [57] propose a self-supervised adaptation technique with gated adapters. Saltori *et al.* [61] mitigate the domain shift by creating two new intermediate domains via sample mixing. Similarly, with the intermediate domain, Ding *et al.* [13] propose a data-oriented framework with a pretraining and a self-training stage for 3D indoor scenes. Despite the significant progress made in DA, the label scarcity of target domain severely handicaps its utility as the performance of such models often lags far behind the supervised learning counterparts. With this consideration, given an acceptable annotation budget, we explore a simple annotating strategy to assist adaptation process and significantly boost the performance of target domain.

Up to now, active learning coupled with domain adaptation has great practical significance [17, 43, 47, 50, 54, 68, 67, 73, 75, 93]. Nevertheless, rather little work has been done to consider the problem in 3D domains. A recent effort, UniDA3D [15], effectively tackles domain adaptation and active domain adaptation tasks for 3D semantic segmentation. UniDA3D employs a unified multi-modal sampling strategy, selecting informative pairs of 2D-3D data from both source and target domains through a domain discriminator, primarily for ADA tasks. The primary distinction is that our *Annotator* serves as a benchmark for active learning, active source-free domain adaptation, and active domain adaptation tasks, delivering a simple and general AL algorithm for LiDAR point clouds. *Annotator* focuses on enabling AL in both in-distribution and out-of-distribution scenarios. In contrast, UniDA3D places a greater emphasis on adaptation tasks. On the other hand, *Annotator* minimizes human labor in a new domain, regardless of the availability of samples from an auxiliary domain. Methodically, *Annotator* adopts a voxel-centric representation for structured LiDAR data, which is different from the scan-based representation in UniDA3D. Furthermore, *Annotator* is more efficient than UniDA3D in terms of both computation and annotation cost.

5 Conclusion

In this work, we present *Annotator*, a generalist active learning baseline, to tackle LiDAR semantic segmentation under three distinct label-efficient settings: active learning (AL), active source-free domain adaptation (ASFDA), and active domain adaptation (ADA). *Annotator* harnesses the power of a purpose-designed voxel confusion degree selection strategy, enabling it to make optimal use of limited budgets while achieving efficient selection and effective performance. Experiments conducted on widely-used simulation-to-real and real-to-real LiDAR semantic segmentation benchmarks demonstrate a substantial performance improvement. Looking forward, we believe the effectiveness and simplicity of *Annotator* has the potential to serve as a powerful tool for label-efficient 3D applications.

Acknowledgements

This paper was supported by National Key R&D Program of China (No. 2021YFB3301503), the National Natural Science Foundation of China (No. 62376026), and also sponsored by Beijing Nova Program (No. 20230484296). We thank Lingdong Kong for helpful discussions on this project.

References

- [1] I. Achituv, H. Maron, and G. Chechik. Self-supervised learning for domain adaptation on point clouds. In *WACV*, pages 123–133, 2021.
- [2] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.
- [3] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020.
- [4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9296–9306, 2019.
- [5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.
- [6] D. S. Chaplot, M. Dalal, S. Gupta, J. Malik, and R. Salakhutdinov. SEAL: Self-supervised embodied active learning using exploration and 3d consistency. In *NeurIPS*, pages 13086–13098, 2021.
- [7] L. Chen, Y. Bai, S. Huang, Y. Lu, B. Wen, A. L. Yuille, and Z. Zhou. Making your first choice: To address cold start problem in vision active learning. *CoRR*, abs/2210.02442, 2022.
- [8] Q. Chen, L. Sun, E. Cheung, and A. L. Yuille. Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization. In *NeurIPS*, pages 21224–21235, 2020.
- [9] X. Chen, Y. Yuan, G. Zeng, and J. Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, pages 2613–2622, 2021.
- [10] C. B. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019.
- [11] T. Cortinhal, G. Tzelepis, and E. E. Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *ISVC*, pages 207–222, 2020.
- [12] S. Deng, Q. Dong, B. Liu, and Z. Hu. Superpoint-guided semi-supervised semantic segmentation of 3d point clouds. In *ICRA*, pages 9214–9220, 2022.
- [13] R. Ding, J. Yang, L. Jiang, and X. Qi. Doda: Data-oriented sim-to-real domain adaptation for 3d semantic segmentation. In *ECCV*, pages 284–303, 2022.
- [14] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun. CARLA: an open urban driving simulator. In *CoRL*, pages 1–16, 2017.
- [15] B. Fei, S. Huang, J. Yuan, B. Shi, B. Zhang, W. Yang, M. Dou, and Y. Li. Unida3d: Unified domain adaptive 3d semantic segmentation pipeline. *CoRR*, abs/2212.10390, 2022.
- [16] B. Fei, W. Yang, L. Liu, T. Luo, R. Zhang, Y. Li, and Y. He. Self-supervised learning for pre-training 3d point clouds: A survey. *CoRR*, abs/2305.04691, 2023.
- [17] B. Fu, Z. Cao, J. Wang, and M. Long. Transferable query selection for active domain adaptation. In *CVPR*, pages 7272–7281, 2021.
- [18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 32(11):1231–1237, 2013.
- [19] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.
- [20] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *ICML*, pages 3809–3820, 2021.
- [21] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018.
- [22] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4338–4364, 2020.

