Multi-Echo LiDAR for 3D Object Detection

Yunze Man[1], Xinshuo Weng[1], Prasanna Kumar Sivakumar[2], Matthew O'Toole[1], Kris Kitani[1]
[1]Carnegie Mellon University, [2]DENSO
{yman, xinshuow, mpotoole, kkitani}@cs.cmu.edu, prasanna.kumar.sivakumar@na.denso.com

Abstract

LiDAR sensors can be used to obtain a wide range of measurement signals other than a simple 3D point cloud, and those signals can be leveraged to improve perception tasks like 3D object detection. A single laser pulse can be partially reflected by multiple objects along its path, resulting in multiple measurements called echoes. Multi-echo measurement can provide information about object contours and semi-transparent surfaces which can be used to better identify and locate objects. LiDAR can also measure surface reflectance (intensity of laser pulse return), as well as ambient light of the scene (sunlight reflected by objects). These signals are already available in commercial LiDAR devices but have not been used in most LiDAR-based detection models. We present a 3D object detection model which leverages the full spectrum of measurement signals provided by LiDAR. First, we propose a multi-signal fusion (MSF) module to combine (1) the reflectance and ambient features extracted with a 2D CNN, and (2) point cloud features extracted using a 3D graph neural network (GNN). Second, we propose a multi-echo aggregation (MEA) module to combine the information encoded in different sets of echo points. Compared with traditional single echo point cloud methods, our proposed Multi-Signal LiDAR Detector (MSLiD) extracts richer context information from a wider range of sensing measurements and achieves more accurate 3D object detection. Experiments show that by incorporating the multi-modality of LiDAR, our method outperforms the state-of-the-art by up to relatively 9.1%.

1. Introduction

LiDAR is a powerful sensor that has the ability to capture a wide range of measurements for perception tasks including object detection. The most commonly used LiDAR measurement type is a set of 3D points (a point cloud) and their reflectance values, which provides accurate 3D shape information of objects in the scene. State-of-the-art object detection methods have made great breakthroughs by leveraging 3D point cloud data. However, despite such success, there are several types of LiDAR measurements that

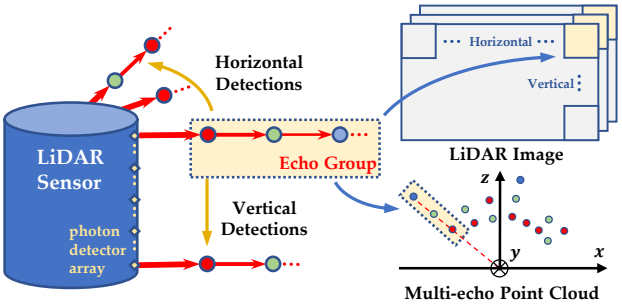

Figure 1. **Illustration of multi-signal measurements from LiDAR sensor.** Each photon detector on the sensor collects a group of signals and forms an "Echo Group", which is converted into a 2D LiDAR image and a multi-echo point cloud representation.

are largely ignored in modern-day LiDAR perception algorithms. In the following, we describe three unique features of the LiDAR sensor, which are available in standard LiDAR sensors, but surprisingly have rarely been used in published LiDAR-based object detection algorithms. We show that by leveraging these features, one can greatly improve 3D object detection performance.

The first important feature of LiDAR is its ability to obtain multiple return signals with a single laser pulse, called echoes. LiDAR is a time-of-flight measurement system which measures the time it takes for a laser pulse to hit an object and return to the sensor. More specifically, the laser emits a short pulse, and a photodetector timestamps the arrival of photons reflected back by object surfaces. It is possible for a photodetector to acquire multiple return signals (echoes) if the laser is partially reflected by multiple objects along its path of propagation. We call the multiple returned signals generated from the same laser beam an 'echo group.' Points in the same echo group lie on one line in 3D space, and they are typically ordered according to their signal strength. In addition to the direct benefit of increasing the number of points available, multiple echoes also imply that high-order echo points are likely on the contour of an object (objects obstruct only a part of the laser) or on a semi-transparent surface (a portion of the laser propagates through the surface). In either case, we hypothesize that echoes encode meaningful features that can help locate

or classify an object.

The second important feature of LiDAR is the ability to capture ambient scene illumination. The photodetector of the LiDAR continuously captures infrared (IR) light and therefore is capturing IR images of the scene (typically reflected sunlight) between laser pulses. Although this information is typically ignored in most LiDAR-based perception algorithms, a LiDAR can be used to capture an image of the scene using the IR spectrum. Ambient measurements can be processed as a 2D image and can be used to extract texture information about the objects in the scene.

The third important feature of LiDAR is the ability to capture surface reflectance. LiDAR captures laser signal returns, so each point will have a corresponding reflectance value which measures the strength of the detected laser pulse. Reflectance also encodes material properties of objects useful for detection and classification. Unlike the ambient signal, different points inside the same echo group will have different reflectance values, resulting in multiple reflectance values which we call multi-echo reflectance.

