

Perspective-Invariant 3D Object Detection

– Supplementary Material –

Ao Liang^{*,1,2,3,4} Lingdong Kong^{*,1,5} Dongyue Lu^{*,1} Youquan Liu⁶
Jian Fang⁴ Huaici Zhao^{4,✉} Wei Tsang Ooi^{1,✉}

¹National University of Singapore

²University of Chinese Academy of Sciences

³Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences

⁴Shenyang Institute of Automation, Chinese Academy of Sciences

⁵CNRS@CREATE

⁶Fudan University

🌐 Project Page: [Link](#)

🐱 GitHub: [Link](#)

📊 Dataset: [Link](#)

Table of Contents

A Pi3DET: Construction & Statistics	1
A.1 Overview	1
A.2 Dataset Statistics	1
A.3 Dataset Examples	2
A.4 Cross-Platform Discrepancies	3
A.5 Comparisons with Other Datasets	3
A.6 Cross-Platform Annotation Toolkit	4
A.7 License	4
B Additional Implementation Details	4
B.1 Benchmark Construction	7
B.2 Summary of Notations	7
B.3 Training Configurations	7
B.4 Evaluation Protocols	7
B.5 Summary of Detection Baselines	8
B.6 Summary of Adaptation Baselines	8
C Additional Experimental Analyses	9
C.1 Additional Quantitative Results	9
C.2 Additional Qualitative Results	12
C.3 Failure Cases	12
D Broader Impact	13
D.1 Potential Societal Impact	13
D.2 Potential Limitations	13
D.3 Future Directions	13

E Public Resources Used	13
E.1. Public Codebase Used	13
E.2. Public Datasets Used	13
E.3. Public Implementations Used	13

A. Pi3DET: Construction & Statistics

In this section, we briefly outline the overview of the proposed **Pi3DET** dataset, present detailed statistics, showcase representative examples, analyze cross-platform discrepancies, compare with existing 3D detection datasets, and describe the annotation toolkit used for precise 3D labeling.

A.1. Overview

Pi3DET is the first benchmark designed for 3D object detection across multiple robot platforms. Built upon M3ED [3], our dataset consists of 25 sequences collected from three distinct platforms: 🚗 **Vehicle**, 🚁 **Drone**, and 🦘 **Quadruped**.

In each sequence, detailed 10 Hz annotations are performed for vehicle and pedestrian targets, resulting in a total of 51,545 annotated frames. The dataset spans a wide range of environmental conditions – including both daytime and nighttime scenes – and encompasses urban, suburban, and rural settings. This extensive and diverse benchmark offers a valuable resource for advancing cross-platform 3D object detection research.

A.2. Dataset Statistics

Tab. A summarizes the detailed statistics of the **Pi3DET** dataset. In total, our dataset comprises 25 sequences collected from three robot platforms: Vehicle, Drone, and Quadruped.

- The 🚗 **Vehicle** subset (eight sequences in total) contains 32,193 frames with approximately 346.95 million LiDAR

(*) Ao, Lingdong, and Dongyue contributed equally to this work.

Table A. Summary of the platform-level and sequence-level statistics of the proposed **Pi3DET** dataset.

Platform	Condition	Sequence	# of Frames	# of Points (M)	# of Vehicles	# of Pedestrians
Vehicle (8)	Daytime (4)	city_hall	2,982	26.61	19,489	12,199
		penno_big_loop	3,151	33.29	17,240	1,886
		rittenhouse	3,899	49.36	11,056	12,003
		ucity_small_loop	6,746	67.49	34,049	34,346
	Nighttime (4)	city_hall	2,856	26.16	12,655	5,492
		penno_big_loop	3,291	38.04	8,068	106
		rittenhouse	4,135	52.68	11,103	14,315
		ucity_small_loop	5,133	53.32	18,251	8,639
Summary (Vehicle)		32,193	346.95	131,911	88,986	
Drone (7)	Daytime (4)	penno_parking_1	1,125	8.69	6,075	115
		penno_parking_2	1,086	8.55	5,896	340
		penno_plaza	678	5.60	721	65
		penno_trees	1,319	11.58	657	160
	Nighttime (3)	high_beams	674	5.51	578	211
		penno_parking_1	1,030	9.42	524	151
		penno_parking_2	1,140	10.12	83	230
		Summary (Drone)		7,052	59.47	14,534
Quadruped (10)	Daytime (8)	art_plaza_loop	1,446	14.90	0	3,579
		penno_short_loop	1,176	14.68	3,532	89
		rocky_steps	1,535	14.42	0	5,739
		skatepark_1	661	12.21	0	893
		skatepark_2	921	8.47	0	916
		srt_green_loop	639	9.23	1,349	285
		srt_under_bridge_1	2,033	28.95	0	1,432
		srt_under_bridge_2	1,813	25.85	0	1,463
	Nighttime (2)	penno_plaza_lights	755	11.25	197	52
		penno_short_loop	1,321	16.79	904	103
Summary (Quadruped)		12,300	156.75	5,982	14,551	
All Three Platforms (25)	Summary (All)		51,545	563.17	152,427	104,809

points, along with 131,911 vehicle and 88,986 pedestrian annotations.

