IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 June 2022, accepted 25 July 2022, date of publication 10 August 2022, date of current version 17 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197765

RESEARCH ARTICLE

LossDistillNet: 3D Object Detection in Point Cloud Under Harsh Weather Conditions

ANH THE DO[1] AND MYUNGSIK YOO[2]
[1]Department of Information Communication Convergence Technology, Soongsil University, Seoul 06978, South Korea
[2]School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea

Corresponding author: Myungsik Yoo (myoo@ssu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology under Grant NRF-2021R1A2B5B01002559.

ABSTRACT Recently, 3D object detection models have achieved very good performance under normal weather conditions, with the SE-SSD model having produced the highest performance by exchanging features between the teacher and student models. However, the performance of this model is significantly reduced by adverse weather conditions. Therefore, instead of training the teacher and student models simultaneously, we applied the knowledge distillation algorithm. In this algorithm, the teacher model is trained first by normal input, and the student model is then trained with distillation and student loss by adverse weather condition input. Although recent research has focused on combining different types of sensor inputs to enhance the original model's performance in inclement weather, there are no studies that directly address the problem of missing points for point clouds. Accordingly, we applied a probability estimation, which includes a Deep Mixture of Factor Analyzers (DMFA) network and loss-convolution layer, to recover lost points. We conducted a model evaluation in both fog and snow environments at three levels of density - light, medium, and heavy - and compared the proposed model's performance with that of two state-of-the-art models: one with normal weather condition, and the other with harsh weather conditions. Consequently, our proposed method was shown to significantly outperform the two existing models.

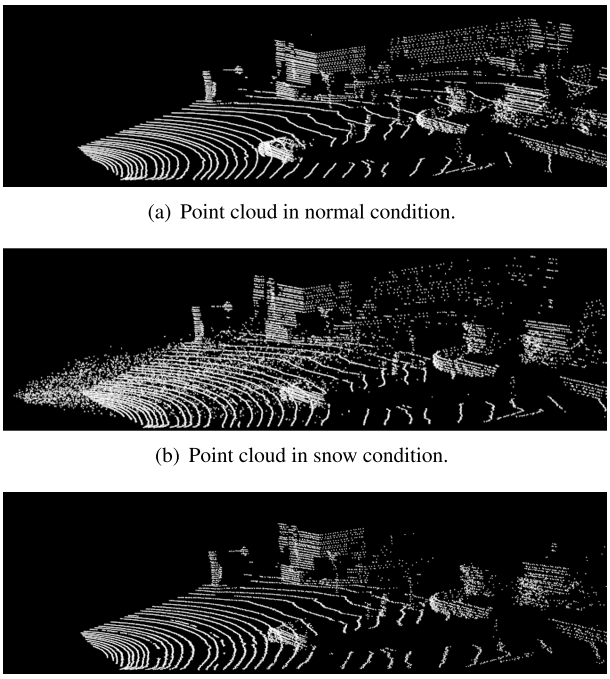
INDEX TERMS Autonomous vehicles, LiDAR, 3D object detection, adverse weather conditions, knowledge distillation.

I. INTRODUCTION

Thanks to the rapid development of 3D sensing technology, a 3D scanner known as LiDAR (Light Detection and Ranging) was designed to use light in the form of laser pulses to measure range. LiDAR sensors can accurately measure their distance from surrounding objects and provide rich geometric information, including scale. Each LiDAR scan generates a 3D point cloud, consisting of a graphical representation of the surroundings, where each point contains information regarding its Euclidean distance. This type of sensor can provide long-range high-resolution detection, and works well under varying lighting conditions. As self-driving technology improves the safety of modern vehicles, makes driving more accessible, and paves the way for fully autonomous vehicles, it relies heavily on precise sensor data, using systems of expensive sensors like LiDAR to ensure accurate environmental awareness. One of the most important problems in the development of autonomous vehicles and driver assistance systems is performance degradation in adverse weather conditions, such as snow and fog. Because these conditions impair human visibility, the proper functioning of driver assistance systems becomes even more essential for the driver. Unfortunately, adverse weather conditions also negatively impact the performance of LiDAR sensors. Fig. 1 illustrates changes in the point cloud for foggy and snowy weather compared to regular conditions. Accordingly, our aim was to improve the accuracy of 3D object detection in snowy and foggy weather conditions.

Leading the way in performance for 3D object detection models [23], [24], [25] is the SE-SSD model [9], which uses voxel-based representation. The SE-SSD model [9] includes

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.



(a) Point cloud in normal condition.

(b) Point cloud in snow condition.

(c) Point cloud in fog condition.

FIGURE 1. Point cloud simulation in adverse weather conditions.

a student network, which plays a major role in 3D object detection, and a teacher network, which transfers knowledge to the student network. The input of the student network is different from that of the teacher network. A shape-aware data augmentation is applied to the point cloud to generate an augmented dataset as input to the student. Although the teacher and student models are trained simultaneously, the latter are updated directly, whereas the former are updated based on student parameters using the exponential moving average (EMA) strategy. By incorporating a voxel-based representation to complement the learning process between teachers and students, SE-SSD [9] achieves optimal results for 3D object detection under normal weather conditions. However, this performance significantly degrades under harsh weather, as the data augmentation methods employed by the SE-SSD model [9] are ineffective due to loss of data on the point cloud.

Our proposed method was designed to improve performance of the SE-SSD model under foggy and snowy conditions. Instead of training the teacher and student models simultaneously with the same dataset, our model trains the teacher model first, and subsequently trains the student model with the student and distillation loss functions. In addition, our knowledge distillation algorithm uses normal weather input for the teacher network, and adverse weather input for the student network. Furthermore, we designed a novel strategy to recover the loss of points in the point cloud caused by adverse weather.

Weather-related problems in 3D object detection have been addressed by numerous studies. Typically, these studies employ methods that fuse input sensors to avoid environmental influences on the LiDAR sensor. The model designed in [5] uses four input sensors: camera, LiDAR, radar, and gated Near-Infrared (NIR) sensors. This scheme is a single-shot model that employs measurement entropy to adaptively fuse features. Specifically, the model handles asymmetric measurement corruption in the four sensor streams by an adaptive deep fusion architecture that exchanges features in intertwined feature extractor blocks. This adaptive deep fusion is steered by measured entropy, which allows for the learning of models that can be generalized to adverse weather conditions. However, the use of multiple sensors is expensive, and the simultaneous processing of data from all sensors requires a large amount of computation. The Sparse LiDAR and Stereo Fusion (SLS-Fusion) [43] model employs a late-fusion method between LiDAR sensors and 2D cameras to generate a pseudo point cloud [26] for 3D object detection. Although this model works well under normal weather conditions, its performance decreases significantly in foggy conditions. Accordingly, the model was modified in a subsequent study [6] by implementing a specific training strategy that uses both normal and foggy weather datasets to achieve higher performance. The study further points out that the late-fusion-based architecture can perform well with a justifiable training strategy in foggy weather conditions. The limitation of both studies is their failure to address the direct problem caused by adverse weather.