We propose a multi-signal LiDAR-based 3D object detector (MSLiD). First, to better leverage the dense texture and surface properties encoded in the ambient and reflectance signals, we re-organize them as a dense 2D representation, called the 'LiDAR Image.' Then, in order to combine the dense 2D image with the sparse 3D point cloud, we propose a multi-signal fusion (MSF) module which incorporates a 2D CNN branch and a 3D GNN branch. The MSF module aims at fusing 2D visual information with 3D positional information by sending pixels and class features from the 2D branch to point-wise features learned in the 3D branch. Furthermore, In order to extract and combine information encoded in different echo groups, we propose a multi-echo aggregation (MEA) module. To resolve the imbalance between the number of points in different echos, The MEA module reassigns multi-echo points into two sets – 'penetrable' and 'impenetrable' sets according to whether the object reflects partial laser signal. Aggregating the features learned from two new sets of points provides richer context information of objects and leads to better location estimation. By cascading the MSF and MEA modules, the proposed system combines dense visual information from ambient/reflectance and sparse geometric information from the point cloud, while also extract richer context features by aggregating multiple echoes. By leveraging multi-signal LiDAR measurements other than a single point cloud, MSLiD learns a more discriminative object representation which leads to accurate object localization and classification.

We collect one real-world and one synthetic dataset with multiple LiDAR measurements, including ambient signal, multi-echo point cloud, and reflectance signals. Experiments on two datasets demonstrate that our method outperforms state-of-the-art single-echo methods by up to 9.1%.

Overall, our contributions can be summarized as follows:

1. MSLiD is the first to propose a 3D detection framework that properly leverages ambient illumination, multiple echos of point clouds and reflectance signals for LiDAR sensor. Our method shows improvement over prior methods using single-echo point cloud with reflectance intensity.
2. We propose a multi-signal fusion module to effectively combine dense visual information from ambient and reflectance signals with sparse 3D positional information from point cloud.
3. We propose a multi-echo aggregation module to form a richer context representation of objects from multiple groups of echoes, resulting in more accurate object localization and classification.

2. Related Work

3D Object Detection with Grid-based Methods. Many existing works convert the point cloud into a regular grid space representation in order to tackle the inherent sparsity and irregular format of the point cloud. [2, 10, 13, 28] project 3D point cloud into 2D birds eye view image to extract feature with mature 2D CNN. For real-time detection, [29, 11] explore more efficient framework for birds eye view transformation. Other work focuses on 3D voxel representation. [33] voxelizes the point cloud and uses 3D CNNs to extract features. Sparse convolution [5] is introduced in [27] for more efficient voxel processing and feature extraction. Also, [26] explores the non-regular shape of 3D voxels while [22] proposes to combine point feature learning with voxelization, leading to higher detection performance. Grid-based methods are generally efficient for proposal generation, but they suffer from information loss during the projection or voxelization process. In contrast, our method does not have the problem of loss of information as we do not voxelize or project point clouds.

3D Object Detection with Point-based Methods. F-PointNet [18] first proposes to use frustum proposals from 2D object detection and regresses the final bounding box directly from point features extracted by PointNet [19, 20]. [23] instead proposes to generate 3D candidate proposals directly from the point cloud in a bottom-up manner, and the following [31] proposes to learn a dense voxel representation of each candidate proposal for more efficient bounding box regression. [30] further reduces the inference time by removing the refinement stage and regressing directly from 3D keypoints. Despite using different point cloud encoding frameworks, these works are working with single-echo point cloud and reflectance value. Compared to prior work, our method leverages multi-echo point clouds and ambient images to learn a richer representation for both point-level features and proposal-level features.