- [23] N. Houlsby, J. M. Hernández-Lobato, and Z. Ghahramani. Cold-start active learning with robust ordinal matrix factorization. In *ICML*, pages 766–774, 2014.
- [24] P. Hu, S. Sclaroff, and K. Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. In *NeurIPS*, pages 21713–21724, 2020.
- [25] Q. Hu, B. Yang, G. Fang, Y. Guo, A. Leonardis, N. Trigoni, and A. Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. In *ECCV*, pages 600–619, 2022.
- [26] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, pages 11105–11114, 2020.
- [27] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):8338–8354, 2022.
- [28] Z. Hu, X. Bai, R. Zhang, X. Wang, G. Sun, H. Fu, and C.-L. Tai. Lidal: Inter-frame uncertainty based active learning for 3d lidar semantic segmentation. In *ECCV*, pages 248–265, 2022.
- [29] B. Hurl, K. Czarnecki, and S. L. Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *IV*, pages 2522–2529, 2019.
- [30] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379, 2009.
- [31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. B. Girshick. Segment anything. *CoRR*, abs/2304.02643, 2023.
- [32] L. Kong, N. Quader, and V. E. Liong. Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. In *ICRA*, pages 9338–9345, 2023.
- [33] L. Kong, J. Ren, L. Pan, and Z. Liu. Lasermix for semi-supervised lidar semantic segmentation. In *CVPR*, pages 21705–21715, 2023.
- [34] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019.
- [35] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss. Domain transfer for semantic segmentation of lidar data using deep neural networks. In *IROS*, pages 8263–8270, 2020.
- [36] L. Li, H. P. Shum, and T. P. Breckon. Less is more: Reducing task and model complexity for 3d point cloud semantic segmentation. In *CVPR*, pages 9361–9371, 2023.
- [37] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, pages 828–838, 2018.
- [38] K. Liu. Rm3d: Robust data-efficient 3d scene parsing via traditional and learnt 3d descriptors-based semantic region merging. *Int. J. Comput. Vis.*, 131(4):938–967, 2023.
- [39] K. Liu, Y. Zhao, Q. Nie, Z. Gao, and B. M. Chen. Weakly supervised 3d scene segmentation with region-level boundary awareness and instance discrimination. In *ECCV*, pages 37–55, 2022.
- [40] M. Liu, Y. Zhou, C. R. Qi, B. Gong, H. Su, and D. Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. In *ECCV*, pages 70–89, 2022.
- [41] Z. Liu, X. Qi, and C.-W. Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *CVPR*, pages 1726–1736, 2021.
- [42] Y. Luo, Z. Chen, Z. Wang, X. Yu, Z. Huang, and M. Baktashmotlagh. Exploring active 3d object detection from a generalization perspective. In *ICLR*, 2023.
- [43] A. D. mathelin, F. Deheeger, M. MOUGEOT, and N. Vayatis. Discrepancy-based active learning for domain adaptation. In *ICLR*, 2022.
- [44] H. Meng, L. Gao, Y. Lai, and D. Manocha. Vv-net: Voxel VAE net with group convolutions for point cloud segmentation. In *ICCV*, pages 8499–8507, 2019.
- [45] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220, 2019.

- [46] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *3DV*, pages 116–125, 2021.
- [47] M. Ning, D. Lu, D. Wei, C. Bian, C. Yuan, S. Yu, K. Ma, and Y. Zheng. Multi-anchor active domain adaptation for semantic segmentation. In *ICCV*, pages 9112–9122, 2021.
- [48] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IV*, pages 687–693, 2020.
- [49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.
- [50] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *ICCV*, pages 8505–8514, 2021.
- [51] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017.
- [52] H. Qiu, B. Yu, and D. Tao. GFNet: Geometric flow network for 3d point cloud semantic segmentation. *Trans. on Mach. Learn. Res.*, 2022.
- [53] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *ISVC*, pages 234–244, 2016.
- [54] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.
- [55] J. Ren, L. Pan, and Z. Liu. Benchmarking and analyzing point cloud classification under corruptions. In *ICML*, pages 18559–18575, 2022.
- [56] S. E. Reutebuch, H.-E. Andersen, and R. J. McGaughey. Light detection and ranging (lidar): an emerging tool for multiple resource inventory. *Journal of forestry*, 103(6):286–292, 2005.
- [57] M. Rochan, S. Aich, E. R. Corral-Soto, A. Nabatchian, and B. Liu. Unsupervised domain adaptation in lidar semantic segmentation with self-supervision and gated adapters. In *ICRA*, pages 2649–2655, 2022.
- [58] R. Roriz, J. Cabral, and T. Gomes. Automotive lidar technology: A survey. *IEEE Trans. Intell. Transp. Syst.*, 23(7):6282–6297, 2022.
- [59] D. Roth and K. Small. Margin-based active learning for structured output spaces. In *ECML*, pages 413–424, 2006.
- [60] K. Ryu, S. Hwang, and J. Park. Instant domain augmentation for lidar semantic segmentation. In *CVPR*, pages 9350–9360, 2023.
- [61] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In *ECCV*, pages 586–602, 2022.
- [62] C. Saltori, E. Krivosheev, S. Lathuilière, N. Sebe, F. Galasso, G. Fiameni, E. Ricci, and F. Poiesi. GIPSO: geometrically informed propagation for online adaptation in 3d lidar segmentation. In *ECCV*, pages 567–585, 2022.
- [63] C. Saltori, A. Osep, E. Ricci, and L. Leal-Taixé. Walking your lidog: A journey through multiple domains for lidar semantic segmentation. In *ICCV*, pages 196–206, 2023.
- [64] N. Samet, O. Siméoni, G. Puy, G. Ponimatkina, R. Marlet, and V. Lepetit. You never get a second chance to make a good first impression: Seeding active learning for 3d semantic segmentation. In *ICCV*, pages 18445–18457, 2023.
- [65] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.
- [66] F. Shao, Y. Luo, P. Liu, J. Chen, Y. Yang, Y. Lu, and J. Xiao. Active learning for point cloud semantic segmentation via spatial-structural diversity reasoning. In *MM*, pages 2575–2585, 2022.
- [67] J.-J. Shao, L.-Z. Guo, X.-W. Yang, and Y.-F. Li. Log: Active model adaptation for label-efficient ood generalization. In *NeurIPS*, pages 11023–11034, 2022.