Under adverse weather conditions, as the laser pulse traverses the scattering medium, the total intensity is attenuated exponentially with distance. This causes erroneous measurements in the point cloud data, which arises from the reception of back-scattered light from the fog or snow particles in the air. Thus, the point cloud obtained in such conditions contains objects with missing point data. Our proposed architecture employs a probability estimation method to recover the lost data points. Specifically, by combining a DMFA network with a loss-convolution layer, our proposed model outperforms the models designed in [5] and [6].

Specifically, we make the following contributions:

- This study successfully used two different inputs for the knowledge distillation algorithm with the purpose of narrowing the performance gap between the student and teacher models.
- By applying Deep Mixture of Factor Analyzers network and loss-convolution, our model is the first to directly resolve the loss of data in the point cloud under harsh weather conditions.
- Our model's results outperform those of the state-of-the-art models at different density levels of simulated weather. Even when the input data is changed, the performance remains satisfactory.

II. RELATED WORK

Currently available 3D object detection methods generally employ three methods of representing point clouds: pillar-based method, point-based method, and voxel-based method.

TABLE 1. Operational information and result comparison for LiDAR-based 3D Object Detectors in normal condition.

Model	Number of stage	Data representation	Feature extraction		Detector network	Accuracy (%)
			3D backbone	2D backbone		
VoxelNet [11]	One	Voxel	3D CNN	FPN	RPN	64.17
PointPillar [12]	One	Pillar	-	FPN	RPN	74.31
TANet [17]	One	Pillar	-	FPN	RPN	75.94
SECOND [14]	One	Voxel	3D sparse CNN	FPN	RPN	75.96
Point-GNN [15]	One	Point	Graph Neural Network	-	Multi-layer perceptrons	79.47
3DSSD [16]	One	Point	PointNet^{++} + Encoder	-	Fully connected layer	79.59
CIA-SSD [21]	One	Voxel	3D sparse CNN	FPN	SSFA + AF + IoU NMS	80.28
SE-SSD [9]	One	Voxel	3D sparse CNN	FPN	SSFA + AF + SE	82.54

This section presents a brief review of existing 3D object detection methods [37], [38], [39] based on point cloud data for autonomous vehicles, as shown in Tab. 1.

A. VOXEL-BASED METHODS

Voxel-based methods [27], [28], [29] divide a point cloud into evenly spaced 3D voxels, and transform a group of points within each voxel into a unified feature representation through the voxel feature encoding (VFE) layer. This enables interaction between points within a voxel by combining point-wise features with a locally aggregated feature. Stacking multiple VFE layers enables the learning of complex features for characterizing local 3D object information.

Inspired by the capabilities of VoxelNet, [11] is a single-stage detector that introduces the voxel feature extraction (VFE) layer. Fixed voxelization is used to voxelize the LiDAR point cloud, with each non-zero voxel encoded with a feature vector derived from the VFE module. To extract high-dimensional characteristics from voxelized 3D data, a 3D backbone network with a 3D CNN encoder architecture is used. The 3D feature map is then concatenated along the Z axis to create a BEV (Bird's Eye View) feature map, which is then sent to a 2D backbone network for feature extraction. The generated high-dimensional BEV feature map is then input into a Region Proposal Network (RPN) [33], with anchors to generate probability scores and 3D bounding box regression maps.

SECOND [14] (Sparsely Embedded Convolutional Detection) is a single-stage detector driven by the computing efficiency of 3D spatially sparse submanifold convolutions. It incorporates a novel type of angle loss regression, as well as a data augmentation technique. Similarly to VoxelNet [11], SECOND uses fixed voxelization to voxelize the LiDAR point cloud, and extracts voxel features using the VFE module. In the instance of yaw angle regression, SECOND [14] uses a sine-error rather than radian-error loss to avoid the huge loss in the cases of 0 and π radians, as the 3D bounding box is comparable in both situations. The network predicts a binary direction classifier to handle the direction problem that results from sine-error loss.

The Confident IoU Aware Single Stage Detector (CIA-SSD) [21] is an extension of SECOND [14] that performs voxelization to a LiDAR point cloud by encoding each voxel with the mean coordinates and reflection intensity of points within it. The Spatial-Semantic Feature Aggregation (SSFA) module is a 2D backbone network that consists of two groups of 2D convolutions - one for spatial characteristics and the other for semantic features - to encode and learn high-dimensional features in BEV while maintaining high spatial resolution. The two BEV feature maps are merged at the end of the 2D backbone network via an attentional fusion (AF) module. A classification score is predicted using an Anchor-based RPN, 3D bounding box regression, and direction classification score maps, as well as an extra IoU confidence score map. The score map is used to remedy classification confidence by increasing the influence of high IoU confidence scores to compensate for the lack of a 3D bounding box prediction refinement stage in two-stage detectors. Finally, a unique distance-variant IoU non-maximum suppression (IoU NMS) [34] algorithm is used to filter the 3D bounding box proposals.

The Self-Ensembling Single Stage Object Detector (SE-SSD) [9] was developed based on CIA-SSD [21] by implementing the self-ensemble (SE) algorithm for the teacher and student models. The training process was optimized by the formulated consistency constraint to better align predictions with the soft targets from the teacher predictions. Furthermore, a new augmentation scheme was designed to produce shape-aware augmented ground-truth objects for the student model's input. In addition, an orientation-aware distance-IoU (ODIoU) loss was applied to supervise the detector using hard targets from the augmented ground truth.

Because Voxel-based models have topped the recent ranking of 3D object detection methods, our model was built upon a Voxel-based model to attain optimal performance under adverse weather circumstances.

B. PILLAR-BASED METHODS

The pillar-based methods [30], [31], [32] utilize PointNet [13] to learn a representation of point clouds organized in vertical columns (pillars). By learning features instead of relying on fixed encoders, this method can leverage the full information represented by the point cloud. Pillar-based methods exhibit fast performance because all key operations can be formulated as 2D convolutions, which are extremely efficient to compute on a GPU.

PointPillars [12] is a single-stage detector, and the first to utilize the concept of segmenting space in pillars for 3D object detection in autonomous driving applications. The LiDAR point cloud is segmented into pillars, and a maximum number of points per pillar is selected. Each point is encoded by a nine-dimensional vector consisting of its original location $[x, y, z]$, reflection intensity r, offset distance from the pillar center $[x_p, y_p]$, and distance from the arithmetic mean of all points within the pillar $[x_c, y_c, z_c]$. This vector is fed through a simplified version of a VFE layer to extract the pillar features. The result is a BEV feature map.

TANet [17], which stands for Triple Attention Network, is a single-stage detector based on PointPillar [12] that utilizes a triple attention module to extract voxel/pillar features. In addition to being robust to noisy data, TANet employs coarse-to-fine regression in its 2D backbone network to improve localization accuracy without adding a significant computational cost. The LiDAR point cloud is segmented into pillars, and two sequential triple attention (TA) modules composed of point-, channel-, and voxel-wise attention groups are used to extract the pillar feature. The excitation operation is used to compute the point- and channel-wise feature vectors, and fully-connected layers are used to compute the voxel-wise feature. All three vectors are fused together to form the TA feature vector. Consequently, TANet achieves competitive results in 3D object detection.