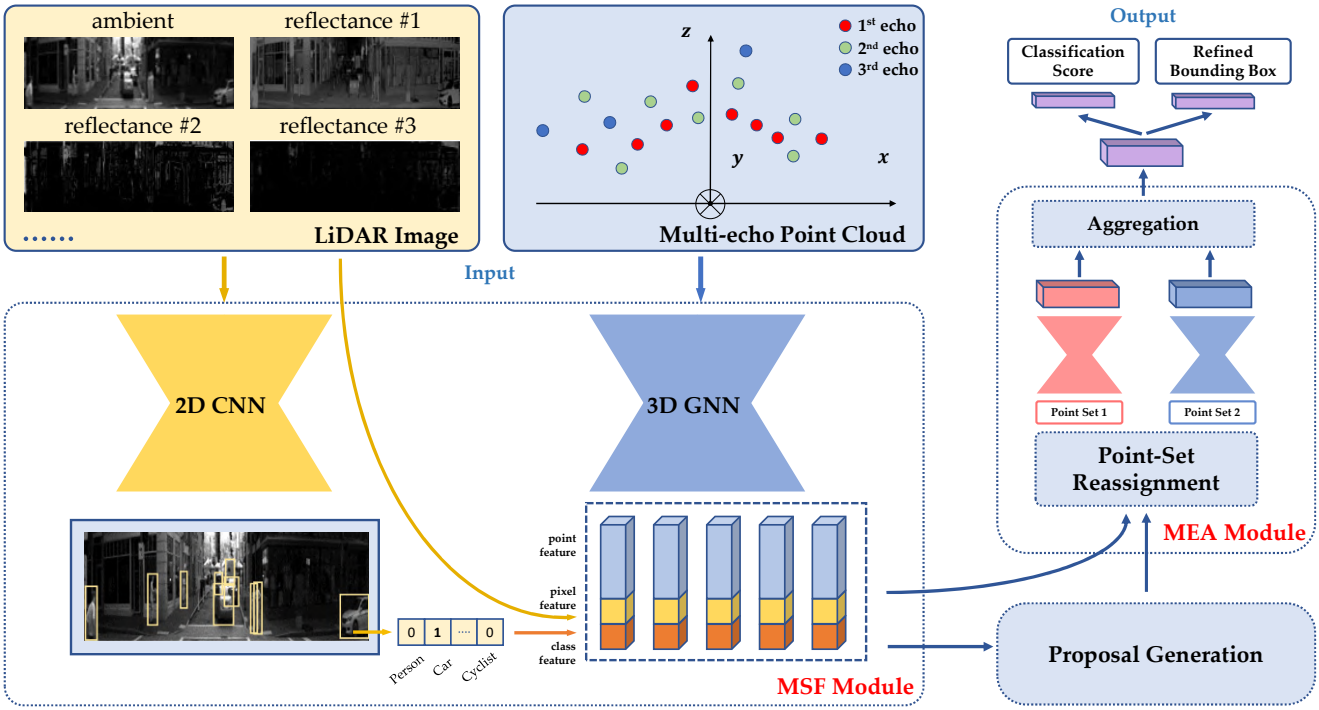

Figure 2. Illustration of our proposed framework. Our method takes multi-echo point cloud and LiDAR image as input. The MSF module learns separate features for 2D and 3D representation, and fuses 2D pixel feature with 3D point feature. The MEA module first performs point set reassignment to group together points with similar properties. Discriminative RoI feature for each proposal candidate is then learned by aggregating features learned from different set of points. The learned feature is then used for confidence estimation and bounding box regression.

3D Object Detection with Multi-modal Fusion Methods. Exploring effective ways to fuse signals from multiple modalities is still an open question in 3D object detection. [10, 2, 12] propose to project point cloud to BEV space, and then fuse 2D RGB features with a BEV feature to generate proposals and regress bounding boxes. [18] does not apply feature fusion, but instead uses 2D detection bounding boxes to guide 3D proposal generation. [34, 32, 8] explores deep feature fusion between LiDAR and RGB sensors. [25] proposes to augment point cloud with image semantic segmentation results. [17] proposes to fuse geometry cue, semantic cue and texture cue from 2D features with 3D point features and obtains promising performance for indoor 3D object detection. [16] proposed a post-detection fusion mechanism to combine the candidate boxes from RGB and LiDAR inputs. In our work, we propose to work with a wider range of modalities coming from the LiDAR sensor. We are the first to explore a proper way of combining multi-echo points with ambient/reflectance signal information.

3. Approach

We design a 3D object detection solution suited for multi-signal measurements provided by the LiDAR sensor, including multi-echo point cloud, as well as ambient and reflectance signals. Our Multi-Signal LiDAR-based Detector, called MSLiD, achieves fusion between multiple LiDAR signals with two modules, *i.e.* Multi-Signal Fusion (MSF) module where visual signals are combined with geometric signals, and Multi-Echo Aggregation (MEA) module where points from different echo groups are fused together. In this section, we first describe how multiple types of signals are processed, encoded and fused in the MSF module for point-wise feature extraction and proposal generation (Sec 3.1). Then, we describe the MEA module where multiple point clouds are reassigned into different point sets, and features of these point sets are extracted and aggregated to form the proper proposal RoI feature for 3D bounding box regression (Sec 3.2). Note that we assume the LiDAR sensor generates k-echo point clouds. The overall pipeline of our method is shown in Fig 2.

3.1. Multi-Signal Fusion for Proposal Generation

Learning features of multiple modalities by multiple streams [12, 17, 25] is proved to be effective in feature fusion. Aiming at fully making use of the complementary nature between different signals, we utilize a two-stream feature extraction and blending framework.

Data Encoding. In order to leverage mature 2D detection

models to extract visual cues from multiple signals, we first convert ambient and reflectance signals into 2D image representation called "LiDAR image". Unlike previous methods where point cloud is projected into image space with the calibration matrix, we re-organize the LiDAR multi-modal measurement into a 2D image according to the alignment of LiDAR detector array (*i.e.*, range view [1]), as shown in Fig 1. Specifically, each column of pixels are signals captured by the vertically aligned LiDAR detectors and each row of pixels are signals captured by the same detector in different horizontal directions. The converted LiDAR image has resolution $[h, w, n]$ where h corresponds to the number of vertically aligned detectors and w corresponds to horizontally aligned ones. n is the number of modalities encoded in each 'pixel', which in our case includes one ambient value and k reflectance values.

