Abstract—Edge computing-based 3D perception has received attention in intelligent transportation systems (ITS) because real-time monitoring of traffic candidates potentially strengthens Vehicle-to-Everything (V2X) orchestration. Thanks to the capability of precisely measuring the depth information on surroundings from LiDAR, the increasing studies focus on lidar-based 3D detection, which significantly promotes the development of 3D perception. Few methods met the real-time requirement of edge deployment because of high computation-intensive operations. Moreover, an inconsistency problem of object detection remains uncovered in the pointcloud domain due to large sparsity. This paper thoroughly analyses this problem, comprehensively roused by recent works on determining inconsistency problems in the image specialisation. Therefore, we proposed a 3D harmonic loss function to relieve the pointcloud based inconsistent predictions. Moreover, the feasibility of 3D harmonic loss is demonstrated from a mathematical optimization perspective. The KITTI dataset and DAIR-V2X-I dataset are used for simulations, and our proposed method considerably improves the performance than benchmark models. Further, the simulative deployment on an edge device (Jetson Xavier TX) validates our proposed model’s efficiency.

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). V2X and ADS are multi-disciplinary research domains where the vehicles and infrastructures collect and analyze the data [1] to make the vehicle move through an intelligent decision-making system. The decision-making system fed the captured data (surrounding areas, including road structure and traffic information). The data is analyzed to make necessary decisions for the vehicle move based on cloned effective object detection methods through Road Side Units (RSU) and On Board Units (OBU) to identify and localize the traffic candidates.

Motivation: Let us consider a vehicle movement based on event-trigger analysis using traffic information. In this process, the traffic data collects through surveillance devices or on-vehicle sensors such as LiDAR and cameras. The important lidar data (pointcloud) is being analyzed by edge devices to accomplish the target with low latency. However, the computation-intensive services are being offloaded to the server to meet the application deadline. A lidar 3D object detection technology has become prominent because of its low price, increased perception of distant objects, and robust characteristics. For pinpoint communication from vehicle to vehicle (V2V) and vehicle to infrastructure (V2I), the recognized and localized vehicle has become an essential factor in measuring the surroundings and infrastructure through RSU-LiDAR deployment. Thus, developing and deploying an efficient and robust 3D detector based on lidar data is an essential and worthy research direction for strengthening V2X.

Problem of task inconsistency and time delay: Modern object detection tasks have diverged into several sub-tasks (e.g. object localization, classification, direction estimation, etc.). In the 2D image domain, most 2D detection considers sub-tasks (classification and localization) independently, resulting in the output of inconsistently unexpected predictions with high classification confidence but insufficient localization after post-processing (e.g. Non-Maximum Suppression), as shown in Fig.1(a). Such inconsistency problem in 2D object detection has recently been fully discussed and partially solved in [2]–[5]. Turning to the pointcloud domain, the guesswork and similar inconsistency issues remain effects the 3D detection accuracy, as shown in Fig.1(b). Real-data experiments further confirmed our conjecture in Fig.1(c). Most recent lidar-based 3D object detection methods [6]–[17] concentrated on achieving the best mAP and treated it as a phenomenal benchmark for representing model accuracy. Nevertheless, time consumption, quality of experience (QoE), and service reliability factors are essential to showcase the model performance, which has not been addressed. Especially for real-time applications, as in V2X, there is a scope to design a cost-effective, task-friendly and task-consistent detection solution with fast runtime and low error rate. On the one side, several researchers [18], [19] noticed the problem of task inconsistency. Nevertheless, their solutions relied on additional modules with extra inference time-cost, contrary to our application target of pursuing less computational burden. On the other side, some...
recent works [20]–[25] attempted to improve the performance of deployment metrics such as computational burden and execution latency. However, these methods have yet to achieve an adequate trade-off between detection accuracy and time consumption towards edge device-based simulations for 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 [18], [19], 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 [26] and DAIR-V2X-I Dataset [27] 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 [20] 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

3D object detection has become more popular due to pointcloud-based deep learning models through various frameworks. Usually, there are two types of frameworks (one-stage and two-stage). One-stage methods predict the 3D bounding boxes (bboxes) of objects instantly. Some of the methods are points-based; such as 3DSSD [9] is designed based on PointNet [28] architecture, and PointGNN network [7] is developed based on graph neural network. Where raw lidar pointcloud is fed to the learning model for 3D shape predictions. For instance, SECOND [6] is a improved time efficiency approach which is based on VoxelNet [10] by proposing sparse 3D
convolutions. Nevertheless, the time performance of one-stage 3D detection is not desirable enough. Besides, VoTr [29] and VoxSeT [14] not only adopt a voxel-based one-stage method but also introduce transformer [30] architecture for better accuracy. However, heavy parameters and complicated operations significantly impair the time performance of 3D detection. Compared to the methods above, PointPillar [20] transforms 3D pointcloud to 2D voxels, followed by highly efficient 2D convolutions to achieve real-time performance and easy deployment of 3D detection [25]. By contrast, two-stage detectors [8], [11]–[13], [17] predict the Region-of-Interests (ROIs) in the first stage; the refinement network server precisely detects the objects based on ROIs in the second stage. However, most two-stage methods cannot satisfy the speed needed in real-time applications. Although several methods, e.g. CenterPoint [17], has the advantage of fast feature encoding and the light refinement head in their network design. However, the research gap (time-cost comparison) remains similar to some one-stage detectors such as PointPillar [20].

