3D harmonic loss: towards task-consistent and time-friendly 3D object detection on edge for V2X orchestration.

ZHANG, H., MEKALA, M.S., YANG, D., ISAACS, J., NAIN, Z., PARK, J.H. and JUNG, H.-Y.

2023

3D Harmonic Loss: Towards Task-consistent and Time-friendly 3D Object Detection on Edge for V2X Orchestration

Haolin Zhang, M S Mekala, Dongfang Yang, John Isaacs, Zulkar Nain, Ju H. Park, Ho-Youl Jung

***Abstract*—The use of edge computing for 3D perception has garnered interest in intelligent transportation systems (ITS) due to its potential to enhance Vehicle-to-Everything (V2X) orchestration through real-time traffic monitoring. The ability to accurately measure depth information in the environment using LiDAR has led to a growing emphasis on 3D detection based on this technology, which has significantly advanced the field of 3D perception. However, the computationally-intensive nature of these operations has made it challenging to meet the real-time deployment requirements using existing methods. The object detection task in the pointcloud domain is hindered by a substantial inconsistency problem caused by its high sparsity, which remains unaddressed. This paper conducts an in-depth analysis of the issue, which has been brought to light by recent research on detecting inconsistency problems in image specialization. To address this problem, we propose a solution in the form of a *3D harmonic loss* function, which aims to alleviate the inconsistent predictions based on pointcloud data. In addition, we showcase the viability of optimizing *3D harmonic loss* mathematically. Our simulations employ the KITTI dataset and DAIR-V2X-I dataset, and our proposed approach significantly surpasses the performance of benchmark models. Additionally, we validate the efficiency of our proposed model through its deployment on an edge device (Jetson Xavier TX) in a simulated environment.**

***Index Terms*—Vehicle technology, Edge computing, Vehicle-to-Everything (V2X) orchestration, 3D harmonic loss.**

I. Introduction

BACKGROUND: Edge computing-based computer vision technology has received global attention for strengthening V2X orchestration and autonomous driving systems (ADS). In the interdisciplinary research areas of V2X and ADS, data is collected and analyzed by vehicles and infrastructures to enable intelligent decision-making for vehicle movement [1]. The decision-making system relies on data captured from surrounding areas, including road structure and traffic information. Through the use of effective object detection methods cloned via Road Side Units (RSU) and On Board Units (OBU), the data is analyzed to identify and localize traffic candidates, enabling necessary decisions for vehicle movement.

Motivation: Let's consider the movement of a vehicle using event-trigger analysis based on traffic information. Traffic data is collected through surveillance devices or on-vehicle sensors such as LiDAR and cameras. Edge devices analyze the important LiDAR data (pointcloud) to achieve the target with low latency. However, computation-intensive services are offloaded to servers to meet application deadlines. The affordability, increased perception of distant objects, and robust characteristics of LiDAR 3D object detection technology have made it prominent. To facilitate pinpoint communication from vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), the recognized and localized vehicle is vital in measuring the surroundings and infrastructure through RSU-LiDAR deployment. Therefore, developing and deploying an efficient and robust 3D detector based on LiDAR data is a crucial research direction to enhance V2X efficiency.

Problem of task inconsistency and time delay:

Object detection tasks in modern times have branched out into various sub-tasks like object localization, classification, and direction estimation. In the 2D image domain, most 2D detections consider sub-tasks independently, leading to inconsistent and unexpected predictions with high classification confidence but inadequate localization after post-processing (e.g. Non-Maximum Suppression), as shown in Fig.1(a). Recently, researchers have addressed and partially solved this inconsistency problem in 2D object detection in [2]–[5]. However, despite advancements in the field, 3D detection accuracy in the point cloud domain continues to be impacted by guesswork and similar inconsistency issues, as illustrated in Fig.1(b), and further validated by real-data experiments displayed in Fig.1(c).

While recent lidar-based 3D object detection methods [6]–[18] focus on achieving the best mAP and consider it as a benchmark for model accuracy, they fail to address other critical factors like time consumption, quality of experience (QoE), and service reliability. For real-time applications like V2X, a cost-effective, task-friendly, and task-consistent detec-

Note: Haolin Zhang, M S Mekala are both contributed equally for accomplishing the targets. Haolin Zhang is with National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China. (E-mail:zhanghaolin@xjtu.edu.cn)

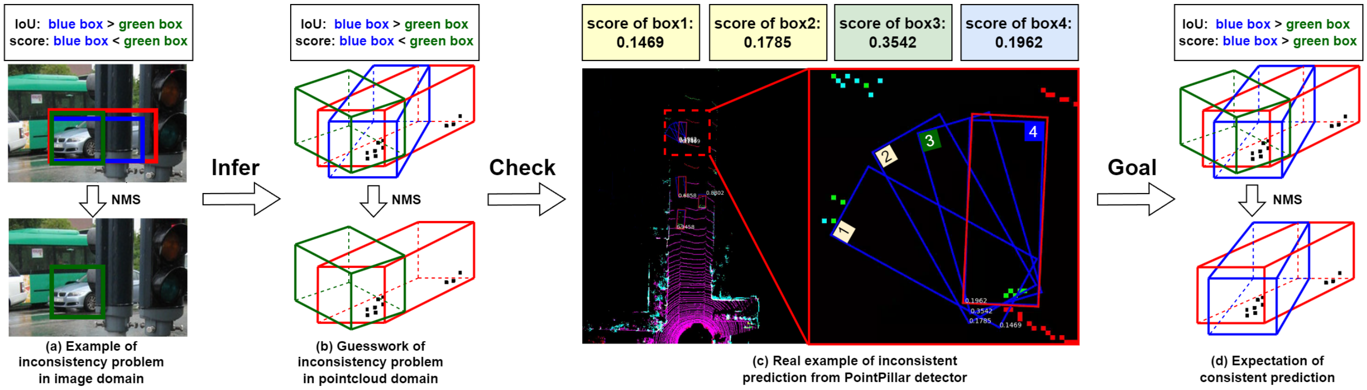
M S Mekala and John Isaacs are with School of Computing, Robert Gordon University, Garthdee Road, Aberdeen, AB10 7QB, Scotland, UK. (E-mail: msmekala@yu.ac.kr & ms.mekala@rgu.ac.uk, j.p.isaacs@rgu.ac.uk)

Zulkar Nain is with Department of Artificial Intelligence and Big Data, Woosong University, Daejeon 34606, South Korea. (Email: zulqarnain@wsu.ac.kr)

Ho-Youl Jung is with department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, Korea as well as RLRC for Autonomous Vehicle Parts and Materials Innovation, Yeungnam University, Gyeongsan 38544, Korea (E-mail: hoyoul@yu.ac.kr (corresponding author)).

Dongfang Yang, Chongqing Chang'an Automobile Co., Ltd. Chongqing, China. (E-mail: yangdf@changan.com.cn).

Ju H. Park, Department of Electrical Engineering, Yeungnam University, Gyeongsan 38544, Korea. (E-mail: jessie@ynu.ac.kr).


IoU: blue box > green box
score: blue box < green box
NMS
Infer
IoU: blue box > green box
score: blue box < green box
NMS
Check
score of box1: 0.1469
score of box2: 0.1785
score of box3: 0.3542
score of box4: 0.1962
1
2
3
4
0.1962
0.3542
0.1785
0.1469
Goal
IoU: blue box > green box
score: blue box > green box
NMS

(a) Example of inconsistency problem in image domain

(b) Guesswork of inconsistency problem in pointcloud domain

(c) Real example of inconsistent prediction from PointPillar detector

(d) Expectation of consistent prediction

Fig. 1. Illustration of the inconsistency problem in object detection. (a) Example of inconsistency problem in image domain: inconsistent bounding boxes with high classification score but low IoU (compared to groundtruth (red box)) in 2D detection, which leads to the suboptimal output (green box) after post processing (NMS). (b) Guesswork of the similar inconsistency problem in pointcloud-based 3D detection. (c) Real example of inconsistency problem from the PointPillar [21]. (d) Expectation of consistent prediction: a better 3D detector is expected to harmonize the localization and classification of predicted objects, resulting in the reasonable output (blue box). Our work focuses on how to alleviate the inconsistent predictions in pointcloud domain, to achieve the expected predictions in real-world applications.

tion solution with fast run-time and low error rate is required. Some researchers [19], [20] have noticed the problem of task inconsistency, but their solutions rely on additional modules that increase inference time, which contradicts our goal of reducing computational burden.

Other recent works like [21]–[26] have attempted to improve deployment metrics like computational burden and execution latency, but they have not achieved a sufficient trade-off between detection accuracy and time consumption for edge device-based simulations in real-time applications.