For the image stream, the converted LiDAR images are passed into a 2D detector to generate 2D bounding boxes. We adopt an FPN-based model [14] as our 2D detector, where the backbone weights are pre-trained on ImageNet classification and the model is pre-trained on COCO object detection. We then fine-tune it on our dataset using LiDAR image as inputs to detect 2D bounding boxes. For the point cloud stream, we utilize PointNet++ [20] as our backbone network to learn discriminative point-wise features from raw 3D point cloud. To leverage point clouds from different echos and increase point density, we group k-echo points together as a whole point cloud and extract features from it.

Multi-Signal Fusion (MSF). In order to generate rich point-wise features for proposal generation, we present a 2D-to-3D feature blending method that augments 2D pixel-wise semantic information to 3D point-wise geometric features. [17] presents an effective indoor 3D object detection framework to extract three different cues (features) from 2D detection, where geometric cue infers 3D proposal center from 2D bounding box center, semantics cue infers the object type of the proposal, and texture cue (RGB value) encodes the texture information of the surface. However, the geometric cue relies on a very strict assumption where the sensor origin, 2D center and 3D center lies on the same line. This assumption approximately holds for indoor scenarios with controllable depth range and object shape. But for the autonomous driving scenario, the error caused by this approximation is often unbearable.

In contrast, we propose an MSF module which appends 2D semantic features from a pixel to its corresponding point – class probability vector (class vector) and dense LiDAR measurements (pixel vector). Specifically, for each pixel of the LiDAR image, we form a one-hot vector to represent which class it belongs to. If the pixel is not inside any bounding box, the vector is set to be all zero, and if the pixel is inside multiple bounding boxes, the corresponding class entries are all set to one. We believe that the regional prediction vector helps to address the ambiguity of object class in the sparse 3D point cloud. On the other hand, the dense LiDAR measurements include the ambient value and reflectance value the point corresponds to. Each point has its unique reflectance value, while multiple points in the same echo group share the same ambient value. The dense LiDAR measurements include lower-level semantic features that 3D point cloud does not possess, including object texture and surface reflectivity. By fusing 2D semantic features with 3D point features, our method can encode multi-modal information which helps better locate and identify objects.

Given fused features, we apply a bottom-up proposal generation strategy [23] to generate 3D candidates for the bounding box refinement stage. Specifically, we learn a binary foreground/background segmentation using the fused point-wise features, where foreground points are defined as points inside any ground truth 3D bounding boxes. Afterward, we generate anchor boxes around foreground points and regress the residual of box parameters [23, 18].

3.2. Multi-Echo Aggregation for Box Refinement

Given several proposal candidates, the bounding box refinement network aims to estimate the proposal confidence and predict the residuals of bounding box parameters (i.e., center, size and orientation) based on a representative RoI feature. Following [23, 31, 24], we first perform a canonical transformation to each point by subtracting their 3D position with the proposal center (X, Y, Z) values and rotating them to the proposal predicted orientation. This makes the model robust under geometrical transformation and thus can learn better local features. Then we want to learn a representative RoI feature of each proposal for confidence estimation and bounding box regression. Our main motivation is to effectively extract discriminative features from multi-echo point clouds. To this end, we propose point set reassignment and multi-echo aggregation modules.

Point Set Reassignment. The most straightforward way to leverage multi-echo points in RoI feature pooling is to learn separate features for each group of echo points. However, this naive idea leads to poor performance for two reasons. The first reason is that the numbers of points in different echo point clouds are extremely biased. If the echoes are ordered by signal reflectance (where low order echoes have higher intensity), then higher echo point clouds will have far fewer points than low echo point clouds. This is because points with lower signal intensity are less likely to be detected by the sensor. Second, too many uncontrollable factors can affect the order of echo points. Since echoes are ordered according to signal reflectance, various unpredictable factors decide with echo group a point belongs to, including but not limited to atmosphere humidity, incident angle, surface roughness and other optical properties of the

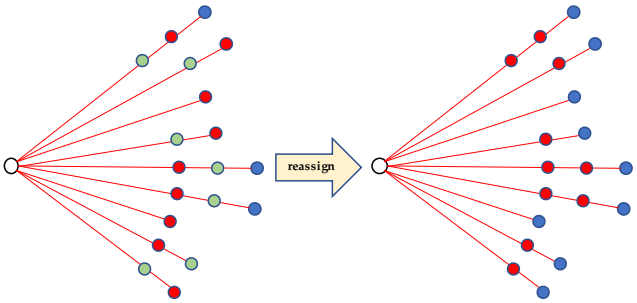

Figure 3. Illustration of point reassignment strategy on a three-echo point cloud, where white dots represent sensor origins. **Left**: original point sets – [1st, 2nd, 3rd] echo points. **Right**: After reassignment – [penetrable, impenetrable] points (*best view in color*).

object surface. It turns out the network is not able to extract useful features to refine the bounding box when a huge pack of factors is entangled in each echo point cloud.