The fusion of image and pointcloud [31]–[33] is an adequate mechanism to achieve better surrounding perception and accuracy for 3D detection. However, the fusion methods are more complex and consume abnormal time in real-time applications than pure lidar-based detection. Therefore, the fusion methods are not mainly discussed but we considered some fusion methods in our experiment to showcase the accuracy accomplishments.

B. Inconsistency problem in object detection

The inconsistency problem was first discovered in 2D object detection methods of the image domain, as shown in Fig.1(a). Early 2D object detection methods independently treat object classification and object localization (two sub-tasks of 2D object detection) in the model training phase, which causes inconsistent predictions in the inference phase. Recent works [2]–[5] focused on this problem in the image domain and tried to bridge the gap between different sub-tasks of 2D detection. In [2], [3], a Generalized Focal Loss method is developed, which is not reached the targeted accuracy, and an improved version of the Focal Loss method is developed in [5] to guarantee coherent 2D detection. In [5], a PAA method is introduced with an additional module to predict IoU for the rational selection of positive training samples. The inconsistency problem in 3D pointcloud object detection mechanisms causes low object detection reliability and quality of experience. For instance, several works [18], [19] attempted to solve it. However, their solutions, similar to PAA [5], consume extra time-cost for a supplementary branch of predicting IoU and additional operations in post-processing, which are not applicable for real-time environments. Our proposed solution deals with the inconsistency problem well in the pointcloud domain without introducing any additional burden for model inference and deployment.

III. PROPOSED WORK

In this section, the proposed method 3D harmonic loss is formulated in both theoretical and mathematical optimization perspective.

Fig.1(d) shows our goal to achieve consistent predictions in 3D detection. From revisiting the learning loss function (Eq.[1]) for a positive training sample i, in many extant methods [6], [8], [12], [20], the cause of inconsistency problem is excavated: three sub-tasks (classification, localization (regression) and direction estimation) of 3D object detection are treated and supervised independently.

$$L_{3D} = L_{cls}(p_i, p_i^{gt}) + L_{reg}(d_i', d_i^{gt}) + L_{dir}(p_i', p_i'^{gt})$$

(1)

Where \(p_i\) is softmax classification score, \(p_i'^{gt}\) 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

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

(2)

In continuation, the regression loss \(L_{reg}\) uses SmoothL1 [38] as follows

$$L_{reg}(d_i', d_i^{gt}) = \sum_{\Delta d_i \in \{0.5 \Delta d_i, |\Delta d_i| - 0.5 \text{ others}\}} \text{SmoothL1}(\Delta d_i)$$

(3)

$$\text{SmoothL1}(\Delta d_i) = \begin{cases} 0.5 \Delta d_i & \text{if } |\Delta d_i| < 1 \\ |\Delta d_i| - 0.5 & \text{others} \end{cases}$$

(4)

Where \(\Delta d_i\) 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 \((\tilde{X}_i, Y_i, Z_i, L_i, W_i, H_i, \alpha_i)\) of anchor boxes as follows

\[
\begin{align*}
    x_i^{gt} &= \frac{X_i^{gt} - X_i}{\sqrt{(W_i)^2 + (L_i)^2}}, \quad y_i^{gt} = \frac{Y_i^{gt} - Y_i}{\sqrt{(W_i)^2 + (L_i)^2}}, \\
    z_i^{gt} &= \frac{Z_i^{gt} - Z_i}{H_i}, \quad w_i^{gt} = \frac{W_i^{gt}}{w_i}, \quad l_i^{gt} = \log \frac{L_i^{gt}}{l_i}, \\
    h_i^{gt} &= \log \frac{H_i}{h_i}, \quad \theta_i^{gt} = \sin (\alpha_i^{gt} - \alpha_i)
\end{align*}
\]

(5)

Inconsistent treatment towards the different sub-tasks leads to inconsistent inference results. Previous work [4] provides a good prior attempt. However, it only focused on 2D detection with the image source and only considered generalizing basic loss functions (e.g. cross-entropy loss, L1 loss, and IoU loss, etc.). For lidar-based 3D detection, with different data modalities (pointcloud) and more dimensions of prediction (direction estimation, height and depth, etc.), we proposed 3D harmonic loss (see Theorem). Three dynamic factors \(1 + \beta_r, 1 + \beta_d, 1 - \frac{\beta_r + \beta_d}{\beta_{dir}}\) are used to make certain that the model learning is conducted in a way of consistency, where \(1 + \beta_r\) and \(1 + \beta_d\) work together to ensure mutual-consistency, and \(1 - \frac{\beta_r + \beta_d}{\beta_{dir}}\) guarantees intrinsic-consistency. Mutual consistency ensures when classification optimisation is inadequate, and the factor derived from the classification part will supervise the regression part and vice versa. Intrinsic consistency guarantees that the direction estimation part is always supervised by both classification and regression...
We consider \( \beta \) supervised by classification loss which is derived as follows.