C. POINT-BASED METHODS

The point-based methods [40], [41], [42] are unified methods that directly take point clouds as input. The model then learns to summarize an input point cloud by a sparse set of key points, which roughly corresponds to the skeleton of objects according to visualization. This method is highly robust to small perturbations of input points, as well as corruption through point insertion and deletion.

Inspired by advances in graph convolutional networks and the research potential of using graph neural networks for 3D object detection in LiDAR point clouds, Point-GNN [15] is a single-stage detector that utilizes a graph-based approach from start to end. The LiDAR point cloud is downsampled to a fixed size through voxelization, and a graph is then constructed with voxels used as vertices. Each vertex is connected to its neighbors within a fixed radius. The initial feature of each vertex is calculated through multi-layer perceptrons (MLPs) [35] in a light version of PointNet [13], as in PointPillars [12]. Unlike common Graph Neural Networks (GNNs) [36], Point-GNN was redesigned to encode spatial information along with learned high-dimensional features. The GNNs execute for a fixed number of iterations, each of which uses different MLPs. Thus, weights are not shared between iterations. Subsequently, in an anchor-free approach, two different MLPs are used: one for classification, and the other for per-class 3D bounding box regression.

3DSSD [16], which stands for 3D Single Stage Detector, is a single-stage detector that uses the natural, unstructured form of a point cloud. The architecture of 3DSSD [16] is designed to speed up the inference time of a point-based detector by discarding the up-sampling layers of a semantic PointNet++ [18] in the 3D backbone network, as well as the second stage of 3D bounding box refinement. This is achieved through a novel fusion sampling strategy and specially designed 3D backbone network following a PointNet++ [18] encoder architecture. The LiDAR point cloud is sub-sampled to a fixed size through random sampling to obtain a more compact representation. A 3D backbone network Pointnet++ [18] encoder architecture is used to extract a subset of points containing high-dimensional features. Accordingly, a subset of points selected from Distance Furthest Point Sampling (DFPS) and Feature Furthest Point Sampling(FFPS) is received as output from the 3D backbone network. 3DSSD [16] and Point-GNN [15] are based on PointNet [13], which presents a novel deep net architecture design suitable for consuming unordered point sets in 3D. However, 3DSSD [16] yields higher performance.

III. METHODOLOGY

The proposed architecture is shown in this section. The subsection III-A shows the overall architecture. Our probability estimation technique, which uses the DMFA network and loss convolution layer, is described in the subsection III-B. The loss functions of training teacher and student model are all shown in the subsection III-C.

A. OVERALL ARCHITECTURE

Fig. 2 illustrates the overall architecture of our proposed model, with a student model (above) and a teacher model (below). We aimed to train a student model that works well in harsh weather environments based on the distillation of knowledge. Therefore, we used the weather simulation LISA [2] for the point cloud as input to the student model. Our idea is founded on the good detection of the teacher model in normal conditions, through which the student model will learn the knowledge distilled from the teacher model, enabling it to perform well in adverse weather conditions. To accomplish this, we first trained the teacher model in normal conditions, and then trained the student model in harsh weather conditions based on the prediction results of the teacher model and the ground truth. Although the teacher

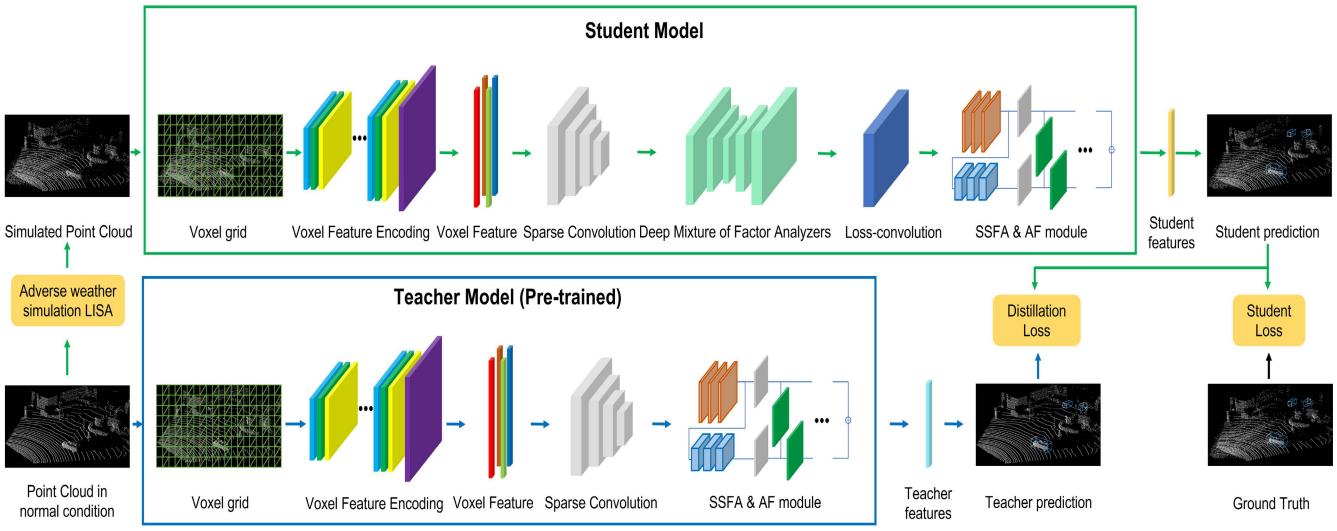


FIGURE 2. Our LossDistillNet consists of two main processing paths. The first component is a pre-trained teacher model that takes a point cloud under normal weather conditions as input. The second component is a student model, which is trained by student and distillation loss, with a simulated point cloud in snowy and foggy conditions as input. The two models feature different architecture, as the student model includes a Deep Mixture of Factor Analyzer (DMFA) network and a loss-convolution layer.

and student models feature different architectures, the object detection principle is similar for both. After the teacher model is pre-trained, our architecture consists of two processing paths, starting from an input point cloud:

In the first path (blue arrows in Fig. 2), the pre-trained teacher model predicts the input point cloud under normal conditions. We used the teacher model's predictions as input for distillation loss to supervise the distillation process for the student model.

In the second path (green arrows in Fig. 2), we used the LISA adverse weather simulation to generate a new KITTI dataset under harsh weather conditions. The new dataset was used as input to the student model to make predictions. These predictions served as inputs for both distillation and student loss to supervise the student model while distilling knowledge from the teacher model and learning from the ground truth.

Architectures of the Teacher and Student Models:

The teacher model has identical architecture to the teacher model in SE-SSD [9]. The model starts with a VoxelNet [11] network, divides the point cloud into box cells, and then uses voxel feature encoding (VFE) to encode into sparse voxel features. Sparse convolution is used to learn information about the z-axis and convert the sparse 3D voxels into 2D bird's eye view (BEV) images. The sparse convolution network (SpconvNet) consists of four blocks ($\{2, 2, 3, 3\}$ submanifold sparse convolution [19] layers), with a sparse convolution layer [20] at the end. Next, we concatenate the sparse 3D feature along z into a 2D dense feature for feature extraction with the spatial-semantic feature aggregation (SSFA) and attentional fusion (AF) modules. We used a single shot detector (SSD)-like [10] to build the SSFA and AF architectures. Finally, three 1×1 convolutions are applied for label classification, location regression, and direction classification.