As a result, we present a grouping method to reassign multi-echo points into two new sets. As shown in Fig 3, the farthest point to the sensor of each 'echo group' is assigned to one set and the rest points are assigned to the other set. We call the two new sets 'penetrable' and 'impenetrable' point set, in the sense that if an echo is not the furthest point in an 'echo group', it means the laser can 'penetrate' this object and be reflected by objects further away. If a point is assigned to the penetrable set, it is likely that the point is on the contour of an object (partial signal keeps propagating forward), or reflected by a semi-transparent surface (partial signal travels through the surface). It is clear that contour information helps better locate the object, and semi-transparent information encodes the existence of certain parts of an object such as the window of a car, both are useful for object box refinement.

Multi-Echo Aggregation. Given two new sets of points, we then aim to learn a regional RoI feature for accurate bounding box refinement. Since points in two sets encode different object information, we learn two separate features for each set with MLP followed by a point-wise pooling [19]. Then we form a joint feature of the RoI region by aggregating two feature vectors together. We explore multiple aggregation schemes and choose concatenation as the final method. The refinement network finally adopts a 2-layer MLP which diverges into 2 branches to perform confidence estimation and proposal regression.

3.3. Loss Function

Our proposed method is trained with a multi-task loss, including proposal generation loss L_{pg} and bounding box refinement loss L_{refine}:

$$L_{\text{overall}} = L_{\text{pg}} + L_{\text{refine}}. \tag{1}$$

Following [23, 31], the proposal generation loss L_{pg} consists of a point cloud binary segmentation loss and the proposal regression loss:

$$\mathcal{L}_{\text{pg}} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{focal}}, \tag{2}$$

where $\mathcal{L}_{\text{focal}}$ is the focal loss used to learn point cloud foreground segmentation as described in [15, 23]. Using a bottom-up bin-based proposal generation module, the proposal regression loss $\mathcal{L}_{\text{reg}}$ is composed of a bin classification loss L_{bin} and a size residual loss L_{res}. Given the proposal parameter $(x, y, z, h, w, l, \theta)$ where (x, y, z) is the object center, (h, w, l) is the object size and θ is the orientation, the loss terms are formulated as:

$$\begin{aligned}
\mathcal{L}^p_{\text{bin}} &= \sum_{u\in\{x,z,\theta\}} (\mathcal{L}_{\text{cls}}(\widehat{\text{bin}}^p_u, \text{bin}^p_u) + \mathcal{L}_{\text{L1}}(\widehat{\text{res}}^p_u, \text{res}^p_u)), \\
\mathcal{L}^p_{\text{res}} &= \sum_{u\in\{y,h,w,l\}} \mathcal{L}_{\text{L1}}(\widehat{\text{res}}^p_v, \text{res}^p_v), \\
\mathcal{L}_{\text{reg}} &= \frac{1}{N_{\text{pos}}} \sum_{p\in\text{pos}} \left(\mathcal{L}^p_{\text{bin}} + \mathcal{L}^p_{\text{res}}\right),
\end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{cls}}$ is the classification cross-entropy loss, and $\mathcal{L}_{\text{L1}}$ is the smooth L1 regression loss. $\widehat{\text{bin}}^p$ and $\widehat{\text{res}}^p$ are predicted bin selection and parameter residual for point p, while bin^p and res^p are the ground truth ones. N_{pos} is the number of total foreground points, so the proposal regression loss is the average sum of bin loss and residual loss for all foreground points. Also, the bounding box refinement loss L_{refine} is composed of a classification loss for confidence estimation and a regression loss for similar to the previous stage.

$$\begin{aligned}
\mathcal{L}_{\text{refine}} =& \frac{1}{N_{\text{a}}} \sum_{i}^{N_{\text{a}}} \mathcal{L}_{\text{cls}}(\text{score}_i, \text{label}_i) \\
&+ \frac{1}{N_{\text{p}}} \sum_{i}^{N_{\text{p}}} (\tilde{\mathcal{L}}^i_{\text{bin}} + \tilde{\mathcal{L}}^i_{\text{res}}),
\end{aligned} \tag{4}$$

where N_{a} is the number of anchor boxes, and N_{p} is the number of positive proposals for regression, score_i and label_i are predicted and ground truth confidence label. $\tilde{\mathcal{L}}^i_{\text{bin}}$ and $\tilde{\mathcal{L}}^i_{\text{res}}$ are bin and residual loss similar to previous stage, except both the predicted and ground truth bounding box parameters are transformed into the canonical coordinate.

4. Experiments

In this section we introduce the dataset (Sec 4.1) we used to train and test our approach. We also introduce the implementation details including network architecture and training parameters (Sec 4.2) used in our experiment. Then we compare our results with other state-of-the-art 3D detection

Table 1. **Performance comparison of 3D object detection with state-of-the-art methods on the real-world dataset**. The evaluation metric is Average Precision (AP) with different IoU thresholds.