\[
\text{loss} = (1 + \beta_c) \times L_{\text{cls}}(p_i, p_i^{gt}) + (1 + \beta_r) \times L_{\text{reg}}(d_i, d_i^{gt}) + (1 - \beta_r) \beta_{\text{dir}} X \times L_{\text{dir}}(p_i, p_i^{gt})
\]

Where

\[
\beta_r = e^{-L_{\text{reg}}(d_i, d_i^{gt})}, \beta_c = e^{-L_{\text{cls}}(p_i, p_i^{gt})}
\]  

We consider \( \beta_{\text{dir}} = 2 \) in our 3D harmonic loss simulation. When the training sample is positive, it is consistently supervised by classification loss \( L_{\text{cls}}(p_i, p_i^{gt}) \), regression loss \( L_{\text{reg}}(d_i, d_i^{gt}) \), and direction loss \( L_{\text{dir}}(p_i, p_i^{gt}) \). The following proofs are mathematically derived and explained briefly to represent the effectiveness of 3D harmonic loss.

**Proof-1:** Let us assume, the \( i^{th} \) training sample is positive, then \( p_i^{gt} = 1 \), and \( \alpha = 0.25 \), \( \gamma = 2 \) are for the focal loss to analyse the effectiveness of 3D harmonic loss towards classification loss which is derived as follows

\[
\frac{\partial L_{\text{cls}}(p_i, p_i^{gt})}{\partial p_i} = \frac{\partial (-\alpha p_i^{gt} (1-p_i)^{1-2} \log(p_i))}{\partial p_i} = -\frac{(1-p_i)(1-2 \log(p_i))}{4} \quad \text{suppose} \quad J(p_i)
\]

\[
\beta_c = e^{-L_{\text{cls}}(p_i, p_i^{gt})} = e^{-\frac{1}{4}(1-p_i)^2 \log(p_i)} \quad K(p_i)
\]  

with the point derivation

\[
\frac{\partial \beta_c}{\partial p_i} = -\frac{(1-p_i)^2}{4} p_i \quad \text{suppose} \quad \frac{-\beta_c}{\beta_{\text{dir}}}
\]

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

\[
\frac{\partial H_{3D-Har}}{\partial p_i} = (1 + e^{-L_{\text{reg}}}) \frac{J(p_i)}{L_{\text{reg}}} + L_{\text{reg}} \frac{\nabla K(p_i)}{L_{\text{dir}}} - L_{\text{dir}} \frac{p_i^{gt}}{K(p_i)} \quad K(p_i)
\]

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

\[
L_{\text{reg}} \quad p_i \quad L_{\text{dir}} \quad \text{regression} \quad \text{classification} \quad \text{direction estimation}
\]

**Analysis-1:** Consequently, the Eq[10] outcomes illustrated in Fig[2b] with averaging sampling ten thousands data. The color intensity indicates the value of the gradient backpropagation for the classification result from corresponding \( p_i \), \( L_{\text{reg}}, L_{\text{dir}} \). Three axis represent the value of \( L_{\text{reg}}, L_{\text{dir}} \) respectively. Its counterpart \( \frac{\partial H_{3D-Har}}{\partial p_i} \) is shown in Fig[2a] with same axis representation.

Note: the gradient backpropagation from the classification part has no relationship with the regression part and direction estimation part which can be observed in Fig[2a] (better view in its vertical view Fig[2e]). In the case of 3D harmonic loss (better view in Fig[2f]), the gradient from classification loss is suppressed by high regression loss (bad localization), leading to relatively low confidence, which brings mutual consistency between classification and localization. Additionally, the \( L_{\text{dir}} \)
Based on the direction part is derived as follows.

\[
\frac{\partial L_{3D-Har}}{\partial \Delta d_i} = -e^{-L_{reg} (\Delta d_i)} L_{cls} \left( \frac{\partial L_{reg} (\Delta d_i)}{\Delta d_i} \right) \\
+ \left( 1 + e^{-L_{cls}} \right) \left( \frac{\partial L_{reg} (\Delta d_i)}{\partial \Delta d_i} \right) \\
+ e^{-L_{reg} (\Delta d_i)} \left( \frac{\partial L_{reg} (\Delta d_i)}{\partial \Delta d_i} \right) \cdot L_{dir}
\]  

(11)

The gradient backpropagation from regression result is highly associated to

\[
\Delta d_i, L_{cls}, L_{dir}, \text{regression, classification, direction estimation}
\]