- [68] J.-J. Shao, Y. Xu, Z. Cheng, and Y.-F. Li. Active model adaptation under unknown shift. In *KDD*, pages 1558–1566, 2022.
- [69] C. Sharma and M. Kaul. Self-supervised few-shot learning on point clouds. In *NeurIPS*, pages 7212–7221, 2020.
- [70] Y. Shen, Y. Yang, M. Yan, H. Wang, Y. Zheng, and L. J. Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *CVPR*, pages 7223–7232, 2022.
- [71] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li. PV-RCNN++: point-voxel feature set abstraction with local vector representation for 3d object detection. *Int. J. Comput. Vis.*, 131(2):531–551, 2023.
- [72] X. Shi, X. Xu, K. Chen, L. Cai, C. S. Foo, and K. Jia. Label-efficient point cloud semantic segmentation: An active learning approach. *CoRR*, abs/2101.06931, 2021.
- [73] I. Shin, D. jin Kim, J. W. Cho, S. Woo, K. Park, and I. S. Kweon. Labor: Labeling only if required for domain adaptive semantic segmentation. In *ICCV*, pages 8588–8598, 2021.
- [74] I. Shin, Y.-H. Tsai, B. Zhuang, S. Schulter, B. Liu, S. Garg, I. S. Kweon, and K.-J. Yoon. Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In *CVPR*, pages 16928–16937, 2022.
- [75] J. Su, Y. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker. Active adversarial domain adaptation. In *WACV*, pages 728–737, 2020.
- [76] J. Sun, Y. Cao, C. B. Choy, Z. Yu, A. Anandkumar, Z. M. Mao, and C. Xiao. Adversarially robust 3d point cloud recognition using self-supervisions. In *NeurIPS*, pages 15498–15512, 2021.
- [77] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, pages 685–702, 2020.
- [78] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.
- [79] X. Tian, H. Ran, Y. Wang, and H. Zhao. Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In *CVPR*, pages 13570–13580, 2023.
- [80] L. T. Triessig, M. Dreissig, C. B. Rist, and J. M. Zöllner. A survey on deep domain adaptation for lidar perception. In *IV Workshops*, pages 350–357, 2021.
- [81] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.
- [82] O. Unal, D. Dai, and L. Van Gool. Scribble-supervised lidar semantic segmentation. In *CVPR*, pages 2697–2707, 2022.
- [83] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, pages 2517–2526, 2019.
- [84] D. Wang and Y. Shang. A new active labeling method for deep learning. In *IJCNN*, pages 112–119, 2014.
- [85] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. E. Campbell, K. Q. Weinberger, and W. Chao. Train in germany, test in the USA: making 3d object detectors generalize. In *CVPR*, pages 11710–11720, 2020.
- [86] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *CoRR*, abs/1807.06288, 2018.
- [87] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, pages 1887–1893, 2018.
- [88] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, pages 4376–4382, 2019.
- [89] T.-H. Wu, Y.-C. Liu, Y.-K. Huang, H.-Y. Lee, H.-T. Su, P.-C. Huang, and W. H. Hsu. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In *ICCV*, pages 15510–15519, 2021.
- [90] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu, and L. Shao. Polarmix: A general data augmentation technique for lidar point clouds. In *NeurIPS*, pages 11035–11048, 2022.

- [91] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI*, pages 2795–2803, 2022.
- [92] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, and G. Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(7):9004–9021, 2023.
- [93] B. Xie, L. Yuan, S. Li, C. H. Liu, and X. Cheng. Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In *CVPR*, pages 8058–8068, 2022.
- [94] B. Xie, L. Yuan, S. Li, C. H. Liu, X. Cheng, and G. Wang. Active learning for domain adaptation: An energy-based approach. In *AAAI*, pages 8708–8716, 2022.
- [95] M. Xie, S. Li, R. Zhang, and C. H. Liu. Dirichlet-based uncertainty calibration for active domain adaptation. In *ICLR*, 2023.
- [96] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu. Rpnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *ICCV*, pages 16004–16013, 2021.
- [97] X. Xu and G. H. Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020.
- [98] S. Ye, Z. Yin, Y. Fu, H. Lin, and Z. Pan. A multi-granularity semisupervised active learning for point cloud semantic segmentation. *Neural Comput. Appl.*, 35(21):15629–15645, 2023.
- [99] L. Yi, B. Gong, and T. Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *CVPR*, pages 15363–15373, 2021.
- [100] Y. You, Y. Wang, W. Chao, D. Garg, G. Pleiss, B. Hariharan, M. E. Campbell, and K. Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *ICLR*, 2020.
- [101] J. Yuan, B. Zhang, X. Yan, T. Chen, B. Shi, Y. Li, and Y. Qiao. Bi3d: Bi-domain active learning for cross-domain 3d object detection. In *CVPR*, pages 15599–15608, 2023.
- [102] M. Yuan, H.-T. Lin, and J. Boyd-Graber. Cold-start active learning through self-supervised language modeling. *CoRR*, abs/2010.09535, 2020.
- [103] Q. Zhang, J. Hou, Y. Qian, Y. Zeng, J. Zhang, and Y. He. Flattening-net: Deep regular 2d representation for 3d point cloud analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):9726–9742, 2023.
- [104] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *NeurIPS*, pages 27061–27074, 2022.
- [105] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li. Pointclip: Point cloud understanding by CLIP. In *CVPR*, pages 8542–8552, 2022.
- [106] Y. Zhang, Y. Qu, Y. Xie, Z. Li, S. Zheng, and C. Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, pages 15520–15528, 2021.
- [107] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, pages 9598–9607, 2020.
- [108] Z. Zhang, M. Bai, and E. L. Li. Self-supervised pretraining for large-scale point clouds. In *NeurIPS*, pages 37806–37821, 2022.
- [109] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018.
- [110] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):6807–6822, 2021.
- [111] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, and D. Cai. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Trans. Knowl. Data Eng.*, 32(4):631–644, 2019.
- [112] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang. Confidence regularized self-training. In *ICCV*, pages 5982–5991, 2019.

A Implementation details

A.1 Dataset

SynLiDAR [91] is a large-scale synthetic dataset that is captured with the Unreal Engine [14]. It has 13 LiDAR point cloud sequences with 198,396 scans in total, where each scan has around 98,000 points on average. Precise point-wise annotations of 32 semantic classes are provided for fine-grained 3D scene understanding. It includes 12 LiDAR point cloud sequences (sequence 00 to 11) and has 19,840 point clouds for training following the authors' instructions [91].

SemanticKITTI [4] is a comprehensive autonomous driving dataset consisting of LiDAR acquisitions of famous KITTI Vision Odometry Benchmark [18, 19]. The LiDAR point clouds are captured in Karlsruhe (Germany) by a 64-beam LiDAR sensor, with point-level annotations over 19 semantic classes. It includes 22 LiDAR point cloud sequences that are split into a **train** set (sequence 00 to 10, where 08 is used for validation) and a **test** set (sequence 11 to 21). Following [61, 63, 90, 91], we do not use the **test** set, and only use the **train** set for training and validation in all experiments.