Similar to the teacher model, the student model has VFE, SpconvNet, SSFA module, and AF module, as well as new additional layers. The point cloud in harsh weather conditions is susceptible to noise and loss of point objects, which increases the disparity between input point cloud distributions of the teacher and student models. This disparity affects the efficiency of knowledge distillation [3], increasing the gap between the two models' performance. Therefore, we applied a DMFA network [4] and a loss-convolution layer to the student model to reduce this disparity.

B. DEEP MIXTURE OF FACTOR ANALYZERS NETWORK AND LOSS-CONVOLUTION LAYER

As mentioned previously, point cloud objects in adverse weather conditions experience point loss, making their detection very difficult and error-prone. To handle this disadvantage, we propose a new learning network that performs well under adverse weather conditions. Our loss-convolution method represents loss data by probability distributions. We used a Gaussian density represented in the form of a Deep Mixture of Factor Analyzers (DMFA) [4] to successfully model image distributions. To model the probability distribution of loss data, we combined the DMFA network with an additional convolution layer responsible for transforming random variables into numerical values.

A voxel feature map is denoted by $v = (v_o, v_m) \in \mathbb{R}^n$, where $v_o \in \mathbb{R}^d$ represents voxel features with known values, whereas $v_m \in \mathbb{R}^{n-d}$ denotes absent values. The set of indices with loss values in sample v is denoted $\mathcal{J} \subset \{1, \ldots, n\}$. While conditional density $p_{v_m|v_o}$ is defined in $(n-d)$-th space, we performed its natural expansion to the whole $\mathbb{R}^n$ space by the following equation:

$$P_{v_m|v_o}(t) = \begin{cases} p_{v_m|v_o}(t_{\mathcal{J}'}), & \text{if } t_{\mathcal{J}'} = v_o. \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where $t_{\mathcal{J}'}$ denotes the restriction of $t \in \mathbb{R}^n$ to the observed points $\mathcal{J}' = \{1, \ldots, n\} \backslash \mathcal{J}$.

The only component of the MFA is the factor analyzer (FA), which has a Gaussian distribution with the covariance matrix extended over low dimensional space, thus greatly reducing the number of model parameters. A single factor analyzer (FA) determined in $\mathbb{R}^n$ is described by the mean vector $\mu \in \mathbb{R}^n$, and the covariance matrix $\Sigma = AA^T + D$, where $A_{n\times l}$ is the rank factor loading matrix consisting of s vector $a_1, \ldots, a_l \in \mathbb{R}^n$, such that $s \ll n$. $D = D_{n\times n} = diag(d)$ is a diagonal matrix representing the noise regardless of $d \in \mathbb{R}^n$. Formally, FA is modeled as a random vector defined by:

$$F_i = \mu_i + \sqrt{d} \odot V + \sum_{j=1}^{s} Y_j \cdot a_j, \tag{2}$$

where $V \sim N(0, I)$, $Y_j \sim N(0, 1)$ are independent, $\sqrt{d}$ denotes the element-wise square root of vector d, and $a \odot b$ refers to element-wise multiplication of vectors a and b.

We considered a random vector F_i with a DMFA distribution P_F to represent a voxel feature map $v = (v_o, v_m)$. Let L be a linear convolution operator that generates a random vector $L * F_i$ when applied to F_i. Then, the random vector $L * F_i$ has a factor analyzer distribution, with mean and variance given by:

$$\begin{aligned} \mathbb{E}[L * F_i] &= L\mu_i \\ \mathbb{V}[L * F_i] &= diag(Ld) + \sum_{j=1}^{s} (La_j) \cdot (La_j)^T. \end{aligned} \tag{3}$$

Next, the activation function is applied to all coordinates of the feature map generated by $L * F_i$. Here, we ignored the correlations between coordinates, and considered only the diagonal elements of the covariance matrix. Consider the 1-dimensional DMFA that represents the distribution of $L * F_i$ on each coordinate of the feature map. Let $P = \sum_{i=1}^{k} p_i N(l_i, \sigma_i^2)$ be the 1-dimensional Gaussian density. The expected value of ReLU applied to a random variable of density P is:

$$\begin{aligned} &\mathbb{E}[ReLU(P)] \\ &= \frac{1}{2} \sum_{i=1}^{k} p_i \Big(l_i + \frac{\sigma_i}{2\sqrt{2\pi}} \exp(-\frac{l_i^2}{2\sigma_i^2}) + l_i \cdot erf(\frac{l_i}{\sigma_i \sqrt{2}}) \Big), \end{aligned} \tag{4}$$

where $erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) dt$ is the error function.

Algorithm 1 summarizes the entire procedure of the DMFA and loss-convolution layer. First, it estimates a distribution of possible replacements for a voxel feature map using a Mixture of Factor Analyzers network. Next, the loss-convolution layer computes the expected value. The DMFA network is trained using log-likelihood loss.

C. LOSS FUNCTION

Our model consists of two main loss functions, one for each of the student and teacher models. We used the common loss functions in object detection to train the teacher model. The loss function for the teacher model consists of three components: localization regression, label classification, and direction classification. We used the Smooth-L1 loss function [10] L_{reg}^t for the bounding box's position and the angle regression task:

$$\begin{aligned} L_{reg}^t &= SmoothL1(\delta_b^t), \\ \delta_b^t &= \begin{cases} |b_t - b_{gt}|, & \text{if } b \in \{x, y, z, w, l, h\} \\ |\sin(b_t - b_{gt})|, & \text{if } b \in \{r\} \end{cases} \end{aligned} \tag{5}$$

where $\{x, y, z\}$, $\{w, l, h\}$, and r denote the center position, size, and orientation of the bounding box, respectively, subscript t denotes the teacher's prediction, and subscript gt denotes the ground truth. Focal loss [8] L_{cls}^t is applied to the label classification task for the teacher model:

$$\begin{aligned} L_{cls}^t &= -\alpha(1 - \delta_c^t)^\gamma \log(\delta_c^t), \\ \delta_c^t &= |\sigma(c_t) - \sigma(c_{gt})| \end{aligned} \tag{6}$$

where α and γ are the focal loss parameters. $\sigma(c_t)$, and $\sigma(c_{gt})$ denote the sigmoid classification scores of the teacher's prediction and ground truth.

The softmax function is used to calculate direction classification loss L_{dir}^t. We use the following approach to create a direction classification target: if the yaw rotation around the z-axis of the ground truth value is higher than 0, the result is positive; otherwise, it is negative. Accordingly, the loss function for training of the teacher model is presented below:

$$L_{teacher} = w_1^t L_{reg}^t + L_{cls}^t + w_2^t L_{dir}^t, \tag{7}$$

where w_1^t and w_2^t are the function's hyper-parameters.