**Analysis-2:** The Eq (11) outcomes are illustrated in Fig [2] using ten thousands data samples. The color intensity indicates the value of the \(\partial H_{3D-Har} / \partial \Delta d_i\) result from corresponding \(\Delta d_i, L_{cls}, L_{dir}\). Three-axis represent the value of \(\Delta d_i\), \(L_{reg}\), and \(L_{dir}\) respectively. Its counterpart \(\partial H_{3D} / \partial \Delta d_i\) is shown in Fig [2c] with same axis representation. In previous 3D detection learning, the gradient backpropagation from the regression part is independent of the classification and direction estimation. Although our proposed method achieved the same regression result (same \(\Delta d_i\)), increasing classification loss will constantly restrain the gradient from maintaining the synchronous learning of classification and regression (better view in Fig [2h]). The global unique optimization of \(\partial H_{3D} / \partial \Delta d_i = 0\) 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.

\[
L_{dir} (p_i) = \left( 1 - p_i \right) \log \left( 1 - p_i \right) - p_i \log (p_i)
\]  

(12)

Such that

\[
\frac{\partial L_{dir}}{\partial p_i} \left( p_i \right) = - \frac{p_i - p_i \log (p_i)}{1 - p_i} = \mu (i)
\]

Based on the \(p_i \log (p_i)\) status, the binary cross entropy loss is updated as follows

\[
\mu (i) = \begin{cases} 
-\frac{1}{p_i}, & \text{if } p_i = 1 \\
1 - \frac{1}{1 - p_i}, & \text{if } p_i = 0
\end{cases}
\]  

(13)

The gradient from the updated direction loss is as follows.

\[
\frac{\partial L_{3D-Har}}{\partial p_i} = \left( 1 - \beta_i + \beta_i \right) \cdot \frac{\partial L_{3D-Har}}{\partial p_i} + \left( 1 - \beta_i + \beta_i \right) \cdot \left[ \frac{-p_i \log (p_i)}{1 - p_i} \right]
\]  

(14)

The type of direction loss estimation is dependent on the \(p_i \log (p_i)\) status, and it is derived as follows.

\[
\frac{\partial L_{3D-Har}}{\partial p_i} = \begin{cases} 
\left( 1 - e^{-L_{cls} + L_{reg}} \right) \left( \frac{-1}{p_i} \right), & \text{if } p_i = 1 \\
\left( 1 - e^{-L_{cls} + L_{reg}} \right) \left( 1 - \frac{1}{1 - p_i} \right), & \text{if } p_i = 0
\end{cases}
\]  

(15)

The gradient backpropagation from the direction result is highly associated to

\[
L_{reg}, L_{cls}, \text{regression, classification, direction estimation}
\]

**Analysis-3:** The Eq [15] outcomes are illustrated in Fig [3] using ten thousands data samples. The color intensity indicates the value of the \(\partial H_{3D-Har} / \partial p_i\) result from corresponding \([p_i, L_{reg}], L_{cls}\). Three-axis represent the value of \(p_i\), \(L_{reg}\), and \(L_{cls}\), respectively. For instance, in Fig [3a], given \(p_i = 0\), the global unique optimization is \(\partial H_{3D} / \partial p_i = 0\) because when \(p_i = 0\), \(L_{cls} = 0\) and \(L_{reg} = 0\). Fig [3b] shows the same paradigm of such intrinsic-consistency when \(p_i = 1\).

Fig. 3. Visualization of the gradients from direction estimation part in our proposed 3D harmonic loss. (a) when \(p_i = 0\). (b) when \(p_i = 1\).

IV. EXPERIMENTS AND ANALYSIS

A. Dataset and evaluation metrics

The proposed 3D harmonic loss method performance is evaluated based on KITTI [26] dataset and DAIR-V2X-I dataset [27]. Both datasets offer LiDAR pointcloud data and 3D object annotations.

KITTI dataset has 7481 training frames and 7518 test frames and following previous work [6], [8], [12], [20] separated the 7481 training frames into the training dataset (3712 frames) and validation dataset (3769 frames). Accordingly, mean average precision (mAP) with 40 recall positions and Average Orientation Similarity (AOS) are adopted as evaluation metrics for detection accuracy analysis.

DAIR-V2X [27] dataset enables DAIR-V2X-I sub dataset for infrastructure-based 3D object detection experiments. It contains 10k lidar pointcloud frames scanned from LiDAR installed on the infrastructure side. About 493k 3D objects are annotated in these frames. Like the KITTI dataset, DAIR-V2X-I has three classes such as car, pedestrian and cyclist. Following the DAIR official toolkit and their experiments [27], we transfer the DAIR-V2X-I dataset into the KITTI data format and adopt the same evaluation metrics as the KITTI dataset.