SemanticPOSS [48] consists of 2,988 real-world scans with point-level annotations over 14 semantic classes. The data is collected in Peking University and uses the same data format as SemanticKITTI. It includes 6 LiDAR point cloud sequences (sequence 00 to 05) and we use the sequence 03 for validation and the remaining sequences for training based on the official benchmark guidelines [48].

SynLiDAR	SemanticKITTI	SemanticPOSS	nuScenes	19 →	13 →	7 →
car	car	car	vehicle.car	1	1	1
	moving-car					
bicycle	bicycle	bike	vehicle.bicycle	2	2	
	truck				4	
truck	moving-truck					
motorcycle	motorcycle				3	
bus	bus					5
	moving-bus					
sidewalk	sidewalk		flat.sidewalk	11	4	
female			human.pedestrian.adult			
male	person	1 person	human.pedestrian.police_officer			
kid	moving-person	2+ person	human.pedestrian.child	6	3	2
			human.pedestrian.construction_worker			
vegetation	vegetation	plants		15	8	
road	road		flat.driveable_surface	9	5	3
	lane-marking	ground				
terrain	terrain		flat.terrain	17	5	
other-ground	other-ground		flat.other	12		
pole	pole	pole		18	10	6
	other-vehicle					
	on-rails				5	
other-vehicle	moving-on-rails					
	moving-other					
building	building	building		13	6	6
bicyclist	bicyclist			7	4	4
	moving-bicyclist					
trunk	trunk	trunk		16	9	7
traffic-sign	traffic-sign		traffic sign 1			
			traffic sign 2	19	11	6
			traffic sign 3			
parking	parking			10	3	
motorcyclist	motorcyclist			8	4	
	moving-motorcyclist					
fence	fence	fence		14	7	6
garbage-can		garbage-can			12	
traffic-cone		cone/stone	movable.trafficcone		13	
		rider			4	
			static.manmade			6

Table A1: Unified label space for SynLiDAR, SemanticKITTI, SemanticPOSS, nuScenes: there are over 50 object categories and we list them for individual datasets. In details, we also list training IDs for SynLiDAR $\xrightarrow{19}$ KITTI, SynLiDAR $\xrightarrow{13}$ POSS, KITTI $\xrightarrow{7}$ nuScenes, and nuScenes $\xrightarrow{7}$ KITTI.

nuScenes [5] is another large-scale LiDAR segmentation dataset widely adopted in academia. It provides 1,000 driving scenes, where each scene is collected by a 32-beam LiDAR sensor from Boston and Singapore. We follow the official `train` and `val` sample splittings. The total number of LiDAR scans is 40000. The training and validation sets contain 28130 and 6019 scans, respectively.

Class mapping. To ensure all tasks are well-defined, we formalize consistent and compatible semantic class vocabulary across the above datasets, ensuring there is a one-to-one mapping between all semantic classes. Table A1 summarizes the unified label space for SynLiDAR [91], SemanticKITTI [4], SemanticPOSS [48], and nuScenes [5].

A.2 Training details

Model configuration. For our main experiments, we employ two common network architectures: MinkNet [10] and SPVCNN [77]. The voxel size $\Delta_1 = 0.05$ for training and we adopt coordinates and intensity of point clouds as input features. For non-voxelization backbones, we set the range image size to 1024×64 for SalsaNext [11] (range-view). We extract point features and set the grid size to $(480, 360, 32)$ for PolarNet [107] (bev-view). All these networks start from randomly initialized weights. As for ASFDA and ADA settings, we have an additional warm-up stage, i.e., the network is pre-trained on the corresponding source domain for 10 epochs with the standard cross-entropy loss.

Training configuration. All methods are implemented using PyTorch [49] on a single NVIDIA Tesla A100 GPU. We utilize the SGD optimizer with an initial learning rate of 0.01. The training process spans 50 epochs and a cosine learning rate decay schedule is also applied for stable training. Both source and target data have a batch size of 16. For our voxel-centric active learning baseline, we maintain $\Delta_2 = 0.25$ for the selection process, unless otherwise specified.

B Additional experimental results

B.1 Computation cost and annotation cost

As previously discussed in the method section, striking a balance between computation cost and annotation cost is a crucial challenge in active learning. In Table A2, we provide a comprehensive breakdown of the computation cost for the SynLiDAR → KITTI task. This highlights the ability of *Annotator* to achieve an optimal equilibrium between high performance and low cost, encompassing both computation and annotation expenses. In the future, we are committed to exploring even more efficient strategies to further reduce the costs associated with both computation and annotation.

phase	total epoch	running time (hour)	mIoU
pre-train on SynLiDAR	10	2.34	22.0
active learning	50	18.04	53.7
active source-free domain adaptation	50	18.39	54.1
active domain adaptation	50	28.48	57.7

Table A2: Computation cost analysis for the SynLiDAR → KITTI task.

B.2 Training curves

In Figure A1, we present the training and validation loss curves for the SynLiDAR → POSS task under both AL and ASFDA settings. Both training loss and validation loss consistently decrease over time, indicating effective model training. Notably, the final validation loss is smaller than the training loss, suggesting a lack of overfitting. Another interesting observation is that the validation loss of the ASFDA approach is smaller than that of the AL approach, underscoring the potency of the auxiliary model in enhancing model performance.

[✉] Corresponding author.

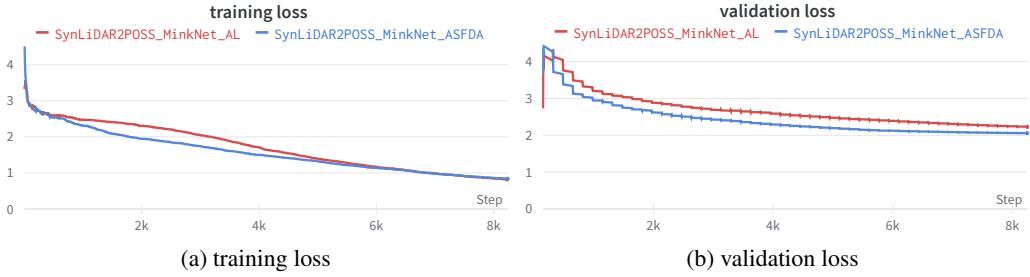


Figure A1: Training and validation loss curves on the task of SynLiDAR → POSS under AL and ASFDA settings (MinkNet [10]).