Our solutions: We derived solutions from the learning optimization perspective to solve the above drawbacks for better edge-computing object detection performance. Firstly, by drawing the lessons from the inconsistent prediction problem in camera-based 2D detection, we indicate a similar inconsistency problem in lidar-based 3D detection. This problem gradually leads to the inaccuracy of the prediction in actual applications and is worth being discovered and resolving. To alleviate inconsistent predictions of 3D detectors, we analyze the cause of the inconsistency problem through the respective characteristics of the image and point cloud. Inspired by the solution in image domain [4], we extend the 2D solution to 3D detection and propose *3D harmonic loss*, a task-consistent learning strategy for optimizing pointcloud-based 3D detectors. It is worth mentioning that our solution, *3D harmonic loss*, not like previous solutions [19], [20], only works for model training and does not bring any extra time-cost to model inference. Secondly, a thorough mathematical analysis is conducted to explain and demonstrate the effectiveness of *3D harmonic loss*. Experiments on KITTI 3D/BEV detection dataset [27] further validate that the proposed strategy can achieve a noticeable performance improvement. Third, our proposed model is deployed on the edge device (Jetson Xavier TX) for simulation, and it achieves an ideal trade-off between time efficiency and detection accuracy.

We deploy the proposed detector on edge devices (Jetson Xavier TX) for realistic simulations to meet the lightweight design and edge-computing benchmark metrics.

Our contributions are as follows.

1) Develop a *3D harmonic loss* method for alleviating inconsistent predictions inspired by related ideas from 2D detection. Thus, we level up the 2D solution to lidar-based 3D detection to map both two-stage and one-stage 3D detection models' learning accuracy without extra time-cost on inference.
2) Experiments on KITTI Dataset [27] and DAIR-V2X-I Dataset [28] demonstrate our proposed work's effectiveness for both on-vehicle and on-infrastructure object detection. Especially for industrial-popular lidar-based detectors such as SECOND [6], and PointPillar [21] are considered to showcase the significant margin of mean average precision (mAP) improvement concerning the proposed *3D harmonic loss*.
3) Realistic simulations by deploying our proposed lightweight detector on the Jetson Xavier device further verify and realise that our solutions are time-friendly and task-consistent towards 3D detection for real applications.

The paper continues as Section II that briefs the extant approaches research gaps. Section III represents the proposed work in detail. Section IV represents the proposed method's effectiveness using qualitative and quantitative analysis. Section V concludes the manuscript.

II. Related Work

A. LiDAR-based 3D object detection

The popularity of 3D object detection has increased with the use of pointcloud-based deep learning models via various frameworks. Typically, two types of frameworks exist, namely one-stage and two-stage. One-stage methods enable instantaneous prediction of object 3D bounding boxes (bboxes). Some of these methods are points-based, like 3DSSD [9], which utilizes the PointNet [29] architecture, and PointGNN [7] network that employs a graph neural network. These methods use raw lidar pointclouds to make 3D shape predictions. Alternatively, voxel-based methods, such as VoxelNet, first convert the lidar pointcloud into 3D voxels to decrease input memory

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3291650

usage. Then, voxel features are fed into a region proposal network using 3D convolutions for 3D detection. SECOND [6] is a more time-efficient approach based on VoxelNet [10] that proposes sparse 3D convolutions. However, the time performance of one-stage 3D detection is still unsatisfactory. VoTr [30] and VoxSeT [14] utilize a voxel-based one-stage method and introduce transformer architecture for improved accuracy. However, their heavy parameters and complicated operations significantly reduce the time performance of 3D detection. PointPillar [21], on the other hand, transforms 3D pointclouds into 2D voxels, followed by highly efficient 2D convolutions to achieve real-time performance and easy deployment of 3D detection [26].

In contrast, the first stage of two-stage detectors [8], [11]–[13], [17], [18] involves predicting the Region-of-Interests (ROIs), while the second stage utilizes a refinement network to detect objects with greater precision. Despite the advantages of some two-stage methods, such as CenterPoint [17], which incorporates a fast feature encoding and a lightweight refinement head in its network design, they often fail to meet the speed requirements of real-time applications. As a result, the time-cost comparison gap between such two-stage detectors and certain one-stage detectors, such as PointPillar [21], remains relatively similar.

Combining image and point cloud data [31]–[35] is a suitable approach for enhancing 3D detection accuracy and surrounding perception. Nonetheless, the fusion techniques are more intricate and time-consuming for real-time applications compared to pure lidar-based detection. Thus, while we did consider some fusion methods in our experiment to demonstrate their accuracy benefits, they are not the primary focus of our discussion.

B. Inconsistency problem in object detection

The issue of inconsistency was first observed in 2D object detection methods within the image domain, as demonstrated in Fig.1(a). In the initial approaches, object classification and localization were treated independently during model training, leading to incongruous predictions during inference. To address this problem, recent studies [2]–[5] have attempted to bridge the gap between these sub-tasks of 2D detection in the image domain. For example, [2], [3] proposed a Generalized Focal Loss approach that did not achieve the desired accuracy, while an improved version of the Focal Loss method was introduced in [36] to ensure consistent 2D detection. [4] introduces a balanced loss function to reconcile prediction consistency. Moreover, [5] proposed a PAA method that included an additional module for predicting IoU, which was useful for selecting positive training samples. Similarly, inconsistency in 3D point cloud object detection systems can lead to lower object detection reliability and quality of experience. Some 3D detection methods [19], [20] may slightly alleviate this issue, even though they are not fundamentally aware of the inconsistency problem. However, these methods modify the structure of the 3D models, requiring additional time and operations for predicting IoU and post-processing, which may not be feasible for real-time environments. Our work is relatively independent to above works, mainly reflected in two aspects: our work first indicates the need to address the inconsistency problem of 3D detection in the point cloud domain. Most importantly, our proposed solution, as a common optimization method for 3D detectors' training, effectively addresses the inconsistency problem without introducing any extra burden during model inference and deployment.

III. Proposed Work

This section presents the formulation of the proposed method 3D harmonic loss, from both a theoretical and mathematical optimization standpoint.

In Fig. 1(d), we aim to attain uniform predictions in 3D detection. The reason behind the inconsistency issue is explored by analyzing the learning loss function (Eq1) for a positive training sample i in several existing methods [6], [8], [12], [21]. It is revealed that the three sub-tasks of 3D object detection (classification, localization (regression), and direction estimation) are handled and monitored separately, resulting in the inconsistency problem.

$$L_{3D}^{i} = L_{cls}\left(p_i, p_i^{gt}\right) + L_{reg}\left(d_i^{'}, d_i^{gt}\right) \; + L_{dir}\left(p_i^{'}, p_i^{'gt}\right) \tag{1}$$

Where p_i is softmax classification score, $p_i^{'}$ is softmax direction score. Also p_i^{gt} and $p_i^{'gt}$ are the ground truths for classification and direction estimation respectively. Consequently, the classification loss ($L_{cls}(p_i, p_i^{gt})$) for positive training samples (p_i^{gt}=1) uses focal loss [37], which is derived as follows

$$L_{cls}\left(p_i, p_i^{gt}\right) = -\alpha(1-p_i)^{\gamma}\log\left(p_i\right) \tag{2}$$

In continuation, the regression loss L_{reg} uses SmoothL_1 [38] as follows

$$L_{reg}\left(d_i^{'}, d_i^{gt}\right) = \sum_{d_i^{\prime}\in\left(x_i^{\prime},y_i^{\prime},z_i^{\prime},l_i^{\prime},w_i^{\prime},h_i^{\prime},\theta_i^{\prime}\right)} \text{Smooth}L_1\left(\Delta_{d_i}\right) \tag{3}$$

$$\text{Smooth}L_1\left(\Delta_{d_i}\right) = \begin{cases} 0.5\Delta_{d_i} & if\,|\Delta_{d_i}|<1 \\ |\Delta_{d_i}|-0.5 & \text{others} \end{cases} \tag{4}$$

Where Δ_{d_i} is the difference between the set of attributes $(x_i^{'}, y_i^{'}, z_i^{'}, l_i^{'}, w_i^{'}, h_i^{'}, \theta_i^{'})$ of predicted offsets $d_i^{'}$ and ground truth offsets d_i^{gt}, which is determined by the parameters $(X_i^{gt}, Y_i^{gt}, Z_i^{gt}, L_i^{gt}, W_i^{gt}, H_i^{gt}, \alpha_i^{gt})$ of ground truth boxes and the parameters $(X_i, Y_i, Z_i, L_i, W_i, H_i, \alpha_i)$ of anchor boxes as follows

$$\begin{aligned} {x_i}^{gt} &= \tfrac{X_i^{gt}-X_i}{\sqrt{(W_i)^2+(L_i)^2}}\,,\; {y_i}^{gt} = \tfrac{Y_i^{gt}-Y_i}{\sqrt{(W_i)^2+(L_i)^2}} \\ {z_i}^{gt} &= \tfrac{Z_i^{gt}-Z_i}{H_i},\;\; {w_i}^{gt} = \tfrac{W_i^{gt}}{W_i}, {l_i}^{gt} = \log\tfrac{L_i^{gt}}{L_i} \\ {h_i}^{gt} &= \log\tfrac{H_i^{gt}}{H_i}, {\theta_i}^{gt} = \sin\left(\alpha_i^{gt}-\alpha_i\right) \end{aligned} \tag{5}$$

The inconsistent handling of various sub-tasks can result in inconsistent inference outcomes, which was addressed in prior research [4]. However, that research was limited to 2D detection using image sources and only focused on generalizing critical loss functions such as cross-entropy, L1, and IoU. Our