B. Implementation

In this work, we conducted experiments on a server with a single NVIDIA GeForce RTX 2080Ti GPU. On KITTI dataset, four widely used models (one-stage detectors: PointPillar [20] and SECOND [6], two-stage detectors: PointRCNN [8] and SECOND [6], two-stage detectors: PointRCNN [8].
### TABLE I

| Method | Type / Modality | IoU threshold: 0.7 | IoU threshold: 0.5 |
|--------|-----------------|--------------------|--------------------|
|        |                 | Easy | Mod. | Hard | Easy | Mod. | Hard |
| PointPillar [20] | one-stage / LiDAR | N/A | 87.70 | N/A | N/A | N/A | N/A |
| PointPillar [20] | one-stage / LiDAR | 92.09 | 87.85 | 83.35 | 95.72 | 94.72 | 90.08 |
| Harmonic PointPillar (Ours) | one-stage / LiDAR | 94.07 | 88.41 | 85.42 | 95.98 | 94.87 | 92.09 |
|                  | N/A              | +1.98 | +0.56 | +2.07 | +0.26 | +0.15 | +2.01 |
| SECOND [6] | one-stage / LiDAR | 89.96 | 87.07 | 79.66 | N/A | N/A | N/A |
| SECOND [6] | one-stage / LiDAR | 93.47 | 88.96 | 86.23 | 96.60 | 95.27 | 92.60 |
| Harmonic Second (Ours) | one-stage / LiDAR | 95.41 | 89.23 | 86.25 | 98.96 | 95.63 | 94.63 |
|                  | N/A              | +1.94 | +0.27 | +0.02 | +2.36 | +0.36 | +2.03 |
| Point RCNN [8] | two-stage / LiDAR | N/A | N/A | N/A | N/A | N/A | N/A |
| Point RCNN [8] | two-stage / LiDAR | 94.97 | 88.27 | 88.02 | 98.45 | 94.30 | 94.01 |
| Harmonic Point RCNN (Ours) | two-stage / LiDAR | N/A | N/A | N/A | N/A | N/A | N/A |
|                  | N/A              | +2.22 | +0.67 | +0.04 | +0.98 | +1.24 | -0.12 |

\( \Delta \): reported results in paper, \( \dagger \): our implementation on mmdetection3D [39]. N/A: not available or not applicable. \( \Delta_{\text{average}} = +0.81 \). Emphases are highlighted in bold.

### TABLE II

| Method | Type / Modality | IoU threshold: 0.7 | IoU threshold: 0.5 |
|--------|-----------------|--------------------|--------------------|
|        |                 | Easy | Mod. | Hard | Easy | Mod. | Hard |
| PointPillar [20] | one-stage / LiDAR | N/A | 77.40 | N/A | N/A | N/A | N/A |
| PointPillar [20] | one-stage / LiDAR | 88.67 | 76.44 | 73.27 | 95.67 | 94.41 | 89.88 |
| Harmonic PointPillar (Ours) | one-stage / LiDAR | 87.66 | 77.76 | 73.44 | 95.95 | 94.72 | 90.12 |
|                  | N/A              | -0.01 | +1.32 | +0.17 | +0.28 | +0.31 | +0.24 |
| SECOND [6] | one-stage / LiDAR | 87.43 | 76.48 | 69.10 | N/A | N/A | N/A |
| SECOND [6] | one-stage / LiDAR | 88.58 | 79.78 | 76.49 | 96.06 | 95.01 | 92.45 |
| Harmonic Second (Ours) | one-stage / LiDAR | 91.16 | 79.68 | 76.06 | 98.88 | 95.36 | 92.56 |
|                  | N/A              | +1.58 | -0.10 | -0.43 | +2.32 | +0.35 | +0.11 |
| Point RCNN [8] | two-stage / LiDAR | 88.88 | 78.63 | 77.38 | N/A | N/A | N/A |
| Point RCNN [8] | two-stage / LiDAR | 90.99 | 80.20 | 77.93 | 98.11 | 94.20 | 93.83 |
| Harmonic Point RCNN (Ours) | two-stage / LiDAR | 91.77 | 80.07 | 77.26 | 98.41 | 94.17 | 93.88 |
|                  | N/A              | +0.78 | -0.13 | -0.67 | +0.60 | -0.03 | +0.05 |
| Part-A\(^2\) [12] | two-stage / LiDAR | 89.47 | 79.47 | 78.54 | 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 |
| Harmonic Part-A\(^2\) (Ours) | two-stage / LiDAR | 91.92 | 82.43 | 80.07 | 97.92 | 94.01 | 93.85 |
|                  | N/A              | +0.11 | +0.08 | -0.09 | +1.01 | -0.08 | -0.10 |

\( \Delta \): reported results in paper, \( \dagger \): our implementation on mmdetection3D [39]. N/A: not available or not applicable. \( \Delta_{\text{average}} = +0.32 \). Emphases are highlighted in bold.