B.3 Comparison with existing active learning methods

We report mIoU results across existing AL approaches in Table A3. Notably, while LESS [36] obtains the best results with the fewest point labels, it does so by incorporating a complex pre-segmentation stage. In contrast, *Annotator* with a simpler baseline manages to deliver promising results.

Method	Budget	MinkNet	SPVCNN	Cylinder3D
ReDAL [89]	1%	47.5	48.5	-
LiDAL [28]	1%	37.8	42.6	-
LESS [36]	0.01%	-	-	61.0
Annotator	0.1%	53.7	52.8	-

Table A3: Performance comparison on the SemanticKITTI val under active learning setting.

Method	Random	Entropy	Margin	SSDR-AL	Annotator
Total budget	40.9%	46.7%	43.0%	11.7%	9.9%

Table A4: Comparing the percentage of labeled points required to achieve 90% accuracy on S3DIS dataset for different active learning methods.

B.4 Comparison with indoor semantic segmentation methods

Following SSDR-AL [66], we apply *Annotator* to indoor semantic segmentation task and conduct experiments on the S3DIS [2] dataset. In Table A4, we compare the percentage of labeled points required to achieve 90% accuracy across various methods based on RandLA-Net [26]. It's noteworthy that *Annotator* is able to annotate 1.8% fewer points than SSDR-AL in achieving the 90% performance of the fully-supervised method.

B.5 Per-class performance

Table A5 and Table A6 provide the class-wise IoU scores on two real-to-real tasks using MinkNet [10] for different algorithms and comparison results with state-of-the-art DA methods [35, 46, 61, 63, 85].

Table A7 - A10 provide the detailed class-wise IoU scores based on SPVCNN [77].

B.6 Additional qualitative results

In order to provide more qualitative insights, we show the error maps that depict the differences between our model’s predictions and the Ground-Truth labels. These error maps are showcased on the KITTI val set and models are trained on the adaptation tasks of SynLiDAR → KITTI (Figure A2) and nuScenes → KITTI (Figure A3), respectively. It is important to emphasize that *Annotator* (ADA) emerges as the top-performer, capitalizing on the advantages of pre-trained models and the presence of annotations in the source domain.

Model	vehicle	person	road	sidewalk	terrain	manmade	vegetation	mIoU
Source-Only	47.1	1.6	52.6	14.6	2.0	33.3	47.9	28.4
DA	Mix3D [46]	33.7	11.2	58.5	12.9	5.3	50.4	48.6
	CoSMix [61]	35.9	0.0	58.1	11.6	9.0	45.2	49.1
	SN [85]	21.4	0.0	60.5	15.1	6.2	31.9	45.7
	RayCast [35]	28.8	0.0	59.3	16.1	12.5	49.7	49.8
	LiDOG [63]	24.0	14.9	70.6	24.6	14.0	45.3	50.9
AL	Random	83.4	15.7	90.7	48.5	65.0	81.2	77.6
	Entropy [84]	86.2	0.0	88.1	38.1	64.8	72.8	67.8
	Margin [30]	82.6	0.0	86.3	38.0	60.4	78.9	75.0
	Annotator	88.1	44.2	91.9	56.7	67.1	75.5	69.5
ASFDA	Random	85.0	23.7	89.9	48.6	65.3	81.6	78.0
	Entropy [84]	86.3	0.0	88.3	42.8	64.3	73.9	66.3
	Margin [30]	81.7	0.0	86.1	39.0	58.1	77.0	72.4
	Annotator	88.5	49.6	92.5	58.6	68.7	77.7	71.0
ADA	Random	83.6	51.6	91.9	56.4	64.5	80.9	75.0
	Entropy [84]	86.2	59.2	90.2	53.9	66.3	80.3	75.5
	Margin [30]	88.5	46.8	91.7	58.1	65.0	78.4	71.2
	Annotator	88.8	58.6	93.1	62.6	68.4	82.4	77.3
Target-Only	89.2	73.2	95.6	71.4	75.2	87.9	85.1	82.5

Table A5: Per-class results on task of KITTI $\xrightarrow{?}$ nuScene (MinkNet [10]) using only 5 voxel budgets. DA results are reported from [63].

Model	vehicle	person	road	sidewalk	terrain	manmade	vegetation	mIoU
Source-Only	44.1	4.3	67.6	39.4	34.9	41.2	10.5	34.6
DA	Mix3D [46]	37.9	6.7	42.0	5.7	27.6	41.2	65.4
	CoSMix [61]	44.6	13.9	36.1	10.2	29.3	54.4	69.1
	SN [85]	25.7	5.5	19.6	2.2	23.5	27.7	61.1
	RayCast [35]	28.3	16.1	45.8	9.4	20.6	38.6	61.8
	LiDOG [63]	60.1	9.0	47.4	16.4	32.6	54.2	68.8
AL	Random	95.5	0.0	86.2	70.3	74.1	83.3	86.8
	Entropy [84]	96.4	0.0	84.3	68.9	75.7	82.8	87.0
	Margin [30]	94.8	35.3	83.6	68.8	65.2	80.8	83.0
	Annotator	96.9	33.9	86.8	73.4	73.3	86.5	87.0
ASFDA	Random	94.3	0.0	85.0	67.5	72.1	83.1	86.3
	Entropy [84]	95.6	0.0	83.7	66.5	73.1	79.8	85.0
	Margin [30]	94.1	24.6	82.1	66.4	64.2	78.8	82.1
	Annotator	96.6	21.9	88.5	75.7	74.1	84.2	86.1
ADA	Random	95.1	42.6	88.7	70.0	69.8	75.3	81.5
	Entropy [84]	96.0	32.2	86.2	69.1	70.8	79.6	84.0
	Margin [30]	94.5	59.6	89.4	69.4	70.2	74.5	79.6
	Annotator	95.8	66.1	88.5	74.9	75.9	84.2	86.9
Target-Only	97.6	60.6	90.7	79.3	76.5	89.1	89.2	83.3

Table A6: Per-class results on task of nuScenes $\xrightarrow{?}$ KITTI (MinkNet [10]) using only 5 voxel budgets. DA results are reported from [63].