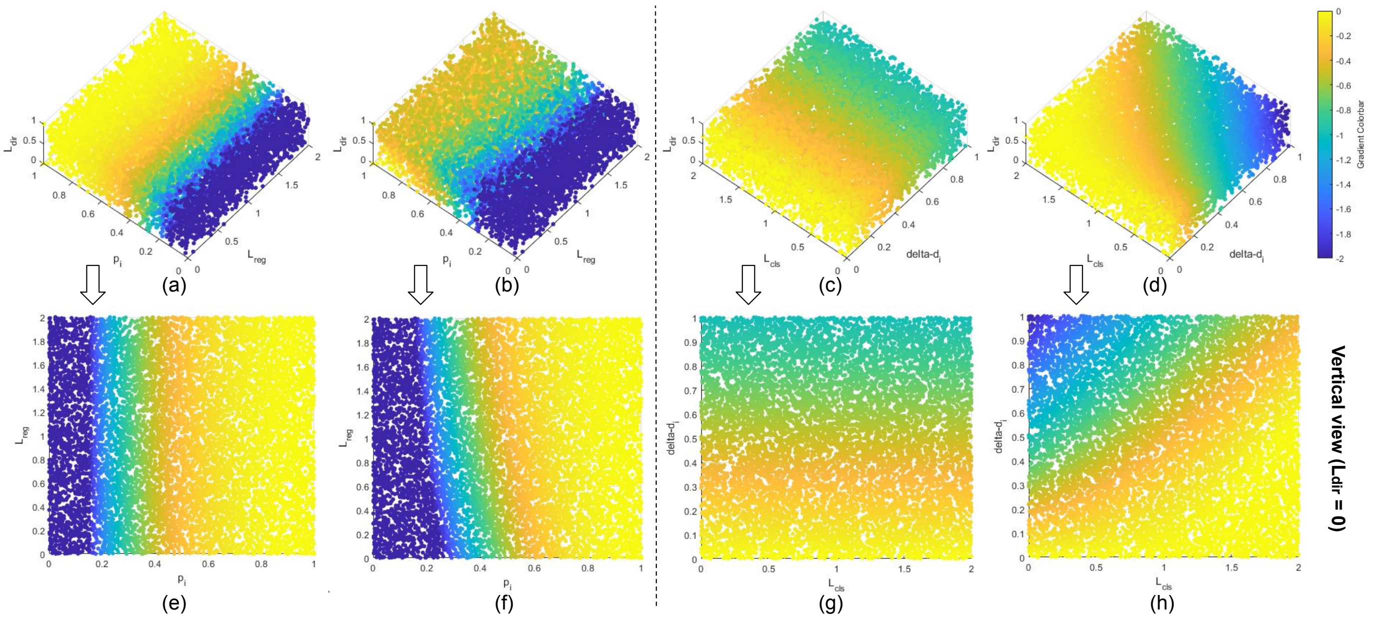


Fig. 2. Visualization of the gradients from 3D detection loss related to different sub-tasks (object classification and object localization (regression)). (a) is drawn with gradients from classification part in common 3d detection loss. (b) is drawn with gradients from classification part in our proposed 3d harmonic loss. (c) is drawn with gradients from regression part in common 3d detection loss. (d) is drawn with gradients from regression part in our proposed 3d harmonic loss. For better view, (e), (f), (g) and (h) show their vertical forms. The color intensity indicates the value of gradients (see colorbar). MATLAB was utilized to analyze the data and plot the diagram.

proposed 3D harmonic loss, detailed in **Theorem**, is specifically designed for lidar-based 3D detection, which involves different data modalities (pointcloud) and multiple prediction dimensions (including direction estimation, height, and depth). To maintain consistency during model learning, three dynamic factors are employed: $1+\beta_r$, $1+\beta_c$, and $1-\frac{\beta_r+\beta_c}{\beta_{dir}}$. The factors $1+\beta_r$ and $1+\beta_c$ work in conjunction to ensure mutual-consistency, while $1-\frac{\beta_r+\beta_c}{\beta_{dir}}$ guarantees intrinsic-consistency. Our approach ensures both mutual and intrinsic consistency among the sub-tasks. In cases where classification optimization falls short, the factor obtained from the classification part supervises the regression part, and vice versa. Additionally, our method guarantees that the classification and regression parts consistently supervise the direction estimation part. This is exemplified by the fact that an accurate direction estimation should align with distinct class recognition and unambiguous boundary regression. Our learning mechanism is well-suited to commonly used loss functions in 3D detection training, including focal loss for object classification, Smooth L_1 for object localization, and binary cross-entropy for object direction estimation. In essence, our approach harmoniously addresses all sub-tasks.

Theorem: 3D harmonic loss

$$L^i_{3D-Har} = (1+\beta_r) \times L_{cls}\left(p_i, p_i^{gt}\right) + (1+\beta_c) \\ \times L_{reg}\left(d_i^{'}, d_i^{gt}\right) + \left(1 - \frac{\beta_r+\beta_c}{\beta_{dir}}\right) \times L_{dir}\left(p_i^{'}, p_i^{'gt}\right)$$

Where

$$\beta_r = e^{-L_{reg}\left(d_i^{'}, d_i^{gt}\right)}, \beta_c = e^{-L_{cls}\left(p_i, p_i^{gt}\right)} \tag{6}$$

The maximum value of β_r (or β_c) cannot exceed 1 unless the regression part (or classification part) of the model has fully converged ($Lreg = 0$ (or $Lcls = 0$)). Consequently, we set β_{dir} to 2 in our 3D harmonic loss function. Once the model has completely converged, the expression $1-\frac{\beta_r(=1)+\beta_c(=1)}{\beta_{dir}(=2)}$ evaluates to 0, indicating that further weighting is unnecessary. When the model is in training, setting $\beta_{dir}=2$ in the expression $1-\frac{\beta_r+\beta_c}{\beta_{dir}}$ ensures that the feedback for optimization from the regression part and the classification part to the orientation estimation part is given equal importance.

The effectiveness of 3D harmonic loss is mathematically proven and briefly explained when the training sample is positively supervised by classification loss $L_{cls}\left(p_i, p_i^{gt}\right)$, regression loss $L_{reg}\left(d_i^{'}, d_i^{gt}\right)$, and direction loss $L_{dir}\left(p_i^{'}, p_i^{'gt}\right)$.

Proof-1: Assuming that the i^{th} training sample is positive, we can assign $p_i^{gt}=1$, and use the values $\alpha = 0.25$ and $\gamma = 2$ for the focal loss. This helps to analyze how effective the 3D harmonic loss is in reducing the classification loss which is derived as follows

$$\frac{\partial L_{cls}\left(p_i, p_i^{gt}\right)}{\partial p_i} = \frac{\partial\left[-\alpha p_i^{gt}(1-p_i)^r \log(p_i)\right]}{\partial p_i} \\ = -\frac{(1-p_i)\left(\frac{1}{p_i}-1-2\log(p_i)\right)}{4} \overset{suppose}{=} J\left(p_i\right) \tag{7}$$

$$\beta_c = e^{-L_{cls}\left(p_i, p_i^{gt}\right)} = e^{-\frac{1}{4}(1-p_i)^2 \log(p_i)} \overset{suppose}{=} K\left(p_i\right) \tag{8}$$

with the point derivation

$$\frac{\partial \beta_c}{\partial p_i} = -\frac{(1-p_i)^2}{4} p_i^{\frac{-(1-p_i)^2}{4}-1} = \nabla K\left(p_i\right) \tag{9}$$

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3291650

Based on Eq.7, Eq.8 and 9 the gradient backpropagation from the classification part is represented as Eq.10.

$$\begin{aligned}\frac{\partial H^{i}_{3D-Har}}{\partial p_i} &= \left(1+e^{-L_{reg}}\right)J\left(p_i\right)+\\ L_{reg}.\nabla K\left(p_i\right) &- L_{dir}\left(p_i^{'},p_i^{'gt}\right).\frac{\nabla K(p_i)}{\beta_{dir}}\end{aligned} \tag{10}$$

Note that, in our experiment $\beta_{dir}=2$, the gradient backpropagation from the classification result is highly associated to

$$\begin{array}{ccccc} L_{reg} & & p_i & & L_{dir} \\ \uparrow & , & \uparrow & , & \uparrow \\ regression & & classification & & direction\ estimation \end{array}$$

Analysis-1: As a result, Fig.2(b) depicts the outcomes of Eq.10 obtained by sampling ten thousand data on average. The intensity of the color corresponds to the value of $\partial H^{i}_{3D-Har}/\partial p_i$ for the corresponding $[p_i, L_{reg}, L_{dir}]$, with the three axes representing the values of L_{reg}, p_i, and L_{dir}. Similarly, Fig.2(a) represents $\partial H^{i}_{3D}/\partial p_i$ using the same axis representation. Note: the backpropagation gradient from the classification section is independent of the regression and direction estimation parts, as can be observed in Fig.2(a) (with a better view in its vertical representation, Fig.2(e)). When using the 3D harmonic loss (best viewed in Fig.2(f)), the high regression loss suppresses the gradient from the classification loss (due to poor localization), resulting in relatively low confidence, which establishes mutual consistency between classification and localization. Furthermore, the L_{dir} gradually influences the gradient propagation to achieve a globally unique optimization, where $\partial H^{i}_{3D}/\partial p_i=0$ occurs only when $p_i=1$, $L_{dir}=0$, and $L_{reg}=0$.