and Part-A\(^2\) [12] are adopted as baselines for estimating proposed model effectiveness. We re-implemented and trained these models using mmdetection3D [39] platform. Besides applying our proposed 3D harmonic loss, these models were trained with their original training settings and parameters. Our models are named Harmonic PointPillar, Harmonic Second, Harmonic PointRCNN, and Harmonic Part-A\(^2\). The post-processing is kept the same as the baselines in the evaluation stage. We also submitted the results of Harmonic PointPillar on the KITTI test dataset to the KITTI official benchmark. We compared the performance of our model with PointPillar (baseline), and other models [7], [9], [11], [14], [31], [33]–[35], [40]–[42], including fusion-based, two-stage lidar-based and one-stage lidar-based methods. PointPillar [20] and SECOND [6], two widely used one-stage detectors, are taken as baselines to assess the proposed model performance using the DAIR-V2X-I dataset. We implemented them on mmdetection3D [39] following the DAIR-V2X toolkit with original training parameters and evaluation settings. For a fair comparison, the only difference between our models and baselines is that our models adopt the proposed 3D harmonic loss.

### C. Quantitative analysis

Experimental results with thorough quantitative analysis are reported below.
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]     | AAI 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]  | AAII 2021    | two-stage / LiDAR     | 94.85       | 88.83      | 86.13      |
| EQ-PVRCCN [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]       | AAII 2020    | one-stage / LiDAR     | 91.58       | 86.54      | 81.19      |
| VoxSet [14]      | CVPR 2022    | one-stage / LiDAR     | 92.70       | 89.07      | 86.29      |
| PointPillar [20] | CVPR '19     | one-stage / LiDAR     | 90.07       | 86.36      | 82.81      |
| Harmonic PointPillar | Ours   | one-stage / LiDAR     | 90.89       | 87.28      | 82.54      |
| ∆                | N/A          | N/A                   | +0.82       | +0.72      | -0.27      |

Results of listed works were extracted from KITTI BEV test benchmark [26] (Date: 14 August 2022). N/A: not applicable. ∆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=cf021462bb1955480c0c5ebe6c175654b08566.

Car detection is an essential element for ITS applications (e.g. V2V and V2X). We test our method’s performance from on-vehicle settings and roadside deployment. Tab. I and Tab. II show the mAP of car detection with IoU=0.7 and 0.5, and our proposed method achieves better average mAP values than baselines. Especially on BEV detection, our models achieved notable mAP rate (at least 0.02% and at most 2.36% beyond SECOND, at least 0.15% and at most 2.07% beyond PointPillar). We also submitted our model Harmonic PointPillar to the KITTI official test benchmark (see Tab. II) and our method’s time efficiency (ref to Tab. VII) makes it popular for industrial applications. Such as, our method optimizes the baseline PointPillar by 0.82% improvement on Easy samples and 0.72% improvement on Moderate samples, with only a 0.27% decrease on Hard samples. Since our method focuses on balancing and harmonizing the gradient from different parts, for hard samples, usually with very sparse points scanned from objects, the classification confidence is low, suppressing the regression part and causing the mAP drop. On the other side, extremely hard samples (outliers) may affect the stability of
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) [20] suffers from inconsistency problem in 3D detection, while Harmonic PointPillar (Ours) shows a great robustness on keeping predictions consistent.

### TABLE IV

mAP evaluation of 3D/BEV object detection on car class of DAIR-V2X-I dataset

| Method                  | BEV Car | 3D Car |
|-------------------------|---------|--------|
|                         | Easy    | Mod.   | Hard   | Easy    | Mod.   | Hard   |
| PointPillar [20]        | 64.77   | 54.76  | 54.77  | 64.40   | 54.39  | 54.43  |
| Harmonic PointPillar (Ours) | +2.12   | -0.30  | -0.29  | +0.30   | -0.29  | +0.30  |
| SECOND [6]              | 67.10   | 54.85  | 54.74  | 66.36   | 54.19  | 54.13  |
| Harmonic SECOND (Ours)  | +0.02   | N/A    | +0.07  | N/A     | +0.08  |        |

Results are from our implementation on mmdetection3D [39] following DAIR-V2X toolkit [27]. N/A: not available or not applicable. IoU threshold: 0.7. \( \Delta \)average = +0.07. Emphases are highlighted in bold.

the model because they always bring large gradients variant. Tab. [V] shows evaluation results on the DAIR-V2X-I dataset. Our model achieves a minor average improvement (0.07%). Since most current LiDAR-based 3D car detection models were designed and experimented with from an on-vehicle LiDAR view. Our future work will be to design better 3D detection methods tailored for on-infrastructure LiDAR.

### Direction estimation

We use the average orientation similarity (AOS) index to evaluate the performance of 3D direction. AOS evaluation under different IoU threshold (0.7 and 0.5) are shown in Tab. [V]. A higher AOS value represents better direction estimation for 3D objects. Our proposed model achieved an average improvement of 0.26% from PointPillar to Harmonic PointPillar and 0.76% from SECOND to Harmonic SECOND. Specifically, the proposed strategy helps vanilla models to gain a large improvement on easy-level objects (at least 0.35% and at most 1.99% improvement under 0.7 IoU threshold; at least 0.22% and at most 1.76% improvement under 0.5 IoU threshold) and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.