Method		Car - IoU = 0.7			Car - IoU = 0.5			Person - IoU = 0.5			Person - IoU = 0.25		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND [27]	1st echo	67.9	37.1	27.3	79.9	57.0	56.2	42.1	25.2	19.8	54.8	35.5	23.9
	full echo	75.0	42.9	30.6	86.8	65.5	65.1	47.4	29.9	20.3	58.5	39.2	25.6
PointRCNN [23]	1st echo	66.9	37.8	28.1	80.2	57.7	52.1	45.2	28.9	20.0	57.0	38.7	25.2
	full echo	73.6	41.9	28.9	85.0	65.6	62.7	51.6	31.4	20.2	61.2	40.7	25.9
3DSSD [30]	1st echo	64.1	36.7	27.0	77.4	55.9	52.4	45.7	28.8	20.2	57.5	38.5	23.8
	full echo	72.4	40.6	28.1	83.9	65.1	63.9	51.9	30.7	19.9	60.8	40.5	26.2
SASSD [6]	1st echo	68.8	37.9	28.6	81.2	58.3	56.8	42.5	26.7	18.2	54.7	36.6	23.2
	full echo	76.1	43.2	29.8	87.2	66.0	63.7	46.9	28.5	19.4	58.1	38.6	23.7
PV-RCNN [22]	1st echo	69.1	38.3	28.4	81.7	59.1	57.4	44.9	28.2	19.7	56.3	38.2	24.9
	full echo	76.9	44.1	**31.2**	88.1	67.2	65.3	52.7	30.9	20.8	62.0	41.2	26.2
MSLiD		**79.5**	**45.3**	30.7	**89.7**	**68.1**	**65.3**	**57.5**	**34.2**	**21.5**	**66.5**	**43.2**	**27.7**
Improvement		*+2.6*	*+1.2*	*-0.5*	*+1.6*	*+0.9*	*+0.0*	*+4.8*	*+2.8*	*+0.7*	*+4.5*	*+2.0*	*+1.5*

methods on two datasets (Sec 4.3, Sec 4.5) and conduct extensive ablation studies (Sec 4.4) to investigate each component of our approach and validate our design choices.

4.1. Dataset

Since there is no publicly available Multi-signal LiDAR benchmark dataset with ambient illumination, multi-echo point cloud and reflectance measurements for 3D object detection evaluation, we collect two new datasets with the multi-signal measurements to evaluate our method.

Real Dataset is collected by a roof-mounted prototype LiDAR on top of a vehicle driving around a North America city. The LiDAR provides three-echo point cloud along with ambient and reflectance value for each point. Each echo group $EG = [(p_1, i_1), (p_2, i_2), (p_3, i_3), a]$, where $p = (x, y, z)$ is a 3D point coordinate, i is reflectance value for each echo point and a is the ambient value for the echo group. The echoes are ordered by signal strength, so the first echo has the highest reflectance value within the echo group. None detected echoes are marked empty with zero reflectance. The converted 'LiDAR image' has resolution $[96, 600]$, meaning that there are 96 vertically aligned echo groups each column and 600 horizontally aligned echo groups each row. The dataset consists of 35,850 frames collected from various driving scenes including downtown, highway, suburban areas, etc. We split our dataset into training and testing sets with a 70/30 ratio, where the training set consists of 25,002 frames and validation set has 10,848 frames, both sampled from different video clips with high diversity. Ground truth labels of the dataset are 3D oriented bounding boxes of 'Person' and 'Car' classes in the 3D space, and the 2D bounding boxes of the same classes in the 'LiDAR image' space.

Synthetic Dataset is collected using the CARLA [3] simulator. It is a large-scale multi-sensor multi-task dataset. Similar to the real dataset, we also collect three-echo point cloud with reflectance value. We approximate ambient value with r-channel of RGB image because they both capture ambient sunlight signal of close wavelength. We use 26,043 frames for training and 8,682 frames for testing. The format of the synthetic dataset is the same as the real dataset. Moreover, the synthetic dataset provides a wide range of ground truth annotation including 2D and 3D bounding boxes and segmentation. Among various classes, we focus on 'Car', 'Person' and 'Cyclist.' We will release our synthetic dataset to public for reproduction and competition. More details of our synthetic dataset are provided in the supplementary.

4.2. Implementation Details

Network Architecture. In order to align network inputs, we randomly subsample 16K points from multi-echo point cloud in each scene. Note that multiple point clouds are treated as one whole set to sample from. In the proposal generation stage, we follow the network structure of [20] with four set abstraction ($[4096, 1024, 256, 64]$ with multi-scale grouping) and four feature propagation layers as our 3D feature extraction backbone. For 2D detector, we use Faster-RCNN [21] with Feature Pyramid Networks (FPN) [14] module and ResNet-50 [7] as backbone.

In the bounding box refinement network, we randomly sample 256 points from each reassigned point set as input to MEA module. We follow the network structure of [20] with three set abstraction layers ([64, 16, 1]) to generate a single feature vector for each of the point sets. Then the two features are concatenated to jointly perform estimation and regression heads.

Training Parameters. The two stages of our methods are trained separately using Adam optimizer [9]. For real-world dataset, stage-1 is trained for 150 epoch with learning rate

0.002, and stage-2 is trained for 50 epochs with learning rate 0.001. For synthetic data, stage-1 is trained for 100 epoch with learning rate 0.002, and stage-2 is trained for 40 epochs with learning rate 0.002. For bin-based proposal generation and refinement module, we adopt the same bin size, search range and orientation numbers as in [23]. During confidence estimation, a 'Car' proposal is considered positive if its maximum 3D IoU is above 0.6, and negative if its below 0.45. For 'Person', the positive and negative thresholds are set to be 0.5 and 0.4.