We divided the student model's loss function into two components: the component used for knowledge distillation is called the distillation loss, whereas the component used to minimize the gap between the model's prediction and the ground truth is called the student loss. For distillation loss, we only used the teacher's predictions with high confidence. To filter out the confident predictions of the teacher model, we calculated the IoU indices between teacher and student pairs, and keep only those pairs whose IoU index is greater than the threshold τ_I. The Smooth-L1 loss function L_{reg}^D is calculated for localization distillation between the teacher and student models as follows:

$$\begin{aligned} L_{reg}^D &= \frac{1}{N'} \sum_{i=1}^{N} \mathbb{1}(IoU_i > \tau_I) \sum_b \frac{1}{7} SmoothL1(\delta_b^D), \\ \delta_b^D &= \begin{cases} |b_t - b_s|, & \text{if } b \in \{x, y, z, w, l, h\} \\ |\sin(b_t - b_s)|, & \text{if } b \in \{r\} \end{cases} \end{aligned} \tag{8}$$

where the subscripts t and s refer to the teacher's and student's predictions, respectively. IoU_i denotes the maximum IoU of the i-th student bounding box with all teacher bounding boxes, and N and N' are the initial and final numbers of box pairs, respectively. The cross-entropy and sigmoid loss function are then used to normalize the two predicted confidences

Algorithm 1: Deep Mixture of Factor Analyzers and Loss-Convolution

INPUT:
$v = (v_o, v_m)-$ a voxel feature map
OUTPUT:
ReLU(Lv) - transformation of v by DMFA and loss-convolution layer
DMFA and loss convolution layer:
Compute a density $F = \sum_{i=1}^{k} p_i N(\mu_i.A_i^T A_i + diag(d_i))$, which is the output of DMFA network.
Compute a distribution $L * F = \sum_{i=1}^{k} p_i N(l_i.diag(\sigma_i))$ with a linear convolution operator where:
$l_i = L\mu_i$
$\sigma_i = Ld + \sum_{j=1}^{s}(La_j) \odot (La_j)$
Compute the expected ReLU activation of $P = \sum_{i=1}^{k} p_i N(l_i^j, (\sigma_i^j)^2)$ on every value j using Eq. 4.

such that the deviation between the normalized values can be kept within a small range.

$$L_{cls}^{D} = -\frac{1}{N'}\sum_{i=1}^{N} \mathbb{1}(IoU_i > \tau_I)\sigma(\frac{c_t}{T})\log(\sigma(\frac{c_s}{T})), \quad (9)$$

where $\sigma(\frac{c_t}{T})$, $\sigma(\frac{c_s}{T})$ denote the soft sigmoid classification scores of the teacher's and student's predictions, and the temperature parameter T is introduced to soften the output.

Student loss consists of three major components: localization regression loss L_{reg}^{s}, label classification loss L_{cls}^{s}, and direction classification loss L_{dir}^{s}. We utilized the same formula as in teacher loss.

$$\begin{aligned} L_{reg}^{s} &= SmoothL1(\delta_b^s), \\ \delta_b^s &= \begin{cases} |b_s - b_{gt}|, & \text{if } b \in \{x, y, z, w, l, h\} \\ |\sin(b_s - b_{gt})|, & \text{if } b \in \{r\} \end{cases} \quad (10) \\ L_{cls}^{s} &= -\alpha(1-\delta_c^s)^{\gamma}\log(\delta_c^s), \\ \delta_c^s &= |\sigma(c_s) - \sigma(c_{gt})| \quad (11) \end{aligned}$$

where subscript s denotes the student's prediction. α and γ are the focal loss parameters. Therefore, the overall loss function for training the student model is presented below:

$$L_{student} = w_1^s L_{reg}^s + L_{cls}^s + w_2^s L_{dir}^s + \mu_t(L_{reg}^D + L_{cls}^D), \quad (12)$$

where w_1^s, w_2^s, and μ_t are hyper-parameters.

The parameters and hyper-parameters in the loss functions for training the teacher and student models are shown in Tab. 2. The value of the focal loss parameters α, γ are referred from SECOND [14]. The value of the hyper-parameter w_1^t, w_2^t, w_1^s, w_2^s, μ_t are taken from SE-SSD [9]. The temperature parameter T is referred from the model [7], and the threshold τ_I is set by us.

IV. EXPERIMENTS

The experimental and parameter settings for our model are presented in this section. The subsection IV-A shows the creation of the synthetic dataset, which is used as the input for our model. The process of training the teacher and student models is shown in subsection IV-B.

TABLE 2. Parameters of the loss function.

Loss function	Parameter	Value
Training the teacher model	α	0.25
	γ	2.0
	w_1^t	2.0
	w_2^t	0.2
Training the student model	τ_I	0.7
	T	1.0
	α	0.25
	γ	2.0
	w_1^s	2.0
	w_2^s	0.2
	μ_t	[0; 1]

A. SYNTHETIC DATASETS

In this section, we demonstrate the generation of a new dataset in harsh weather conditions, including fog and snow. We used the LISA simulator [2] and KITTI point cloud data [1] under normal conditions to create new synthetic datasets. The setting up parameters for LISA simulation is shown in Tab. 3. KITTI-Fog are the point cloud simulation datasets under fog conditions, whereas KITTI-Snow are those under snow conditions. Because the regulation of fog and snow density is important for our purposes, we classified the harsh weather conditions into three levels - light, medium, and heavy - as shown in Tab. 4. However, for the sake of consistency the rate of snow and fog is fixed at each density level. Specifically, the selected density scales for light, medium, and heavy levels are 0.1, 0.8, and 1.5 millimeters per hour, respectively. Finally, to accurately evaluate the effectiveness of the proposed model, we generated a new dataset based on the normal dataset KITTI [1] for each density level. Consequently, we used six new synthetic datasets for the point cloud in harsh weather conditions, each of which was divided

into 3,712 training samples and 3,769 validation samples in accordance with the standard protocol.

TABLE 3. Parameters for the LISA simulation.

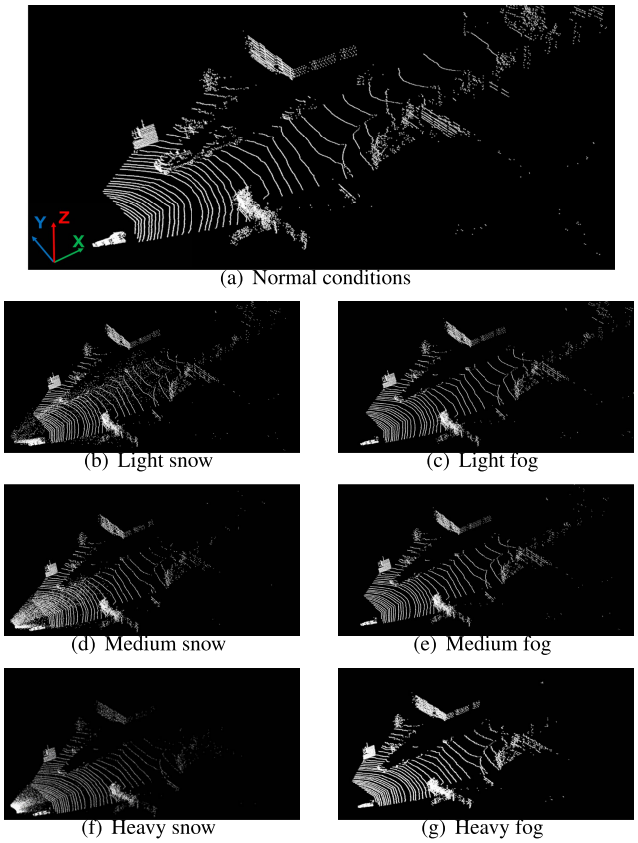
Parameter	Value
The maximum range R_{max}	120 m
The beam divergence B_{div}	3 x $10^{-3} rad$
The range accuracy ΔR	0.09 m
The size of the smallest snow/fog particle D_{st}	50 μm

TABLE 4. Level of simulated weather density.