Model	car	bike	mt.cle	truck	oth.v.	pers.	b.cls	mcsl	road	park.	sidew.	oth.g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU	
Source-Only	67.1	6.9	22.8	0.5	5.9	30.1	56.9	4.2	18.3	6.3	31.1	0.3	30.8	11.8	63.9	29.9	42.9	25.5	4.1	24.2	
AL	Random	92.2	0.0	10.4	25.8	18.5	23.6	0.0	0.8	85.2	23.2	69.4	1.2	83.2	44.5	86.2	56.8	73.1	54.8	27.8	40.9
	Entropy [84]	94.4	6.1	56.2	67.9	38.6	57.2	72.4	0.0	80.3	22.2	64.1	3.2	83.6	44.5	86.4	58.7	72.7	58.7	35.0	52.7
	Margin [30]	90.4	0.0	34.2	53.9	31.2	45.8	68.1	0.0	80.5	19.4	66.8	0.2	79.3	46.3	82.3	56.8	64.4	51.5	23.2	47.1
	Annotator	93.9	1.6	49.9	48.9	36.4	50.2	71.8	0.1	86.2	24.6	72.3	1.4	86.6	53.1	86.4	63.1	72.5	61.7	42.8	52.8
ASFDA	Random	92.8	0.0	0.0	33.3	24.6	1.1	0.0	0.0	90.0	35.7	76.1	0.0	86.9	52.5	87.1	59.3	75.4	57.9	22.9	41.7
	Entropy [84]	95.1	1.0	59.8	62.3	44.0	55.4	77.3	1.0	77.2	18.4	60.3	0.1	82.6	44.5	84.9	59.5	70.4	60.0	36.1	52.1
	Margin [30]	90.7	1.6	45.6	63.2	31.4	48.6	63.8	0.0	85.0	27.6	70.5	0.0	81.5	48.1	84.1	57.4	69.9	54.8	23.2	49.9
	Annotator	94.5	10.6	47.6	71.9	43.5	53.9	67.1	0.0	86.9	24.6	73.4	1.8	85.8	51.1	85.6	64.1	71.6	60.8	41.7	54.6
ADA	Random	92.3	10.9	40.7	42.3	28.8	50.8	71.9	0.0	88.1	27.5	73.8	2.5	84.3	49.6	83.6	59.6	69.9	54.2	38.6	51.0
	Entropy [84]	94.2	16.1	53.3	60.1	39.2	61.4	79.8	2.2	82.4	18.6	65.4	1.4	81.7	46.1	83.8	61.0	65.2	55.1	35.3	52.8
	Margin [30]	92.1	1.0	56.9	47.5	32.1	50.7	82.8	0.0	84.5	25.5	68.5	0.2	78.3	54.0	81.7	57.8	64.8	52.2	31.8	50.7
	Annotator	94.7	14.8	56.7	56.8	45.3	60.4	79.0	1.3	87.3	28.6	73.0	1.8	85.4	54.3	83.9	65.2	66.5	60.0	40.9	55.6
Target-Only	96.7	25.6	73.6	81.0	61.5	73.6	90.9	0.2	93.0	46.1	79.9	0.1	89.9	58.7	86.8	67.3	71.5	65.1	48.8	63.7	

Table A7: Per-class results on task of SynLiDAR $\xrightarrow{19}$ KITTI (SPVCNN [77]) with 5 voxel budgets.

Model	car	bike	pers.	rider	grou.	buil.	fence	plants	trunk	pole	traf.	garb.	cone.	mIoU	
Source-Only	51.7	3.1	46.7	46.0	80.0	57.7	37.2	66.4	29.2	28.8	1.1	21.3	12.3	37.0	
AL	Random	35.3	43.9	37.7	9.2	77.0	67.8	42.5	70.7	27.8	28.8	21.5	0.0	0.0	35.5
	Entropy [84]	24.1	35.9	35.0	22.6	78.4	61.2	42.1	71.9	14.4	22.1	15.2	16.0	18.6	35.2
	Margin [30]	33.5	41.3	55.1	47.0	78.4	54.4	36.6	67.0	41.7	27.5	23.3	20.1	31.4	42.9
	Annotator	31.6	44.9	56.4	46.8	78.7	65.8	50.4	73.2	32.6	26.9	36.2	16.1	23.7	44.9
ASFDA	Random	38.3	48.1	44.5	16.8	76.7	68.9	46.7	71.1	20.8	30.2	29.6	0.0	0.0	37.8
	Entropy [84]	34.5	42.7	54.4	39.4	77.6	66.6	39.7	71.3	19.0	27.5	31.5	2.3	19.9	40.5
	Margin [30]	32.7	44.1	57.6	52.2	77.9	59.3	42.8	70.3	41.6	33.4	31.4	22.9	16.4	44.8
	Annotator	42.8	49.3	58.7	52.9	76.2	67.4	52.7	71.4	26.9	31.5	33.7	16.0	37.6	47.5
ADA	Random	51.8	44.8	55.0	47.1	75.4	69.6	51.6	71.2	32.3	27.3	20.2	2.1	1.3	42.3
	Entropy [84]	54.5	42.7	65.3	57.6	79.4	60.8	55.0	70.2	29.2	28.8	18.7	5.0	40.5	46.8
	Margin [30]	36.8	30.0	65.5	58.1	81.4	65.1	44.2	70.7	37.4	31.5	25.4	35.5	30.6	47.1
	Annotator	64.2	40.8	62.1	55.7	77.8	67.5	57.2	70.8	31.2	35.3	28.5	24.3	46.2	50.9
Target-Only	46.5	57.3	69.6	53.9	79.8	79.6	60.5	80.9	37.9	32.3	32.4	16.1	17.6	51.9	

Table A8: Per-class results on task of SynLiDAR $\xrightarrow{13}$ POSS (SPVCNN [77]) with 5 voxel budgets.