Table V shows evaluation results on the DAIR-V2X-I dataset. Our model achieves a minor average improvement (0.07%). Since most current LiDAR-based 3D car detection models were designed and experimented with from an on-vehicle LiDAR view. Our future work will be to design better 3D detection methods tailored for on-infrastructure LiDAR.

### Vulnerable road users detection

Besides car detection, detecting vulnerable road users (pedestrians and cyclists) provides additional security monitoring value to the V2X applications. From our observation and analysis, unlike vehicles, the scanned pointcloud shapes of pedestrians and cyclists are more irregular with different postures, which creates severe difficulty in model optimization. Tab. [VI] lists the mAP comparison of detecting pedestrians and cyclists under the official 0.5 IoU threshold. The proposed method achieved a better synchronous learning rate for classification, localization and direction estimation (related to the object shape). Compared to the base models, our model markedly boosts the accuracy under 0.5 IoU threshold and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.

| Method                  | IoU threshold: 0.5 | IoU threshold: 0.7 |
|-------------------------|-------------------|-------------------|
|                         | Easy   | Mod. | Hard  | Easy   | Mod. | Hard  |
| SECOND [6]              | 96.27  | 92.15 | 88.94 | 96.50  | 94.95 | 91.84 |
| Harmonic SECOND (Ours)  | +1.99  | 0.13  | -0.30 | +1.76  | 0.18  | +1.52 |
| PointPillar [20]        | 95.31  | 91.33 | 87.42 | N/A    | N/A   | N/A   |
| Harmonic PointPillar (Ours) | +0.35  | 0.09  | -0.30 | +0.22  | -0.10 | +1.50 |

\( \diamond \): results on KITTI validation dataset. \( \ast \): results on KITTI test benchmark. N/A: not applicable. Emphases are highlighted in bold.

Table VI shows evaluation results on the DAIR-V2X-I dataset. Our model achieves a minor average improvement (0.07%). Since most current LiDAR-based 3D car detection models were designed and experimented with from an on-vehicle LiDAR view. Our future work will be to design better 3D detection methods tailored for on-infrastructure LiDAR.

### Direction estimation

We use the average orientation similarity (AOS) index to evaluate the performance of 3D direction. AOS evaluation under different IoU threshold (0.7 and 0.5) are shown in Tab. [V]. A higher AOS value represents better direction estimation for 3D objects. Our proposed model achieved an average improvement of 0.26% from PointPillar to Harmonic PointPillar and 0.76% from SECOND to Harmonic SECOND. Specifically, the proposed strategy helps vanilla models to gain a large improvement on easy-level objects (at least 0.35% and at most 1.99% improvement under 0.7 IoU threshold; at least 0.22% and at most 1.76% improvement under 0.5 IoU threshold) and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.

### Vulnerable road users detection

Besides car detection, detecting vulnerable road users (pedestrians and cyclists) provides additional security monitoring value to the V2X applications. From our observation and analysis, unlike vehicles, the scanned pointcloud shapes of pedestrians and cyclists are more irregular with different postures, which creates severe difficulty in model optimization. Tab. [VI] lists the mAP comparison of detecting pedestrians and cyclists under the official 0.5 IoU threshold. The proposed method achieved a better synchronous learning rate for classification, localization and direction estimation (related to the object shape). Compared to the base models, our model markedly boosts the accuracy under 0.5 IoU threshold and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.

Table V shows evaluation results on the DAIR-V2X-I dataset. Our model achieves a minor average improvement (0.07%). Since most current LiDAR-based 3D car detection models were designed and experimented with from an on-vehicle LiDAR view. Our future work will be to design better 3D detection methods tailored for on-infrastructure LiDAR.

### Direction estimation

We use the average orientation similarity (AOS) index to evaluate the performance of 3D direction. AOS evaluation under different IoU threshold (0.7 and 0.5) are shown in Tab. [V]. A higher AOS value represents better direction estimation for 3D objects. Our proposed model achieved an average improvement of 0.26% from PointPillar to Harmonic PointPillar and 0.76% from SECOND to Harmonic SECOND. Specifically, the proposed strategy helps vanilla models to gain a large improvement on easy-level objects (at least 0.35% and at most 1.99% improvement under 0.7 IoU threshold; at least 0.22% and at most 1.76% improvement under 0.5 IoU threshold) and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.