Level of density	Light	Medium	Heavy
The rate of snow/fog (mm/hr)	0 - 0.8	0.8 - 0.9	> 0.9

The changes in point clouds according to density level under snow and fog are shown in Figs. 4 and 5, respectively. We composed these histograms from the point cloud data shown in Fig. 3. We used the coordinate system available from the KITTI dataset [1], where the origin is the location of the LiDAR sensor, and the direction of the coordinate axes is shown in Fig. 3(a). The horizontal axis represents the distance from a point to the origin based on the x-axis of the point cloud coordinate system, whereas the vertical axis shows the number of points corresponding to the value on the horizontal axis. The four graphs presented in each figure helped us track the distribution of points at different levels based on distance to the LiDAR sensor.

Fig. 3 illustrates the point cloud changes in fog and snow conditions under three density levels. Based on Figs. 4, 3(b), 3(d), and 3(f), it can be seen that in snowy weather, noise points appear concentrated at a location near the LiDAR sensor. In contrast, locations far from the sensor exhibit large amounts of point loss compared to the point cloud in normal weather conditions. As the rate of snowfall increases, the number of noise points increases, and the point cloud loses more data. This decreases the number of valid representation points for objects, and thus detector performance. As shown in Figs. 5, 3(c), 3(e), and 3(g), foggy conditions also result in significant data point loss compared to normal conditions. As the rate of fog increases, the point cloud loses more data, which causes the same issues that are present under snowy conditions. One problem that occurs in both foggy and snowy weather is the loss of point data. Points at different locations in the cloud randomly disappear because the shielding in front of the sensor by fog and snow causes reflections to be suppressed. Therefore, we utilized a DMFA network and loss-convolution layer to recover data points in both snowy and foggy weather conditions.



(a) Normal conditions (b) Light snow (c) Light fog (d) Medium snow (e) Medium fog (f) Heavy snow (g) Heavy fog

FIGURE 3. Point cloud simulation in adverse weather conditions at different levels of intensity.

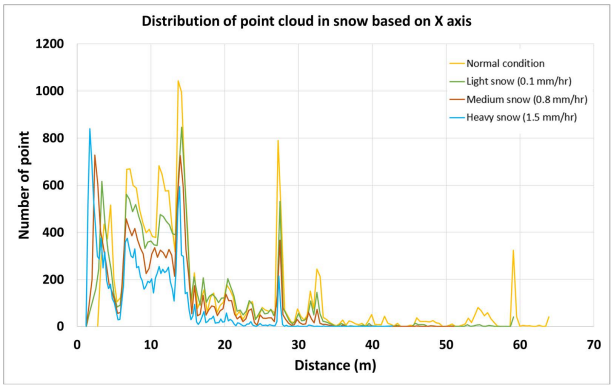


FIGURE 4. Density levels of snow simulation.

B. IMPLEMENTATION DETAILS

Our task consists of three training stages: training the teacher model, training the DMFA network, and training the student model. In this section, we describe all three training phases in detail.

1) TRAINING DETAILS OF THE TEACHER MODEL

The settings for the overall architecture of our teacher model are those used in SE-SSD [9]. We trained the teacher model with a batch size of 4 for 60 epochs. This training stage was

TABLE 5. Comparison between our model and SE-SSD [9] in simulated snow conditions.

Model	Snow								
	Light			Medium			Heavy		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SE-SSD [9]	76.63	60.76	52.32	71.29	57.18	50.14	66.23	55.58	48.14
LossDistillNet (our)	**77.19**	**61.28**	**52.87**	**73.88**	**58.02**	**50.32**	**68.46**	**57.10**	**48.52**

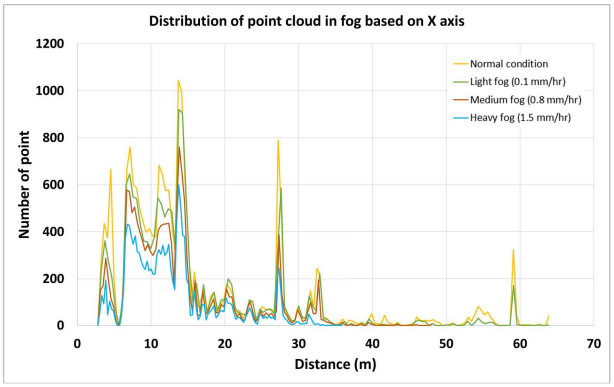


FIGURE 5. Density levels of fog simulation.

run on a NVIDIA GeForce GTX 3060 GPU with 12 GB memory and on the Ubuntu operating system.

2) TRAINING DETAILS OF THE DMFA NETWROK

To apply DMFA architecture to the 3D point cloud, we set it immediately after the sparse convolution block with the goal of recovering loss data for the 2D BEV map. Therefore, the architecture used for DFMA training includes the Voxel encoder network, Sparse convolution, and DMFA layers. Here, we only present detailed architecture for the DMFA layer, as the settings for the remaining network layers are identical to those in SECOND [14].

Due to the increased dimensionality of these datasets, we used a fully convolutional variant of DMFA, which consists of a fully convolutional feature extractor composed of Convolution-ReLU-BatchNorm blocks, followed by a downsampling/upsampling convolution with a stride of 2. The network returns three heads that predict (μ, A, D). Thus, the extractor consists of the following layers:

$$[conv_{128}] \times 2, [conv_{256}] \times 2, [conv_{512}] \times 4,$$
$$[conv_{256}] \times 2, [conv_{128}] \times 2$$

We trained the DMFA for 100 epochs, with a batch size of 4 and a learning rate of 3×10^{-3}. The number of predicted factor analyzers was $l = 4$. We note that a fully convolutional DMFA trained by minimizing only the Negative Log-likelihood (NLL) loss finds it difficult to find a good mean vector μ of the returned density. We mitigated this by supplying the NLL loss with the mean squared error (MSE) loss for the first 10 training epochs. This training stage also was run on a NVIDIA GeForce GTX 3060 GPU with 12 GB memory and on the Ubuntu operating system.

3) TRAINING DETAILS OF THE STUDENT MODEL

For the student model, we used the ADAM optimizer with a batch size of 4 for 60 epochs. We increased μ_t in Eq. (12) from 0 to 1 in the first 15 epochs using the following sigmoid-shaped function: $e^{-5(1-x)^2}$. This training stage also was run on two NVIDIA GeForce GTX 3060 GPUs with 12 GB memory of each and on the Ubuntu operating system.