Our 2D detector is pre-trained on ImageNet classification and COCO object detection. We then fine-tune it on our datasets using LiDAR image as inputs to detect 2D bounding boxes. Batch size, weight decay and momentum are set to be 8, 1e-4 and 0.9 for both datasets. Learning rate is set to be 0.005 for real-world dataset and 0.01 for synthetic dataset. Horizontal flipping is used for data augmentation.

4.3. Results on Real-World Dataset

We compare our model against state-of-the-art point-based and grid-based 3D object detectors under different point sets. For evaluation metric we use average precision (AP) under different IoU thresholds, where for 'Car' class we use IoU $= \{0.5, 0.7\}$, and for 'Person' we use IoU $= \{0.25, 0.5\}$. Difficulty level is chosen based on the depth value, where easy class contains objects within $40m$, moderate class contains objects between 40-80m and hard class contains objects between 80-200m.

Comparison with state-of-the-art methods. We show the evaluation results and comparison with SOTA methods in Table 1. Note that rows corresponding to "1st echo" refers to experiments using the strongest echo, i.e., with the highest intensities – This mimics the classic single-echo point cloud data where only the strongest point is preserved. Rows corresponding to full echo means "multi-echo points grouped as one point cloud." On both 'Car' and 'Person' class with different IoU threshold, our method outperforms state-of-the-art methods with remarkable margins. For 'Car' class, our method achieves up to 2.9 AP increase and for 'Person' class we achieve up to 4.8 improvement. The improvements over 'Easy' and 'Moderate' classes are both noticeable. Note that we do not get noticeable improvement for hard objects farther than 80m, because multiple modalities provide less information faraway objects. First, the number of multi-echo points decreases quickly as distance increases because of the signal attenuation. Also, they are hard to detect on LiDAR image because of the small size. Under 'Person' class, our method outperforms previous methods on all three difficulty levels.

Notice that all previous methods using "full echo" perform better than using only "1st echo", despite that this is achieved by simply merging points from available echos together as a single point cloud. This means that raw information contained in other echos (e.g., 2nd, 3rd echos) in addition to the strongest echo (i.e., the 1st echo) can be used to improve performance. Moreover, our method is designed to better extract and aggregate features from multi-signal information, so can achieve even higher performance.

Table 2. Effects of different input signals on overall 3D detection performance, where SE stands for single-echo and ME stands for multi-echo.

Point Cloud	Ambient Signal	ME Reflectance Signal	Overall AP
SE			50.5
ME			53.6
ME	✓		53.7
ME		✓	55.1
ME	✓	✓	**55.5**

Table 3. Effects of our proposed components on overall detection, including the MSF module and MEA module.

Method	MSF Module		MEA module	Overall AP
	class feature	pixel feature		
baseline				50.5
MSF only	✓	✓		52.1
MEA only			✓	53.6
w/o class feature		✓	✓	54.0
w/o pixel feature	✓		✓	55.3
MSLiD	✓	✓	✓	**55.5**

4.4. Ablation Study

We conduct extensive ablation experiments to analyze the effectiveness of different proposed components of our model and other design choices. Following [33, 31], all ablation studies are conducted on the 'Car' class.

Effect of different input signal. In Table 2, we first test the point cloud sets where we take only single-echo (SE) or multi-echo (ME) point cloud as input. For ME we use three echo groups of points and for SE, we merge three groups together as one single group. The first two rows shows that our method which leverages multi-echo point features through the MEA module is better than simply merging all echo groups, achieving a 3-point improvement in AP. We then ablate the ambient signal as well as the ME reflectance signal to show their effect. As the table indicates, MSLiD also makes improvement on detection results by properly leveraging these two types of LiDAR signals – Adding ambient and ME reflectance signal improves the overall AP by 1.9% from the point-cloud-only baseline.

Effect of MSF and MEA modules. In Table 3 the 1st row, we remove both the MSF module (use 3D GNN feature only) and the MEA module (treats all points identically). This results in a baseline working with only single-echo point cloud. In the following two rows, we disable one of the two modules to validate how it contributes to the our method. Then, we look inside MSF module and ablate each of the two 2D branch feature vectors. As table shows, the two proposed modules together improve absolute AP by 5.0 from the baseline. For the 'Car' class, multi-echo aggre-

Table 4. **Performance on synthetic dataset**. Previous SOTA methods are trained with all multi-echo points grouped as one (Full echo training). The evaluation metric is Average Precision (AP).