Model	vehicle	person	road	sidewalk	terrain	manmade	vegetation	mIoU
Source-Only	34.4	0.2	29.7	8.5	6.5	25.9	44.0	21.3
AL	Random	82.8	31.9	87.6	41.2	59.3	77.8	74.8
	Entropy [84]	77.9	36.3	85.6	28.9	58.3	73.6	68.5
	Margin [30]	77.9	20.4	84.3	34.6	48.8	71.7	67.3
	Annotator	85.9	47.0	92.3	57.6	65.8	77.9	71.4
ASFDA	Random	83.8	34.9	88.9	45.4	59.5	79.3	76.5
	Entropy [84]	87.6	34.4	87.4	36.6	60.1	80.3	75.7
	Margin [30]	77.6	25.1	85.4	40.6	54.8	70.9	67.7
	Annotator	88.4	46.8	92.8	58.7	66.4	78.1	72.1
ADA	Random	84.1	27.1	91.0	53.0	55.1	72.1	67.9
	Entropy [84]	89.7	44.2	89.8	51.3	53.7	71.7	63.7
	Margin [30]	85.4	17.1	91.8	55.0	55.3	72.1	65.8
	Annotator	89.5	50.2	92.1	57.5	66.3	78.6	72.2
Target-Only	93.1	71.7	94.6	66.3	72.1	87.0	84.0	81.3

Table A9: Per-class results on task of KITTI $\xrightarrow{7}$ nuScene (SPVCNN [77]) with 5 voxel budgets.

Model	vehicle	person	road	sidewalk	terrain	manmade	vegetation	mIoU
Source-Only	65.7	44.5	56.2	32.6	30.5	53.2	47.0	47.1
AL	Random	94.2	0.0	84.7	68.1	73.9	84.4	87.6
	Entropy [84]	94.2	13.0	79.0	60.4	73.1	80.8	85.8
	Margin [30]	93.6	29.4	84.0	69.9	64.4	81.4	83.5
	Annotator	96.7	48.2	87.9	74.6	75.8	85.8	87.8
ASFDA	Random	94.3	3.5	81.6	60.8	68.8	81.9	85.7
	Entropy [84]	95.1	11.0	77.2	55.6	69.5	78.8	84.5
	Margin [30]	92.5	32.9	85.0	70.5	65.2	81.6	83.1
	Annotator	96.7	55.4	87.7	75.3	75.6	85.5	87.5
ADA	Random	94.1	39.5	88.9	73.0	71.3	80.0	83.4
	Entropy [84]	90.2	63.6	84.2	70.9	65.9	66.5	72.6
	Margin [30]	92.7	58.6	88.2	71.0	69.3	71.4	75.3
	Annotator	95.0	55.0	86.5	69.0	74.9	82.8	86.0
Target-Only	98.0	71.7	91.0	79.9	74.6	90.5	89.0	85.0

Table A10: Per-class results on task of nuScenes $\xrightarrow{?}$ KITTI (SPVCNN [77]) with 5 voxel budgets.

C Public Resources Used

We acknowledge the use of the following public resources, during the course of this work:

- SynLiDAR¹ MIT License
- SemanticKITTI² CC BY-NC-SA 4.0
- SemanticKITTI-API³ MIT License
- SemanticPOSS⁴ CC BY-NC-SA 3.0
- nuScenes⁵ CC BY-NC-SA 4.0
- nuScenes-devkit⁶ Apache License 2.0
- Minkowski Engine⁷ MIT License
- SPVNas⁸ MIT License
- PCSeg⁹ Apache License 2.0
- LaserMix¹⁰ CC BY-NC-SA 4.0
- GIPSO¹¹ GNU General Public License v3.0
- SalsaNet¹² MIT License
- PolarNet¹³ BSD 3-Clause License
- RIPU¹⁴ MIT License

¹<https://github.com/xiaoaroruan/SynLiDAR>.

²<http://semantic-kitti.org>.

³<https://github.com/PRBonn/semantic-kitti-api>.

⁴<http://www.poss.pku.edu.cn/semanticposs.html>.

⁵<https://www.nuscenes.org/nuscenes>.

⁶<https://github.com/nutonomy/nuscenes-devkit>.

⁷<https://github.com/NVIDIA/MinkowskiEngine>.

⁸<https://github.com/mit-han-lab/spvna>.

⁹<https://github.com/PJLab-ADG/PCSeg>.

¹⁰<https://github.com/ldkong1205/LaserMix>.

¹¹[https://github.com/salториクリスチяно/gipso-sfouda](https://github.com/salториクリスチアノ/gipso-sfouda).

¹²<https://gitlab.com/aksoyeren/salsanet>.

¹³<https://github.com/edwardzhou130/PolarSeg>.

¹⁴<https://github.com/BIT-DA/RIPU>.