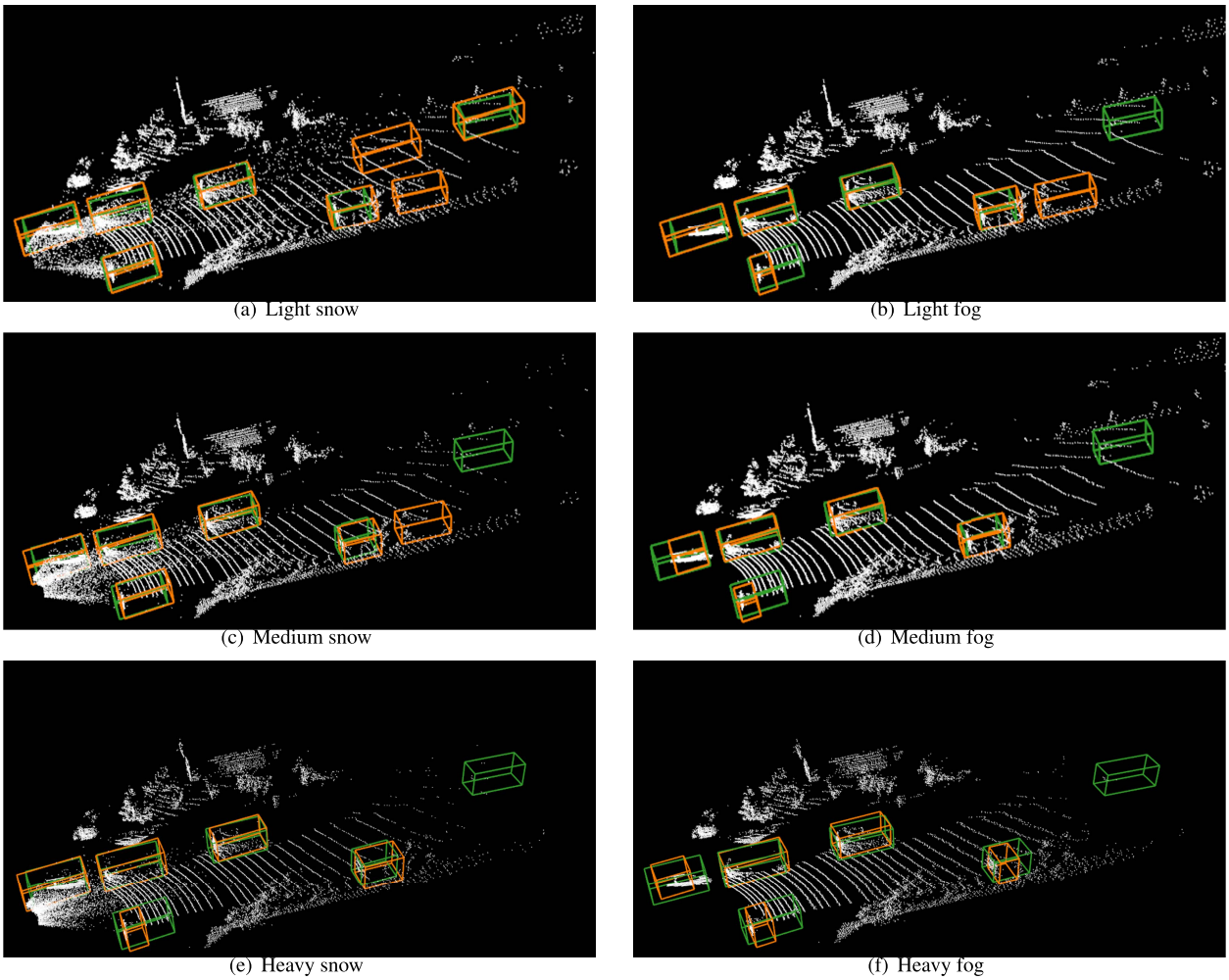
V. RESULTS

A. EVALUATION METRICS

To estimate the performance of the 3D object detection task, average precision (AP) was computed across 40 recall position values between 0 and 1, as described in [22], using an intersection over union (IoU) threshold of 0.7. According to [1], predictions may be classified as Easy, Moderate, and Hard, based on the bounding box size, occlusion level, and truncation level. We evaluated only "Car" object predictions because it was the most important and widely-occurring object in the KITTI dataset.

B. COMPARISON WITH SE-SSD MODEL

In this section, we compared the 3D object detection performance of our proposed model with that of SE-SSD [9]. Results are shown in Tabs. 5 and 6. Although SE-SSD [9] yields the best performance under normal weather conditions, its performance decreases under harsh weather conditions. Therefore, the goal of our study was to improve 3D object detection under adverse conditions. To prove that our proposed model achieves that goal, this comparison was applied for all density levels, as shown in Section IV.A. All of our results outperform those of the SE-SSD [9] model, which proves that the change in training strategy between teacher and student, and the recovery of lost data, are effective for 3D object detection in foggy and snowy weather conditions. The application of the knowledge distillation algorithm improves the student model's learning capacity with respect to the teacher model. Furthermore, the application of a probabilistic estimation method helps the student model reconstruct part of the data lost due to the influence of the harsh environment. Fig. 6 illustrates a sample of car detection results, showing



(a) Light snow (b) Light fog (c) Medium snow (d) Medium fog (e) Heavy snow (f) Heavy fog

FIGURE 6. 3D Object detection in adverse weather conditions at different levels of intensity. 3D bounding boxes in green and orange denote the ground truth and prediction for objects in the scene, respectively.

TABLE 6. Comparison between our model and SE-SSD [9] in simulated fog conditions.

Model	Fog								
	Light			Medium			Heavy		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SE-SSD [9]	73.72	52.36	48.11	67.41	44.20	42.61	55.38	36.43	34.18
LossDistillNet (our)	**74.64**	**52.47**	**48.59**	**69.23**	**45.13**	**43.19**	**57.20**	**37.14**	**35.01**

3D predicted bounding boxes in all density levels of snowy and foggy weather conditions.

C. COMPARISON WITH DEEP ENTROPY FUSION MODEL

Comparison results between the proposed model and Deep Entropy Fusion [5] are shown in Tab. 7. Our proposed model currently yields the optimal performance for 3D object detection in harsh weather conditions. Deep Entropy Fusion [5] used the DENSE dataset as input, which includes data from four sensors: camera, LiDAR, radar, and gated NIR sensors. An adaptive deep fusion architecture was used to handle asymmetric measurement corruptions between the four sensors. For a fair comparison, we used the same dataset as input. However, because our model uses only a LiDAR sensor, results from the other three sensors were not considered. Due to dataset limitations, Deep Entropy Fusion was only evaluated in light fog, heavy fog, and snow conditions. Most results from our model outperform those obtained by Deep

TABLE 7. Comparison between our model and Deep Entropy Fusion model [5].