Proof-2: The effectiveness analysis of 3D harmonic loss on regression part is derived as follows

$$\begin{aligned}\frac{\partial L^{i}_{3D-Har}}{\partial \Delta d_i} &= -e^{-L_{reg}(\Delta d_i)}L_{cls}\left(\frac{\partial L_{reg}(\Delta d_i)}{\Delta d_i}\right)\\ &+\left(1+e^{-L_{cls}}\right)\left(\frac{\partial L_{reg}(\Delta d_i)}{\partial \Delta d_i}\right)\\ &+\frac{e^{-L_{reg}(\Delta d_i)}\left(\frac{\partial L_{reg}(\Delta d_i)}{\Delta d_i}\right)}{2}\cdot L_{dir}\end{aligned} \tag{11}$$

The gradient back propagation from regression result is highly associated to

$$\begin{array}{ccccc} \Delta d_i & & L_{cls} & & L_{dir} \\ \uparrow & , & \uparrow & , & \uparrow \\ regression & & classification & & direction\ estimation \end{array}$$

Analysis-2: Fig.2(d) depicts the results of Eq.11 using ten thousand data samples. The intensity of color on the graph reflects the value of $\partial H^{i}_{3D-Har}/\partial \Delta d_i$ for corresponding $[\Delta d_i, L_{cls}, L_{dir}]$, with the three axes representing the values of Δd_i, L_{reg}, and L_{dir}, respectively. Fig.2(c) shows the corresponding $\partial H^{i}_{3D}/\partial \Delta d_i$ with the same axis representation. In traditional 3D detection learning, the regression part's gradient backpropagation is independent of classification and direction estimation. Even though our proposed method achieved the same regression result (i.e., the same Δd_i), increasing classification loss will consistently restrict the gradient from maintaining synchronous learning of classification and regression (as shown in Fig.2(h)). The global unique optimization of $\partial H^{i}_{3D}/\partial \Delta d_i=0$ is achieved only when $\Delta d_i=0$, $L_{cls}=0$, and $L_{dir}=0$.

Proof-3: The effectiveness analysis of 3D harmonic loss on the direction part is derived as follows.

$$L_{dir}\left(p_i^{'}\right)=\left(1-p_i^{'gt}\right)\log\left(1-p_i^{'}\right)-p_i^{'gt}log\left(p_i^{'}\right) \tag{12}$$

Such that

$$\frac{\partial L_{dir}\left(p_i^{'}\right)}{\partial p_i^{'}}=-\frac{p_i^{'gt}-p_i^{'}}{\left(1-p_i^{'}\right)p_i^{'}}=\mu\left(i\right)$$

Based on the $p_i^{'gt}$ status, the binary cross entropy loss is updated as follows

$$\mu\left(i\right)=\begin{cases}-\frac{1}{p_i^{'}},\ \ if\ \ p_i^{'gt}=1\\ -\frac{1}{1-p_i^{'}},\ \ if\ \ p_i^{'gt}=0\end{cases} \tag{13}$$

The gradient from the updated direction loss is as follows.

$$\begin{aligned}\frac{\partial L^{i}_{3D-Har}}{\partial p_i^{'}} &= \left(1-\frac{\beta_r+\beta_c}{2}\right)\cdot\frac{\partial L^{i}_{3D-Har}}{\partial p_i^{'}}\\ &= \left(1-\frac{\beta_r+\beta_c}{2}\right)\cdot\left[-\frac{p_i^{'gt}}{p_i^{'}}-\frac{\left(1-p_i^{'gt}\right)}{1-p_i^{'}}\right]\end{aligned} \tag{14}$$

The type of direction loss estimation is dependent on the $p_i^{'gt}$ status, and it is derived as follows.

$$\frac{\partial L^{i}_{3D-Har}}{\partial p_i^{'}}=\begin{cases}\left(1-\frac{e^{-L_{cls}}+e^{-L_{reg}}}{2}\right)\left(-\frac{1}{p_i^{'}}\right),\ if\,p_i^{'gt}=1\\ \left(1-\frac{e^{-L_{cls}}+e^{-L_{reg}}}{2}\right)\left(-\frac{1}{1-p_i^{'}}\right),\ if\,p_i^{'gt}=0\end{cases} \tag{15}$$

The gradient backpropagation from the direction result is highly associated to

$$\begin{array}{ccccc} L_{reg} & & L_{cls} & & p_i^{'} \\ \uparrow & , & \uparrow & , & \uparrow \\ regression & & classification & & direction\ estimation \end{array}$$

Analysis-3: Fig.3 depicts the outcomes of Eq.15 based on ten thousand data samples. The color intensity in the figure indicates the value of $\partial H^{i}_{3D-Har}/\partial p_i^{'}$ for the corresponding $[p_i^{'}, L_{cls}, L_{reg}]$. The three axes in the figure represent the values of $p_i^{'}$, L_{reg}, and L_{cls}, respectively. For example, in Fig.3(a), the global unique optimization is $\partial H^{i}_{3D}/\partial p_i^{'}=0$ when $p_i^{'gt}=0$ because $L_{cls}=0$ and $L_{reg}=0$ when $p_i^{'}=0$. Similarly, Fig.3(b) illustrates the same intrinsic-consistency paradigm for $p_i^{'gt}=1$.

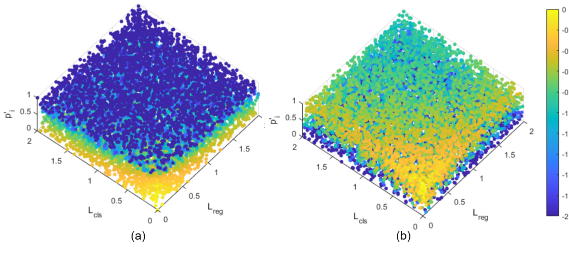


Fig. 3. Visualization of the gradients from direction estimation part in our proposed 3D harmonic loss. (a) when $p_i^{'gt}=0$. (b) when $p_i^{'gt}=1$. MATLAB was utilized to analysis the data and plot the diagram.

The standard approach for 3D detection based on point clouds involves the utilization of three distinct loss functions,

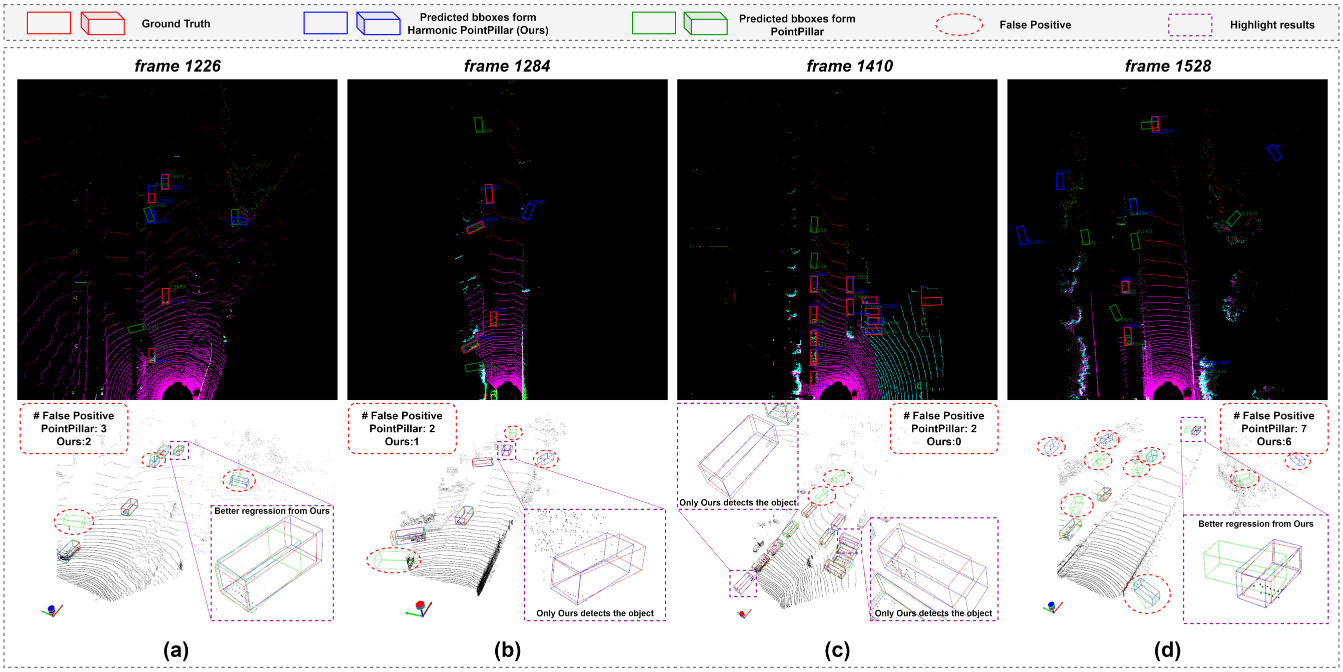

Ground Truth
Predicted bboxes form Harmonic PointPillar (Ours)
Predicted bboxes form PointPillar
False Positive
Highlight results
frame 1226
frame 1284
frame 1410
frame 1528
BEV Visualization
3D Visualization
# False Positive PointPillar: 3 Ours:2
Better regression from Ours
# False Positive PointPillar: 2 Ours:1
Only Ours detects the object
Only Ours detects the object
# False Positive PointPillar: 2 Ours:0
Only Ours detects the object
# False Positive PointPillar: 7 Ours:6
Better regression from Ours
(a)
(b)
(c)
(d)