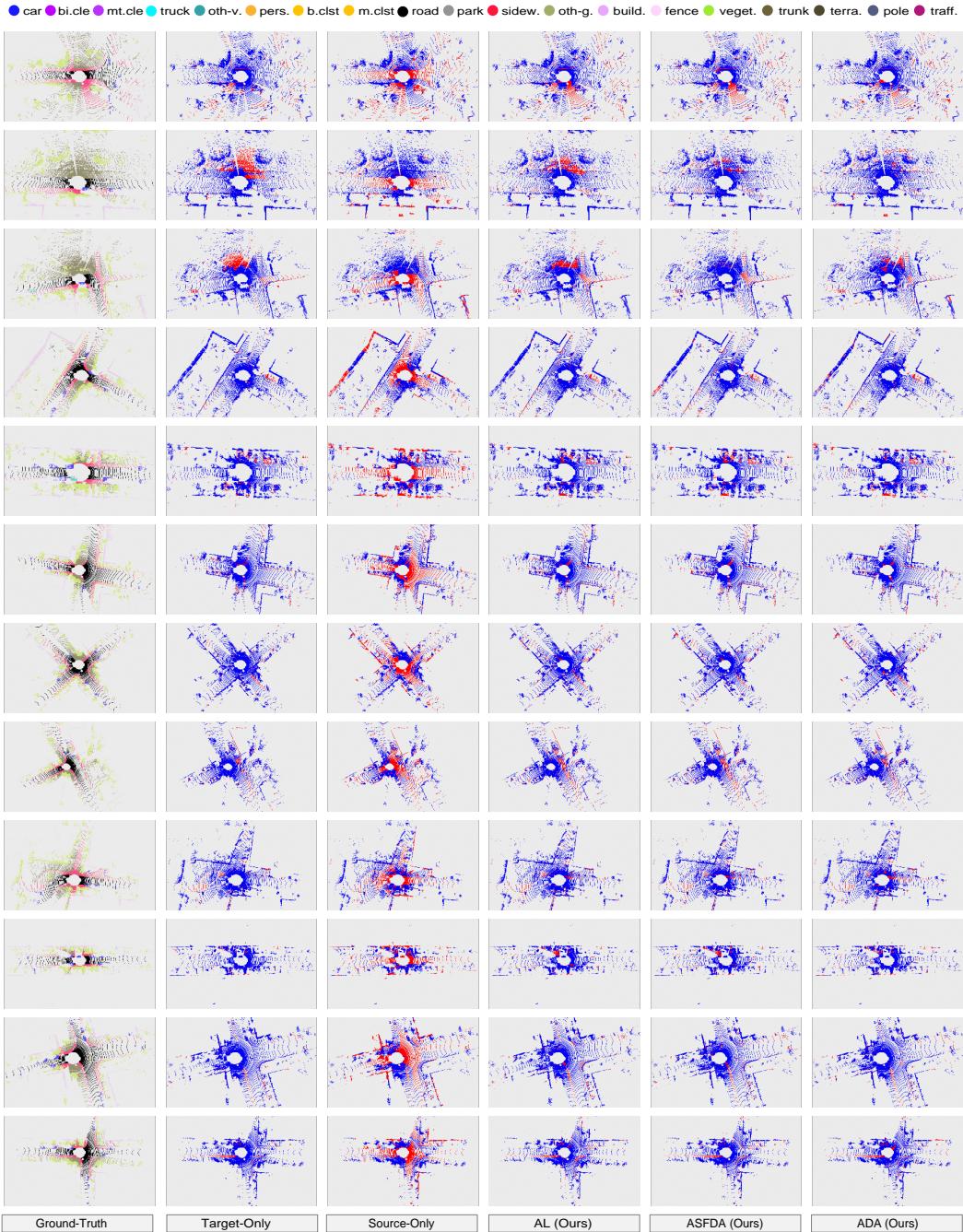


Figure A2: Visualization of error maps for the task SynLiDAR $\xrightarrow{19}$ KITTI (MinkNet [10]). From left to right: Ground-Truth, Target-Only, Source-Only, our Annotator under AL, ASFDA, and ADA are shown one by one. The **correct** and **incorrect** predictions are painted in **blue** and **red** to highlight the differences. Best viewed in color.

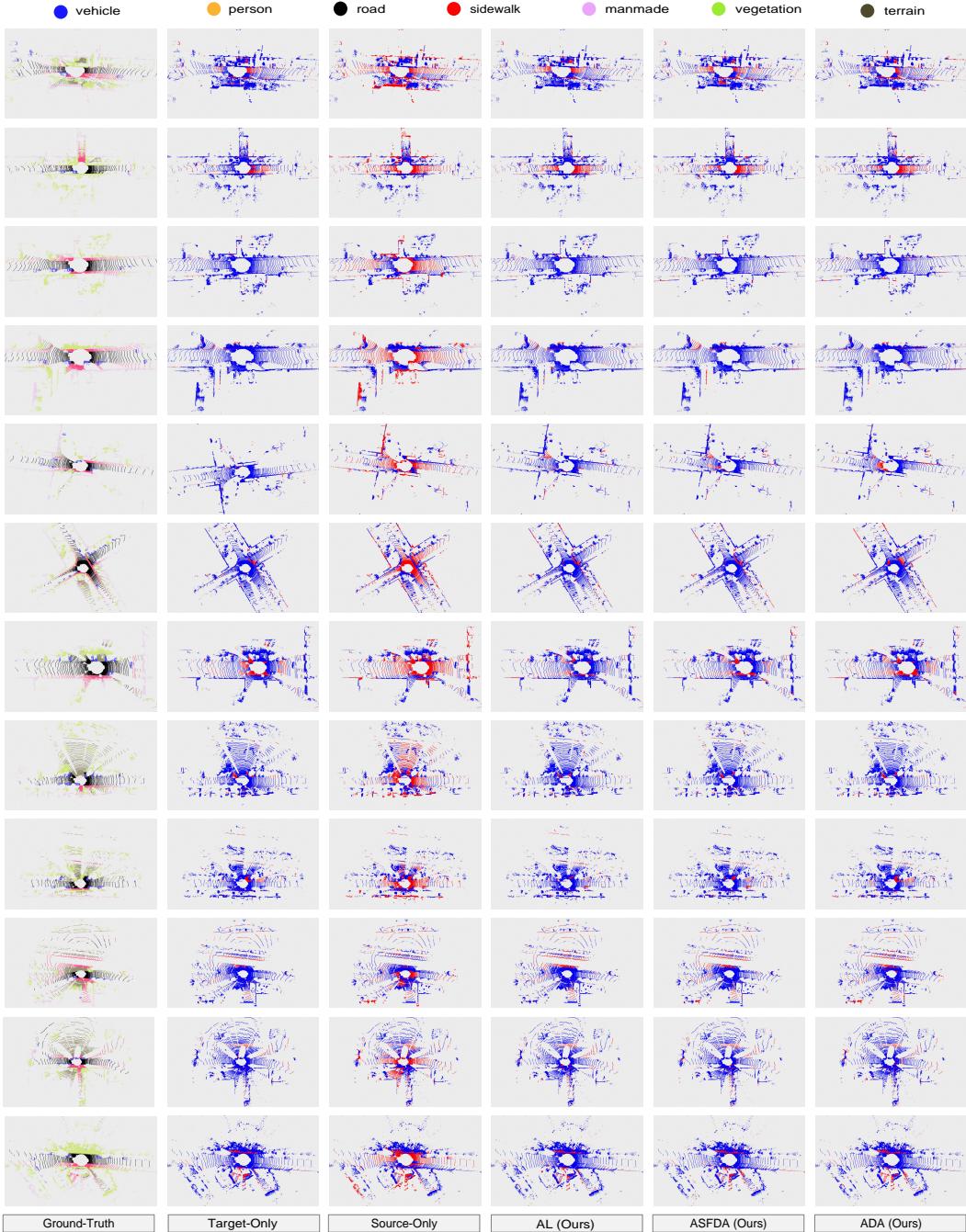


Figure A3: Visualization of error maps for the task nuScenes $\xrightarrow{7}$ KITTI (MinkNet [10]). From left to right: Ground-Truth, Target-Only, Source-Only, our Annotator under AL, ASFDA, and ADA are shown one by one. The **correct** and **incorrect** predictions are painted in **blue** and **red** to highlight the differences. Best viewed in color.