Model	Fog						Snow		
	Light			Heavy					
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Deep Entropy Fusion [5]	68.43	54.82	**51.91**	28.98	25.24	24.56	67.5	52.26	46.83
LossDistillNet (our)	**69.56**	**55.07**	51.82	**32.07**	**26.84**	**25.72**	**68.92**	**54.21**	**47.81**

TABLE 8. Comparison between our model and SLS-Fusion model [6].

Model	Fog		
	Easy	Moderate	Hard
SLS-Fusion [6]	69.07	47.95	45.50
LossDistillNet (our)	**70.42**	**48.14**	**45.85**

Entropy Fusion [5], with the exception of light fog levels under Hard difficulty.

D. COMPARISON WITH SLS-FUSION MODEL

Tabs. 8 shows comparison results between the LossDistillNet and SLS-Fusion models [6]. SLS-Fusion [6] exhibits superior performance only in foggy weather conditions, as it uses a different fog simulator from that used in our study. Furthermore, the designers of SLS-Fusion created a new dataset based on the KITTI dataset [1], which includes point clouds in normal weather conditions. The SLS-Fusion model [6] uses data from LiDAR sensors and 2D cameras, and generates a pseudo point cloud based on the late-fusion method between cameras and LiDAR sensors. To ensure a fair comparison, we employed the same fog simulation algorithm [5] that was used by the SLS-Fusion model [6]. Because our model only uses LiDAR sensor input, we only generated the synthetic fog dataset for the point cloud. The comparison results in Tab. 8 show that our model outperforms SLS-Fusion at all three levels of fog density. In other words, our proposed model retains optimal performance under a different weather simulation method, even compared to a model that uses two sensor inputs.

VI. CONCLUSION

Currently, few studies have been conducted for 3D object detection in harsh weather conditions. Therefore, we used the LISA algorithm [2] to generate new datasets in snow and fog based on the available KITTI dataset [1]. Our proposal is the first to focus on the task of reconstructing missing data on a 3D point cloud. We succeeded in reconstructing loss data for the point cloud in both foggy and snowy weather by applying a DMFA network and a loss-convolution layer. Furthermore, by implementing the knowledge distillation algorithm [3] we achieved results that are more competitive than those produced by three existing models: SE-SSD [9], Deep multimodal fusion [5], and SLS-Fusion [6]. Future studies will combine 2D images with the 3D point cloud to reduce data loss caused by harsh weather. Furthermore, there is potential for further research on networks capable of reducing noise and reconstructing loss data for point clouds in harsh weather conditions.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[2] V. Kilic, D. Hegde, V. Sindagi, A. B. Cooper, M. A. Foster, and V. M. Patel, "LiDAR light scattering augmentation (LISA): Physics-based simulation of adverse weather conditions for 3D object detection," 2021, *arXiv:2107.07004*.
[3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Stat*, vol. 1050, p. 9, Mar. 2015.
[4] M. Przewiezlikowski, M. Śmieja, and L. Struski, "Estimating conditional density of missing values using deep Gaussian mixture model," in *Proc. Int. Conf. Neu. Inf.* Cham, Switzerland: Springer, 2020, pp. 220–231.
[5] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11682–11692.
[6] N. A. M. Mai, P. Duthon, L. Khoudour, A. Crouzil, and S. A. Velastin, "3D object detection with SLS-fusion network in foggy weather conditions," *Sensors*, vol. 21, no. 20, p. 6711, Oct. 2021.
[7] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 742–751.
[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[9] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-ensembling single-stage object detector from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14494–14503.
[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2016, pp. 21–37.
[11] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
[12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.
[13] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
[14] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.
[15] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1711–1719.

[16] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11040–11048.
[17] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "TaNet: Robust 3D object detection from point clouds with triple attention," in *Proc. AAAI Conf. Conf. Artif. Intell.*, 2020, pp. 11677–11684.
[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5105–5114.
[19] B. Graham, M. Engelcke, and L. V. D. Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.
[20] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 806–814.
[21] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "CIA-SSD: Confident IoU-aware single-stage object detector from point cloud," 2020, *arXiv:2012.03015*.
[22] A. Simonelli, S. R. Bulo, L. Porzi, M. Lopez-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1991–1999.
[23] A. Mahmoud, J. S. K. Hu, and S. L. Waslander, "Dense voxel fusion for 3D object detection," 2022, *arXiv:2203.00871*.
[24] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3D object detection," 2022, *arXiv:2204.12463*.
[25] C. Chen, Z. Chen, J. Zhang, and D. Tao, "SASA: Semantics-augmented set abstraction for point-based 3D object detection," presented at the AAAI Conf. Artif. Intell., 2022.
[26] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8445–8453.
[27] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," in *Proc. Conf. Robot Learn.*, 2018, pp. 146–155.
[28] X. Li, J. Guivant, N. Kwok, Y. Xu, R. Li, and H. Wu, "Three-dimensional backbone network for 3D object detection in traffic scenes," 2019, *arXiv:1901.08373*.
[29] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," 2019, *arXiv:1908.09492*.
[30] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds," *Sensors*, vol. 20, no. 3, p. 704, Jan. 2020.
[31] M. Ye, S. Xu, and T. Cao, "HVNet: Hybrid voxel network for LiDAR based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1631–1640.
[32] R. Ge, Z. Ding, Y. Hu, Y. Wang, S. Chen, L. Huang, and Y. Li, "AFDet: Anchor free one stage 3D object detection," 2020, *arXiv:2006.12671*.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[35] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, nos. 14–15, pp. 2627–2636, Aug. 1998.
[36] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[37] C.-H. Wang, H.-W. Chen, and L.-C. Fu, "VPFNet: Voxel-pixel fusion network for multi-class 3D object detection," 2021, *arXiv:2111.00966*.
[38] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1951–1960.
[39] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9775–9784.
[40] J. Li, S. Luo, Z. Zhu, H. Dai, A. S. Krylov, Y. Ding, and L. Shao, "3D IoU-Net: IoU guided 3D object detector for point clouds," 2020, *arXiv:2004.04962*.
[41] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3D instance segmentation and object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1839–1849.
[42] Q. Meng, W. Wang, T. Zhou, J. Shen, L. V. Gool, and D. Dai, "Weakly supervised 3d object detection from LiDAR point cloud," in *Proc. Eur. Conf. Comp. Vis.* Cham, Switzerland: Springer, 2020, pp. 515–531.
[43] N.-A.-M. Mai, P. Duthon, L. Khoudour, A. Crouzil, and S. A. Velastin, "Sparse LiDAR and stereo fusion (SLS-Fusion) for depth estimation and 3D object detection," in *Proc. 11th Int. Conf. Pattern Recognit. Syst. (ICPRS)*, 2021, pp. 150–156.

ANH THE DO received the B.S. degree in mechatronics from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2020. He is currently pursuing the master's degree with Soongsil University. His research interest includes visible light communication (VLC).

MYUNGSIK YOO received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea, in 1989 and 1991, respectively, and the Ph.D. degree in electrical engineering from the State University of New York at Buffalo, New York, in 2000. He was a Senior Research Engineer at Nokia Research Center, Burlington, MA, USA. He is currently a full-time Professor with the School of Electronic Engineering, Soongsil University, Seoul. His research interests include visible light communications, cloud computing, and internet protocols.