Fig. 4. Qualitative analysis of overall 3D detection performance. Predicted bboxes from Pointpillar (baseline) (green bboxes) [21] and predicted bboxes from Harmonic Pointpillar (Ours) (blue bboxes) are visualized in same frames. Ground Truths (red bboxes) are also drawn for qualitative check. Harmonic PointPillar (Ours) shows better recall rate and localization accuracy with less false positive than PointPillar (baseline).

namely L_{cls}, L_{reg}, and L_{dir}, which aim to optimize the model's ability to predict the object's category, object 3D position, and object orientation. The three losses have a relatively independent relationship since the model training employs a direct sum of losses. This independence can result in inconsistent predictions. Nonetheless, the 3D harmonic loss, a suggested solution, presents a unified formula that reconciles the three losses. To explain the harmonization mechanism, Proof-1, 2 and 3, as well as Analysis-1, 2 and 3, are provided. By implementing 3D harmonic loss, the three losses become synchronized to enhance the training of the model, leading to simultaneous convergence and reducing prediction inconsistency.

IV. Experiments and Analysis

A. Dataset and evaluation metrics

The performance of the proposed 3D harmonic loss method is assessed using the KITTI dataset [27] and the DAIR-V2X-I dataset [28], both of which contain LiDAR pointcloud data and 3D object annotations. The KITTI dataset includes 7481 training frames and 7518 test frames, which were split into training (3712 frames) and validation (3769 frames) datasets following the approach of previous works [6], [8], [12], [21]. Detection accuracy is evaluated using mean average precision (mAP) with 40 recall positions and Average Orientation Similarity (AOS) as metrics.

The DAIR-V2X dataset, as described in [28], facilitates infrastructure-based 3D object detection experiments by providing a sub dataset called DAIR-V2X-I. This sub dataset consists of 10,000 lidar pointcloud frames obtained from the infrastructure side, containing annotated 3D objects (493k in total) belonging to three categories: car, pedestrian, and cyclist. To conduct our experiments in alignment with those in [28], we utilize the DAIR official toolkit to convert the DAIR-V2X-I dataset to the KITTI data format and employ the same evaluation metrics as those used in the KITTI dataset.

B. Implementation

The experiments in this study were performed on a server equipped with a single NVIDIA GeForce RTX 2080Ti GPU. The KITTI dataset was used to evaluate the effectiveness of the proposed model, and five widely used models (one-stage detectors: PointPillar [21] and SECOND [6], two-stage detectors: PointRCNN [8], Part-A^2 [12]) and PV-RCNN [13]) were adopted as baselines. These models were re-implemented and trained using the mmdetection3D platform [39], while also applying the proposed 3D harmonic loss. Additionally, the models were trained using their original training settings and parameters.

We have named our models Harmonic PointPillar, Harmonic SECOND, Harmonic PointRCNN, Harmonic Part-A^2, and Harmonic PV-RCNN. During the evaluation stage, we kept the post-processing the same as the baselines. We submitted the results of Harmonic PointPillar to the KITTI official benchmark for testing on the KITTI test dataset. In our assessment, we compared the performance of our models to PointPillar (baseline) and other models [7], [9], [11], [14], [31], [33]–[35], [40]–[42], including two-stage lidar-based, one-stage lidar-based, and fusion-based methods. To assess the performance of the proposed model using the DAIR-V2X-I dataset, we used PointPillar [21] and SECOND [6], two widely used one-stage detectors, as baselines. We implemented our models and baselines on the DAIR-V2X official benchmark [28] using the original training parameters and evaluation settings. The

TABLE I
mAP evaluation of BEV object detection on car class of KITTI validation dataset

Method	Type / Modality	IoU threshold: 0.7			IoU threshold: 0.5			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	
PointPillar [21] ⋆	one-stage / LiDAR	N/A	87.70	N/A	N/A	N/A	N/A	N/A
PointPillar [21] †	one-stage / LiDAR	92.09	87.85	83.35	95.72	94.72	90.08	90.64
Harmonic PointPillar (Ours) †	one-stage / LiDAR	94.07	88.41	85.42	95.98	94.87	92.09	91.81
Δ	N/A	**+1.98**	+0.56	**+2.07**	+0.26	+0.15	**+2.01**	+1.17
SECOND [6] ⋆	one-stage / LiDAR	89.96	87.07	79.66	N/A	N/A	N/A	N/A
SECOND [6] †	one-stage / LiDAR	93.47	88.96	86.23	96.60	95.27	92.60	92.18
Harmonic Second (Ours) †	one-stage / LiDAR	95.41	89.23	86.25	98.96	95.63	94.63	93.35
Δ	N/A	**+1.94**	+0.27	+0.02	**+2.36**	+0.36	**+2.03**	+1.17
Point RCNN [8] ⋆	two-stage / LiDAR	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Point RCNN [8] †	two-stage / LiDAR	94.69	88.53	88.14	97.85	94.31	94.15	92.94
Harmonic Point RCNN (Ours) †	two-stage / LiDAR	94.97	88.27	88.02	98.45	94.30	94.01	93.00
Δ	N/A	+0.28	-0.26	-0.12	+0.60	-0.01	-0.14	+0.06
Part-A^2 [12] ⋆	two-stage / LiDAR	90.42	88.61	87.31	N/A	N/A	N/A	N/A
Part-A^2 [12] †	two-stage / LiDAR	92.78	89.47	88.34	96.95	94.17	94.14	92.64
Harmonic Part-A^2 (Ours) †	two-stage / LiDAR	95.00	90.14	88.38	97.93	95.41	94.02	93.48
Δ	N/A	**+2.22**	+0.67	+0.04	**+0.98**	**+1.24**	-0.12	+0.84
PV-RCNN [13] ⋆	two-stage / LiDAR	95.76	91.11	88.93	N/A	N/A	N/A	N/A
PV-RCNN [13] †	two-stage / LiDAR	94.53	90.69	88.62	98.05	96.23	94.38	93.75
Harmonic PV-RCNN (Ours) †	two-stage / LiDAR	94.54	90.72	88.63	98.27	96.36	94.51	93.84
Δ	N/A	+0.01	+0.03	+0.01	+0.22	+0.13	+0.13	+0.09

⋆: reported results in paper, †: our implementation on mmdetection3D [39]. N/A: not available or not applicable. Emphases are highlighted in bold.

TABLE II
mAP evaluation of 3D object detection on car class of KITTI validation dataset

Method	Type / Modality	IoU threshold: 0.7			IoU threshold: 0.5			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	
PointPillar [21] ⋆	one-stage / LiDAR	N/A	77.40	N/A	N/A	N/A	N/A	N/A
PointPillar [21] †	one-stage / LiDAR	87.67	76.44	73.27	95.67	94.41	89.88	86.22
Harmonic PointPillar (Ours) †	one-stage / LiDAR	87.66	77.76	73.44	95.95	94.72	90.12	86.61
Δ	N/A	-0.01	**+1.32**	+0.17	+0.28	+0.31	+0.24	+0.39
SECOND [6] ⋆	one-stage / LiDAR	87.43	76.48	69.10	N/A	N/A	N/A	N/A
SECOND [6] †	one-stage / LiDAR	89.58	79.78	76.49	96.56	95.01	92.45	88.31
Harmonic Second (Ours) †	one-stage / LiDAR	91.16	79.68	76.06	98.88	95.36	92.56	88.95
Δ	N/A	**+1.58**	-0.10	-0.43	**+2.32**	+0.35	+0.11	+0.64
Point RCNN [8] ⋆	two-stage / LiDAR	88.88	78.63	77.38	N/A	N/A	N/A	N/A
Point RCNN [8] †	two-stage / LiDAR	90.99	80.20	77.93	97.81	94.20	93.83	89.16
Harmonic Point RCNN (Ours) †	two-stage / LiDAR	91.77	80.07	77.26	98.41	94.17	93.88	89.26
Δ	N/A	+0.78	-0.13	-0.67	+0.60	-0.03	+0.05	+0.10
Part-A^2 [12] ⋆	two-stage / LiDAR	89.47	79.47	78.54	N/A	N/A	N/A	N/A
Part-A^2 [12] †	two-stage / LiDAR	91.81	82.35	80.16	96.91	94.09	93.95	89.88
Harmonic Part-A^2 (Ours) †	two-stage / LiDAR	91.92	82.43	80.07	97.92	94.01	93.85	90.03
Δ	N/A	+0.11	+0.08	-0.09	**+1.01**	-0.08	-0.10	+0.15
PV-RCNN [13] ⋆	two-stage / LiDAR	92.57	84.83	82.69	N/A	N/A	N/A	N/A
PV-RCNN [13] †	two-stage / LiDAR	92.09	84.52	82.56	98.05	94.51	94.29	91.00
Harmonic PV-RCNN (Ours) †	two-stage / LiDAR	91.91	84.54	82.44	98.21	94.66	94.41	91.01
Δ	N/A	-0.18	+0.02	-0.22	+0.16	+0.15	+0.12	+0.01

⋆: reported results in paper, †: our implementation on mmdetection3D [39]. N/A: not available or not applicable. Emphases are highlighted in bold.

only difference between our models and the baselines is the adoption of the proposed 3D harmonic loss, to ensure a fair comparison.