### Vulnerable road users detection

Besides car detection, detecting vulnerable road users (pedestrians and cyclists) provides additional security monitoring value to the V2X applications. From our observation and analysis, unlike vehicles, the scanned pointcloud shapes of pedestrians and cyclists are more irregular with different postures, which creates severe difficulty in model optimization. Tab. [VI] lists the mAP comparison of detecting pedestrians and cyclists under the official 0.5 IoU threshold. The proposed method achieved a better synchronous learning rate for classification, localization and direction estimation (related to the object shape). Compared to the base models, our model markedly boosts the accuracy under 0.5 IoU threshold and hard-level objects (at least 1.50% and at most 1.52% improvement under 0.5 IoU threshold). The results prove that our proposed method can better estimate object direction.
of pedestrian and cyclist detection at most mAP improvement of 1.39%, and 0.49%, respectively. It shows that our proposed method is highly reliable in promoting the 3D detection of vulnerable traffic objects.

**TABLE VI**

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) | +0.82 | +1.19 | +0.90 | +0.26 | +0.14 | +0.09 |
| PointPillar [20]     | 52.31 | 46.31 | 41.33 | 77.33 | 60.49 | 56.48 |
| Harmonic PointPillar (Ours) | -0.18 | -0.16 | -0.21 | +1.39 | +1.30 | +0.59 |

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

**Time efficiency** of 3D detection matters in V2X applications. Roadside units (RSU) enable fast 3D detection models to catch vehicle signals in real-time. Tab. VII shows the runtime comparison of different state-of-the-art 3D detectors. On average, our method, same as PointPillar [20], is at least 1.5X times fast in speed than other methods since it works as a training optimizer, and does not cause a delay in detection inference. This measurement certifies that our method is time-friendly.

**TABLE VII**

Average runtime comparison of 3D/BEV object detection

| Method        | Source       | Type / Modality | Speed (Hz) |
|---------------|--------------|-----------------|------------|
| Point-RCNN [8] | CVPR 2019    | one-stage / LiDAR | 2.7        |
| Part-A$^2$ [12] | TPAMI 2021   | one-stage / LiDAR | 9.5        |
| CenterPoint [17]| CVPR 2021    | one-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 2021    | one-stage / LiDAR | 29.4       |
| VoxSet [14]   | CVPR 2022    | one-stage / LiDAR | 24.2       |
| PointPillar [20]| 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 with detailed qualitative analysis are reported below.

**Overall performance**: Fig.4 shows comparison of overall performance of 3D detection. In the examples of Fig.4(a) and Fig.4(d), our model (Harmonic PointPillar) achieves better localization accuracy than baseline model (PointPillar). In Fig.4(b) and Fig.4(c), our model detected more valid objects which are missed by the baseline models. Furthermore, our model’s false positive (FP) ratio is less than the baseline model in all example frames.

**Dealing with inconsistency problem**: In order to further verify the effectiveness of the proposed method in solving inconsistency problems in 3D detection, we show a more detailed qualitative visualization in Fig.5 (zoom in for detail check). The baseline model (PointPillar) has not succeeded in predicting the targets in all example frames because of the inconsistency between classification and localization. Consequently, our model achieved outstanding robustness in preserving consistent 3D detection, resulting in the best predictions in all example frames. It confirms that our method can be used to build a task-consistent 3D detector.

**E. Simulations on Realistic Deployment**

Based on the experience from previous work [25], we converted PyTorch-style Harmonic PointPillar into TensorRT-format. The tensorRT-format model is deployed on Jetson Xavier TX with float16 quantization techniques, and the same experiments as on PC were repeated on Jetson Xavier TX. A notable 2x-speed (75.4Hz on Jetson Xavier TX VS 43.1Hz on PC Single 2080Ti) is measured along with at most 1% mAP drop. Moreover, the jetson results are evidence that our proposed method is feasible for edge orchestration due to the consistent, continuous trade-off between time efficiency and model accuracy with low energy consumption. Fig.6 shows the qualitative example of on-infrastructure detection from TensorRT-format Harmonic PointPillar on Jetson Xavier TX.

**V. Conclusion**

In this paper, an inconsistency problem is formulated consistently to meet the comprehensive results compared to the state-of-the-art methods. Simulation outcomes show that the proposed method achieved better results on-edge 3D object detection for strengthening the V2X frameworks. First, the foremost cause of the inconsistency among classification, localization and direction estimation is analyzed and derived theoretically and mathematically. Second, the proposed 3D harmonic loss function effectively resolved the inconsistency problem in the pointcloud domain and achieved higher mAP with 75.4 Hz deployment speed than the baseline models. Mathematical derivatives are amended to claim the effectiveness of our proposed loss mechanism. Comprehensive experiment results demonstrate that our proposed method resolved the inconsistency problem to a large extent and achieved detection accuracy on average with no extra inference time cost. Our future work will 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. Roccaletti, 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. 21002–21012, 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, “Voxelnets: 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-r-cnn: 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] 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. 7516–7524.
[19] C. Lin and D. Tian, “C3lid: Camera-lidar 3d object detection with point feature enhancement and point-guided fusion,” IEEE Transactions on Intelligent Transportation Systems, 2022.