Method	Car - IoU = 0.7			Pedestrian - IoU = 0.5			Cyclist - IoU = 0.5		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND [27]	76.9	70.8	63.2	56.2	51.8	43.1	60.7	55.5	48.7
PointRCNN [23]	77.8	71.4	63.3	59.9	53.4	44.8	61.1	55.7	48.8
PV-RCNN [22]	80.4	72.8	64.5	61.2	54.7	45.5	63.5	57.0	49.3
MSLiD	**81.6**	**73.9**	**66.9**	**66.2**	**59.1**	**49.3**	**64.6**	**58.2**	**51.2**
Improvement	*+1.4*	*+1.1*	*+2.4*	*+5.0*	*+4.4*	*+3.8*	*+1.1*	*+1.2*	*+1.9*

Table 5. Performance for difference feature aggregation schemes (column 1-3) and point cloud set definition (column 4-5) of bounding box refinement stage. pc^{echo} represents the raw multi-echo point cloud, and pc^{reassign} represents point sets after using reassignment strategy described in Sec 3.2.

Aggregation Scheme			Point Cloud Sets		Overall AP
Max-P	Mean-P	Concat.	pc^{echo}	$\text{pc}^{\text{reassign}}$	
✓			✓		51.1
	✓		✓		53.9
		✓	✓		54.5
		✓		✓	**55.5**

gation tends to contribute more between the two modules. This improvement comes from that multi-echo points encodes contour and surface reflectivity information useful to estimate location, size and orientation of the object.

Effect of point cloud reassignment and aggregation. We show the results in Table 5, where Max-P, Mean-P and Concat represent different methods to aggregate features learned from different point sets, pc^{echo} and pc^{reassign} represent different point sets definition. The overall AP is calculated over all ground truth proposals without considering difficulty levels. From the 1st-3rd row we can see that using concatenation to aggregate features of different sets results in the best performance, while max-pooling tends to get low performance. This performance gap is because features learned from multiple point sets encode complementary information, which can be better encoded by concatenation than pooling. From the last two rows, we can see that our point set reassignment strategy helps better learn the RoI feature and further improve the absolute AP of 1.

4.5. Results on Synthetic Dataset

We also validate our method on the synthetic dataset. For the synthetic dataset, the difficulty level follows the same definition as in the KITTI benchmark [4], where easy, moderate and hard are arranged by 2D bounding box size and occlusion/truncation level. For 'Car', we use IoU $= 0.7$, and for 'Pedestrian' and 'Cyclist' we use IoU $= 0.5$. As shown in Table 4, our proposed method outperforms state-of-the-art methods by large margins on all three classes. For 'Car' and 'Cyclist' classes, we get even higher improvements on the hard class, because the hard class is not defined merely by distance, so multi-signal information still helps better detect truncated/occluded objects. The experiment further proves that our proposed method has a good generalization property and works better for different datasets.

4.6. Discussion on Cost-Benefit Trade-off

The size of MSLiD is 10G with a batch size of 4. The inference time is 110ms on a RTX 2080Ti GPU, with PointNet++[20] backbone taking bulk of the time. We also show that leveraging multi-echo points for performance improvements does not significantly increase the running time. In our experiments, we always sample a total of 16K points in different settings (full echo or only the 1st echo). Thus, adding the MSF module or not requires the same time to process the full echo or only the 1st echo. The only source of additional time cost when using multiple echoes in our method comes from the MEA module, where a 3-layer MLP is used to learn separate RoI features for two point sets. It increases AP by 3.4 according to Table 3 with marginal increase of inference time. Also, there is no additional space needed for multi-echo fusion as we sample the same number of points in different settings.

5. Conclusion

We proposed the first method exploring to fuse a wide range of Multi-signal LiDAR information for 3D object detection. Our method takes advantage of multi-echo point clouds and ambient/reflectance signals to learn discriminative point features and proposal regional RoI features. In the proposal generation stage, a multi-signal fusion (MSF) module was proposed to fuse 2D CNN features learned from 'LiDAR image' with 3D GNN features learned from point cloud. In the refinement stage, a multi-echo aggregation (MEA) module was proposed to learn a better object context RoI feature from multi-echo point clouds. The better RoI feature leads to accurate bounding box refinement. Our proposed method achieved state-of-the-art performance in two datasets with multiple LiDAR measurements.

References

[1] Lucas Caccia, Herke Van Hoof, Aaron Courville, and Joelle Pineau. Deep Generative Modeling of LiDAR Data. *IROS*, 2019. 4

[2] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2, 3

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 6

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 8

[5] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2

[6] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 6

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[8] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2020. 3

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[10] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2, 3

[11] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 2

[12] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019. 3

[13] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 2

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4, 6

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[16] S. Pang, D. Morris, and H. Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393, 2020. 3

[17] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4404–4413, 2020. 3, 4

[18] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2, 3, 4

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *CVPR*, 2017. 2, 5

[20] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NeurIPS*, 2017. 2, 4, 6, 8

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6

[22] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 6, 8

[23] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 2, 4, 5, 6, 7, 8

[24] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 4

[25] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 3

[26] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749. IEEE, 2019. 2

[27] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 6, 8

[28] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155, 2018. 2

[29] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[30] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 2, 6

[31] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019. 2, 4, 5, 7

[32] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. *arXiv preprint arXiv:2004.12636*, 2020. 3

[33] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2, 7

[34] Ming Zhu, Chao Ma, Pan Ji, and Xiaokang Yang. Cross-modality 3d object detection. *arXiv preprint arXiv:2008.10436*, 2020. 3