C. Quantitative analysis

Experimental results with thorough quantitative analysis are reported below.

Detecting cars is a crucial aspect of Intelligent Transportation Systems (ITS) such as V2V and V2X. Our method's ability to detect cars was evaluated in both on-vehicle and roadside settings, with the resulting mAP values shown in Tab.I and Tab.II. Our proposed method outperformed the baseline models in terms of average mAP values, particularly in BEV detection where our models achieved significant mAP rates (at least 0.02% and up to 2.36% better than SECOND, and at least 0.15% and up to 2.07% better than PointPillar). We have also submitted our Harmonic PointPillar model to the official KITTI test benchmark (refer to Tab.III), and its high time efficiency (as indicated in Tab.VIII) makes it a popular choice for industrial applications. Our method has optimized the baseline PointPillar model with an improvement of 0.82%

TABLE III
mAP evaluation of BEV object detection on car class of KITTI test benchmark

Method	Source	Type / Modality	Car (IoU threshold:0.7)		
			Easy	Mod.	Hard
F-PointNet [31]	CVPR 2018	fusion / LiDAR+camera	91.17	84.67	74.77
PointPainting [35]	CVPR 2020	fusion / LiDAR+camera	92.45	88.11	83.36
CLOCs [33]	IROS 2020	fusion / LiDAR+camera	91.16	88.23	82.63
StructuralIF [34]	CVIU 2021	fusion / LiDAR+camera	91.78	88.38	85.67
Point-RCNN [8]	CVPR 2019	two-stage / LiDAR	92.13	87.39	82.72
PI-RCNN [40]	AAAI 2020	two-stage / LiDAR	91.44	85.81	81.00
Part-A^2 [12]	TPAMI 2021	two-stage / LiDAR	91.70	87.79	84.61
Voxel-RCNN [11]	AAAI 2021	two-stage / LiDAR	94.85	88.83	86.13
EQ-PVRCNN [41]	CVPR 2022	two-stage / LiDAR	94.55	89.09	86.42
3DSSD [9]	CVPR 2020	one-stage / LiDAR	92.66	89.02	85.86
Point-GNN [7]	CVPR 2020	one-stage / LiDAR	93.11	89.17	83.90
TANet [42]	AAAI 2020	one-stage / LiDAR	91.58	86.54	81.19
VoxSet [14]	CVPR 2022	one-stage / LiDAR	92.70	89.07	86.29
PointPillar [21]	CVPR 2019	one-stage / LiDAR	90.07	86.56	82.81
Harmonic PointPillar	Ours	one-stage / LiDAR	90.89	87.28	82.54
Δ	N/A	N/A	**+0.82**	**+0.72**	-0.27

Results of listed works were extracted from KITTI BEV test benchmark [27] (Date: 14 August 2022). N/A: not applicable. $\Delta_{average}$ = +0.42. Results worse than Harmonic PointPillar (Ours) are colored in orange. Check our submitted result at https://www.cvlibs.net/datasets/kitti/eval_object_detail.php?&result=cf021462bb1955480c0c5ebe6c1756545bf98566.

TABLE IV
mAP evaluation of BEV object detection on car class of DAIR-V2X-I dataset

Method	Type / Modality	IoU threshold: 0.7			IoU threshold: 0.5			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	
PointPillar [21] †	one-stage / LiDAR	72.01	54.39	54.40	72.40	54.50	54.50	60.36
Harmonic PointPillar (Ours) †	one-stage / LiDAR	71.95	54.42	54.42	72.60	54.51	54.51	60.40
Δ	N/A	-0.06	+0.03	+0.03	**+0.20**	+0.01	+0.01	+0.04
SECOND [6] †	one-stage / LiDAR	72.32	61.97	62.01	72.64	62.36	62.38	65.61
Harmonic Second (Ours) †	one-stage / LiDAR	72.44	63.09	63.08	72.65	63.27	63.28	66.30
Δ	N/A	+0.12	**+1.12**	**+1.07**	+0.01	**+0.91**	**+0.90**	+0.69

DAIR-V2X benchmark [28] does not offer the evaluation results of BEV object detection. All models † are from our implementation on DAIR-V2X dataset [28]. Emphases are highlighted in bold.

TABLE V
mAP evaluation of 3D object detection on car class of DAIR-V2X-I dataset

Method	Type / Modality	IoU threshold: 0.7			IoU threshold: 0.5			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	
PointPillar [21] ⋆	one-stage / LiDAR	N/A	N/A	N/A	63.07	54.00	54.01	N/A
PointPillar [21] †	one-stage / LiDAR	71.33	54.07	54.08	72.30	54.50	54.50	60.13
Harmonic PointPillar (Ours) †	one-stage / LiDAR	71.30	54.15	54.15	72.57	54.51	54.51	60.20
Δ	N/A	-0.03	+0.08	+0.08	**+0.27**	+0.01	+0.01	+0.07
SECOND [6] ⋆	one-stage / LiDAR	N/A	N/A	N/A	71.47	53.99	54.00	N/A
SECOND [6] †	one-stage / LiDAR	71.14	53.92	53.93	72.61	62.29	62.31	62.70
Harmonic Second (Ours) †	one-stage / LiDAR	71.57	54.12	54.13	72.63	63.28	63.28	63.17
Δ	N/A	**+0.43**	**+0.20**	**+0.20**	+0.02	**+0.99**	**+0.97**	+0.47

⋆: reported results on DAIR-V2X benchmark [28]. †: our implementation on DAIR-V2X dataset [28]. N/A: not available or not applicable. Emphases are highlighted in bold.

on Easy samples and 0.72% on Moderate samples, with only a 0.27% decrease on Hard samples. However, due to our method's focus on balancing and harmonizing the gradient from different parts, the classification confidence for hard samples is usually low, as they typically consist of very sparse points scanned from objects, leading to a drop in mAP due to suppression of the regression part. Furthermore, extremely hard samples, such as outliers, can adversely affect the model's stability due to their large gradient variance.

The evaluation results for the DAIR-V2X-I dataset are presented in Table IV and Table V. Our model has achieved the average improvement of at least 0.04% and at most 0.69%. As the majority of current 3D car detection models based on LiDAR were developed and tested from an on-vehicle LiDAR perspective, our future work will focus on developing more effective 3D detection techniques specifically tailored for on-infrastructure LiDAR.

Direction estimation: The performance of 3D direction

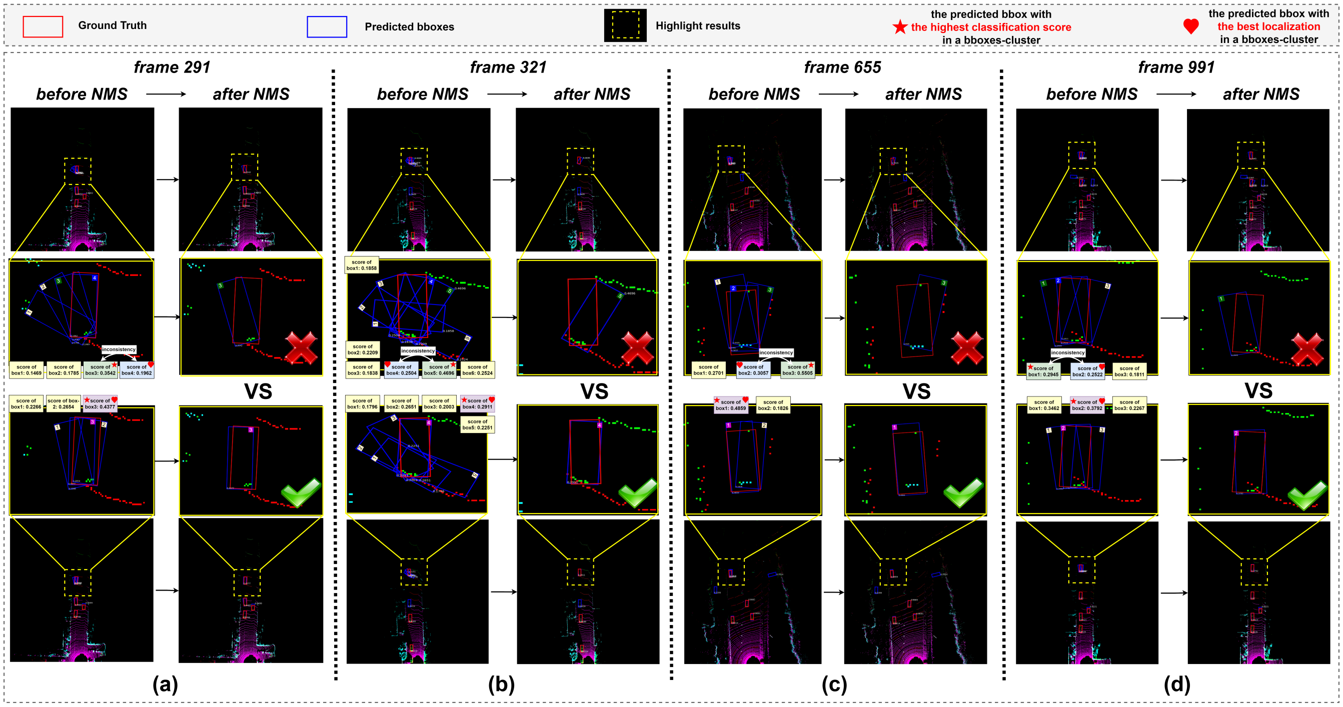


Fig. 5. Qualitative analysis of inconsistent/consistent 3D detection. For better view, the results are in BEV visualization (zoom in for detail check). PointPillar (baseline) [21] suffers from inconsistency problem in 3D detection, while Harmonic PointPillar (Ours) shows a great robustness on keeping predictions consistent.

is evaluated using the average orientation similarity (AOS) index. Tab.VI presents the AOS evaluation results for different IoU thresholds (0.7 and 0.5). A higher AOS value indicates better direction estimation for 3D objects. Our proposed model achieved an average improvement of 0.14% from PV-RCNN to Harmonic PV-RCNN, 0.26% from PointPillar to Harmonic PointPillar, and 0.76% from SECOND to Harmonic SECOND. Specifically, our proposed strategy significantly improved the performance of vanilla models on easy-level objects (improvement ranging from at least 0.23% to up to 1.99% under 0.7 IoU threshold, and at least 0.22% to up to 1.76% under 0.5 IoU threshold) and hard-level objects (improvement ranging from at least 0.16% to up to 1.52% under 0.5 IoU threshold). These results demonstrate that our proposed method can more accurately estimate the direction of objects.

TABLE VI
Average Orientation Similarity (AOS) evaluation of one-stage 3D/BEV object detection on car class of KITTI validation dataset and KITTI test benchmark

Method	IoU threshold: 0.7			IoU threshold: 0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard
PV-RCNN [13]◇	97.96	94.32	93.82	98.17	96.62	94.42
Harmonic PV-RCNN (Ours)◇	98.19	94.34	93.86	98.41	96.78	94.58
Δ	+0.23	+0.02	+0.04	+0.24	+0.16	+0.16
SECOND [6]◇	96.27	92.13	88.94	96.50	94.95	91.84
Harmonic SECOND (Ours)◇	98.26	92.00	88.56	98.69	94.77	93.36
Δ	**+1.99**	-0.13	-0.38	**+1.76**	-0.18	**+1.52**
PointPillar [21]◇	95.31	91.42	86.51	95.71	94.38	89.33
Harmonic PointPillar (Ours)◇	95.66	91.33	86.21	95.93	94.28	90.83
Δ	+0.35	-0.09	-0.30	+0.22	-0.10	**+1.50**
PointPillar [21]∗	93.84	90.70	87.47	N/A	N/A	N/A
Harmonic PointPillar (Ours)∗	94.23	90.78	87.42	N/A	N/A	N/A
Δ	+0.39	+0.08	-0.05	N/A	N/A	N/A

◇: results on KITTI validation dataset. ∗: results on KITTI test benchmark. N/A: not applicable. Emphases are highlighted in bold.

Vulnerable road users detection: In addition to car detection, detecting vulnerable road users such as pedestrians and cyclists is essential for enhancing the security monitoring capabilities of V2X applications. Based on our observations and analysis, the scanned pointcloud shapes of pedestrians and cyclists are more irregular with varying postures, which poses a significant challenge for optimizing the detection models. Tab.VII presents a comparison of mAP scores for detecting pedestrians and cyclists using the official 0.5 IoU threshold. Our proposed method achieved better synchronous learning rates for classification, localization, and direction estimation (related to object shape). Compared to the base models, our method significantly improved the accuracy of pedestrian and cyclist detection, with maximum mAP improvements of 1.39% and 0.49%, respectively. These results demonstrate that our proposed method is highly reliable in promoting the 3D detection of vulnerable traffic objects.

TABLE VII
mAP evaluation of 3D object detection on pedestrian/cyclist class of KITTI validation dataset

Method	3D Pedestrian			3D Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND [6]	61.59	54.27	48.03	83.52	65.04	60.98
Harmonic SECOND (Ours)	62.41	55.46	48.93	83.78	64.90	61.07
Δ	**+0.82**	**+1.19**	**+0.90**	+0.26	-0.14	+0.09
PointPillar [21]	52.49	46.47	41.52	75.94	59.19	55.89
Harmonic PointPillar (Ours)	52.31	46.31	41.33	77.33	60.49	56.48
Δ	-0.18	-0.16	-0.21	**+1.39**	**+1.30**	+0.59

Results are from our implementation on mmdetection3D [39]. IoU threshold: 0.5. $\Delta_{average}$ = +0.49. Emphases are highlighted in bold.

Time efficiency of 3D detection matters in V2X applications. Fast 3D detection models are necessary for roadside units (RSUs) to capture vehicle signals in real-time. Tab.VIII presents a runtime comparison of various state-of-the-art 3D detectors. On average, our proposed method, like PointPillar [21], is at least 1.5 times faster than other methods since it works as a training optimizer and does not cause delays in detection inference. This measurement confirms that our method is highly time-friendly. **Time efficiency** of 3D detection matters in V2X applications.

TABLE VIII
Average runtime comparison of 3D/BEV object detection

Method	Source	Type / Modality	Speed (Hz)
Point-RCNN [8]	CVPR 2019	two-stage / LiDAR	2.7
Part-A^2 [12]	TPAMI 2021	two-stage / LiDAR	9.5
CenterPoint [17]	CVPR 2021	two-stage / LiDAR	39.2
SECOND [6]	Sensors 2018	one-stage / LiDAR	18.0
3DSSD [9]	CVPR 2020	one-stage / LiDAR	10.9
Point-GNN [7]	CVPR 2020	one-stage / LiDAR	3.3
TANet [42]	AAAI 2020	one-stage / LiDAR	29.4
VoxSet [14]	CVPR 2022	one-stage / LiDAR	24.2
PointPillar [21]	CVPR 2019	one-stage / LiDAR	**43.1**
Harmonic PointPillar	Ours	one-stage / LiDAR	**43.1**

Inference speed was tested on Pytorch with single GPU 2080Ti.

D. Qualitative analysis

Visualized results along with detailed qualitative analysis are presented below, depicting the overall performance of 3D detection.

Overall performance: A comparison of the overall performance is illustrated in Fig.4. In Fig.4(a) and Fig.4(d), the Harmonic PointPillar model outperforms the baseline model (PointPillar) in terms of localization accuracy. On the other hand, in Fig.4(b) and Fig.4(c), our model detects more valid objects, which were missed by the baseline models. Additionally, our model has a lower false positive (FP) ratio than the baseline model in all example frames.

Dealing with inconsistency problem: To further confirm the effectiveness of the proposed method in resolving inconsistency issues in 3D detection, we present a more detailed qualitative visualization in Fig.5, where viewers can zoom in for a closer inspection. The baseline model (PointPillar) failed to predict the targets in all example frames due to inconsistency between classification and localization. In contrast, our model demonstrated remarkable robustness in maintaining consistent 3D detection, resulting in the most accurate predictions in all example frames. This confirms that our method can construct a task-consistent 3D detector.

E. Simulations on realistic deployment

We utilized our previous experience with PyTorch-style Harmonic PointPillar [26] to convert it into TensorRT-format for deployment. The converted model was deployed on Jetson Xavier TX using float16 quantization techniques, and the same experiments as on PC were conducted. The results show a notable 2x-speed improvement (75.4Hz on Jetson Xavier TX vs 43.1Hz on PC Single 2080Ti) with at most a 1% mAP drop. The Jetson results demonstrate that our proposed method is feasible for edge orchestration due to its consistent, continuous trade-off between time efficiency and model accuracy with low energy consumption. Fig.6 presents a qualitative example of on-infrastructure detection using TensorRT-format Harmonic PointPillar on Jetson Xavier TX.

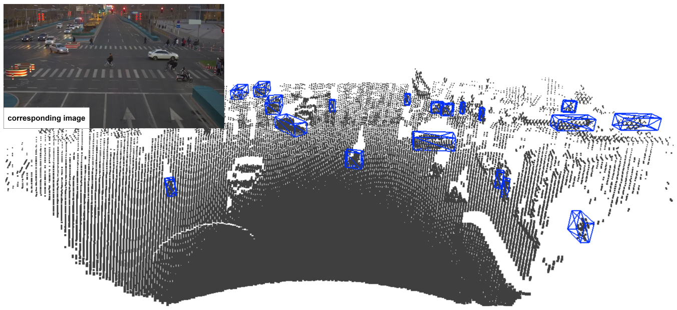


Fig. 6. Qualitative example of on-infrastructure LiDAR-based 3D object detection (detection results (blue bboxes) are made by our proposed method: Harmonic PointPillar).

V. Conclusion

In this paper, we propose a method to address the inconsistency problem in 3D object detection and achieve better results compared to state-of-the-art methods. Our simulations demonstrate that our proposed method is effective in strengthening V2X frameworks. We first analyze the causes of inconsistency among classification, localization, and direction estimation and derive theoretical and mathematical solutions. Second, we introduce the 3D harmonic loss function, which effectively resolves the inconsistency problem in the point cloud domain and achieves higher mAP with a deployment speed of 75.4 Hz, surpassing baseline models. Mathematical derivatives are provided to support the effectiveness of our proposed loss mechanism. Our comprehensive experiments demonstrate that our proposed method significantly improves detection accuracy without incurring extra inference time cost. In the future, we plan to focus on improving on-infrastructure detection.

Acknowledgments

This work was supported in part of Basic Science Research Programs of the Ministry of Education (NRF-2018R1A2B6005105) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2019R1A5A8080290).

References

[1] C. E. Palazzi, M. Roccetti, and S. Ferretti, “An intervehicular communication architecture for safety and entertainment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 90–99, 2009.

[2] X. Li, W. Wang, and Wu, “Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 002–21 012, 2020.

[3] X. Li and W. Wang, “Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 632–11 641.

[4] K. Wang and L. Zhang, "Reconcile prediction consistency for balanced object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3631–3640.

[5] K. Kim and H. S. Lee, "Probabilistic anchor assignment with iou prediction for object detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 355–371.

[6] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[7] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1711–1719.

[8] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[9] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 040–11 048.

[10] Y. Zhou and O. 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*, 2018, pp. 4490–4499.

[11] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1201–1209.

[12] S. Shi and Z. Wang, "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*, vol. 43, no. 8, pp. 2647–2664, 2021.

[13] S. Shi and C. Guo, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[14] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3d object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8417–8427.

[15] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "Se-ssd: Self-ensembling single-stage object detector from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 494–14 503.

[16] J. S. Hu, T. Kuai, and S. L. Waslander, "Point density-aware voxels for lidar 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8469–8478.

[17] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[18] T. Guan, J. Wang, S. Lan, R. Chandra, Z. Wu, L. Davis, and D. Manocha, "M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 772–782.

[19] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "Cia-ssd: Confident iou-aware single-stage object detector from point cloud," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 4, 2021, pp. 3555–3562.

[20] C. Lin and D. Tian, "Cl3d: Camera-lidar 3d object detection with point feature enhancement and point-guided fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[21] A. H. Lang and S. Vora, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[22] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[23] S. Mohapatra and S. Yogamani, "Bevdetnet: bird's eye view lidar point cloud based real-time 3d object detection for autonomous driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2809–2815.

[24] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9775–9784.

[25] Y. Zeng, Y. Hu, and S. Liu, "Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[26] L. Stäcker and J. Fei, "Deployment of deep neural networks for object detection on edge ai devices with runtime optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1015–1022.

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[28] H. Yu and Luo, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 361–21 370.

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[30] J. Mao, Y. Xue, and M. Niu, "Voxel transformer for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3164–3173.

[31] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[32] H. Zhang, D. Yang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner, "Faraway-frustum: Dealing with lidar sparsity for 3d object detection using fusion," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2646–2652.

[33] 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)*. IEEE, 2020, pp. 10 386–10 393.

[34] P. An, J. Liang, K. Yu, B. Fang, and J. Ma, "Deep structural information fusion for 3d object detection on lidar–camera system," *Computer Vision and Image Understanding*, vol. 214, p. 103295, 2022.

[35] S. Vora and A. H. Lang, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[37] ——, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[39] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d, 2020.

[40] L. Xie, C. Xiang, and Yu, "Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 460–12 467.

[41] Z. Yang, L. Jiang, Y. Sun, B. Schiele, and J. Jia, "A unified query-based paradigm for point cloud understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8541–8551.

[42] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 677–11 684.

Haolin Zhang received his B.E. degree in Electronics Information Engineering from Shanghai University of Electric Power, Shanghai, China, and the M.S. degree in Electrical and Computer Engineering from The Ohio State University, Columbus, OH, USA. He is currently a full-time research engineer at Xi'an Jiaotong University, Xi'an, Shaanxi, China. His research interests include deep learning, computer vision, robotics, intelligent vehicle, intelligent transportation system, and V2X.

M. S. Mekala (Senior Member, IEEE, AFHE) received the Ph.D. degree from VIT University. He is currently working as an Assistant Professor with the School of Computing, Robert Gordon University, Aberdeen, U.K. He is a Former Post-Doctoral Researcher at RLRC LAB, Yeungnam University, gyeonnsan, Korea, and a Research Coordinator and Member of FEB Laboratory at KL University. He has published more than 20 peer-review research papers (indexed in SCI-SCIE). His research interests include service computing, intelligent machine vision, data communication, decision making system design, edge computing, CPS, IoT communication, and Reliability Analysis. He is serving as the Guest Editor of more than two special issues in various peer-reviewed journals. He is a recipient of Best Research award for two consecutive years in 2018 and 2019. He received the research scientist award in 2020. He is a Former-GE of IEEE TRANSACTION ON SERVICE COMPUTING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTION ON ROBOTICS, Measurement-Elsevier, computer networks, neural computing and applications, Applied soft-computing Methods.

Dongfang Yang (Member, IEEE) received his B.E. degree in microelectronics from Sun Yat-sen University, Guangzhou, China, in 2014 and the M.S. and Ph.D. degrees in Electrical and Computer Eng. from The Ohio State University, Columbus, OH, USA, in 2019 and 2020, respectively. He is currently a senior algorithm engineer at Chongqnig Chang'an Automobile Co., Ltd. and a postdoc researcher at Chongqing University in Chongqing, China. His research interests include data analysis, machine learning, deep learning, and control, with applications in behavior prediction, decision-making, and motion planning of autonomous systems.

John Isaacs received the Ph.D. degree from Abertay University, UK. currently serves as the Dean of the School of Computing at Robert Gordon University in Aberdeen, U.K. Additionally, he is recognized as a Fellow of UK Higher Education. He has founded the Digital Innovation Lab with the objective of offering constructive solutions for the 3D Visualization community. Furthermore, he is a highly regarded Principal Investigator for more than three projects. In the past six years, he has been honored with several teaching excellence awards, including the Continued Excellence Award in 2020, the Extracurricular Award in 2019, the Personal Tutor of the Year in 2018, and the Continued Excellence Award in 2017. His research papers, indexed in SCI-SCIE, have been published extensively. His research interests encompass software design and development, the development of custom 3D Lidar engine/framework, intricate 3D Lidar data visualization, and computational modeling.

Zulqar Nain received his M.S. degree from COMSATS University Islamabad in 2018 and his Ph.D. degree in information and communication engineering from Yeungnam University in 2022. Currently, he is working as an assistant professor with AI and Big Data Department at Woosong university. His research interests include routing in NoC, fault-tolerant routing in NoC, IoT, machine learning, and wireless NoCs

Ju H. Park (Senior Member, IEEE) received the Ph.D. degree in electronics and electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea, in 1997. From May 1997 to February 2000, he was a Research Associate in Engineering Research Center-Automation Research Center, POSTECH. He joined Yeungnam University, Kyongsan, Republic of Korea, in March 2000, where he is currently the Chuma Chair Professor. He has published a number of articles in these areas. His research interests include robust control and filtering, neural/complex networks, fuzzy systems, multiagent systems, and chaotic systems. Since 2015, he has been a recipient of the Highly Cited Researchers Award by Clarivate Analytics (formerly, Thomson Reuters) and listed in three fields, Engineering, Computer Sciences, and Mathematics, in 2019 to 2022. He is a Subject Editor, Advisory Editor, Associate Editor and Editorial Board Member of several international journals, including IET Control Theory and Applications, Applied Mathematics and Computation, Journal of The Franklin Institute, Nonlinear Dynamics, Engineering Reports, Cogent Engineering, the IEEE TRANSACTION ON FUZZY SYSTEMS, the IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the IEEE TRANSACTION ON CYBERNETICS. He is a fellow of the Korean Academy of Science and Technology (KAST).

Yo-Houl Jung received the Ph.D. degree in electronics engineering from the INSA de Lyon (Institute National des Sciences Appliquées de Lyon), France, in 1998. He is currently a Professor with the Department of Information and Communication Engineering, Yeungnam University, Korea. His Teaching and research interests include digital signal processing, computer vision, deep-learning, autonomous vehicle, computer graphics, control signal processing, and IoT.