- The 🚁 **Drone** subset (seven sequences in total) contains 7,052 frames, 59.47 million points, 14,534 vehicle annotations, and 1,272 pedestrian annotations.
- The 🦘 **Quadruped** subset (ten sequences in total) contains 12,300 frames with 156.75 million points, 5,982 vehicle annotations, and 14,551 pedestrian annotations.

Overall, **Pi3DET** consists of **51,545** frames and **563.17** million points, offering a diverse benchmark captured under varying conditions (daytime and nighttime) and across urban, suburban, and rural environments, thereby providing a comprehensive resource for real-world, cross-platform 3D object detection research.

For each platform in the **Pi3DET** dataset, we collect comprehensive statistics to characterize the data from multiple perspectives. Specifically, we compile point cloud distribution statistics including p^x , p^y , p^z coordinates and intensity

values to capture spatial density and spread. In addition, we gather 3D object statistics, such as the number of objects per frame and the average number of points per bounding box, to assess detection challenges across varying environments. Finally, we documented 3D bounding box statistics, detailing dimensions such as length (l), width (w), and height (h). Details are provided in the following sections.

A.3. Dataset Examples

In this section, we present some examples that demonstrate the rich diversity of the **Pi3DET** dataset. See Fig. F through Fig. I for details.

Pi3DET encompasses a wide range of scenes and temporal conditions. In particular, the quadruped platform is capable of operating in complex environments such as under bridges and on stairs, while the drone platform collects aerial views with significantly different imaging characteristics from the vehicle platform.

Overall, the vehicle platform generally provides a slightly downward-facing view; the quadruped platform offers an upward view, yet its motion is highly dynamic and terrain-dependent, leading to a broader distribution of view angles; and the drone platform, although it typically captures targets below, exhibits considerable jitter and a wider range of view distributions due to its increased degrees of freedom.

Specifically, for the quadruped platform, Fig. F displays several scenes captured in a skatepark, where the quadruped is positioned very close to people, and the individuals appear taller than the platform. Fig. H further shows the quadruped traversing stairs and operating under bridges, where the terrain induces significant tilting of the ego coordinate system. These examples clearly demonstrate that the quadruped’s viewpoint is markedly different from that of the vehicle, leading to distinctly varied imaging effects.

For the drone platform, Fig. F and Fig. H illustrate sample frames captured during flight, showing that targets are predominantly located below the drone. The drone’s inherent jitter further contributes to imaging effects that differ substantially from those observed on the vehicle platform.

In addition, Fig. G and Fig. I showcase data collected under nighttime conditions across all three platforms. Collectively, these examples underscore the rich diversity of the Pi3DET dataset and highlight the unique challenges associated with cross-platform 3D object detection.

A.4. Cross-Platform Discrepancies

Our statistical analyses and visualizations reveal that cross-platform discrepancies are primarily influenced by differences in the z -axis distribution, object geometry, and target bounding box characteristics.

Tab. H, Tab. I, and Tab. J show that while the distributions of x , y , and intensity are largely similar across platforms, significant differences emerge along the z -axis. This is likely attributable to variations in sensor mounting height and motion space: vehicles, with higher, fixed sensor mounts, tend to produce point clouds concentrated just below the sensor (with z values slightly below zero); quadruped platforms, operating at lower heights near the ground, generate point clouds with z values closer to zero; and drone platforms, which operate at even greater altitudes, yield broader Z -axis distributions that remain mostly below zero.

Furthermore, Tab. K, Tab. L, and Tab. M show that vehicle targets typically measure around 4–5 meters in length, 2 meters in width, and 1.6–1.7 meters in height (with pedestrians around 1.7–1.9 meters). And the Vehicle platform also exhibits a wider range of object sizes (including larger vehicles like buses or trams exceeding 10 meters in length).

Analysis of the number of foreground objects and points per bounding box (Tab. N, Tab. O, Tab. P) further indicates that the Vehicle platform generally contains more diverse and numerous targets, while some sequences from the Drone and

Quadruped platforms may include only pedestrian targets.

In summary, our analyses demonstrate that differences in ego height and motion space significantly affect the z -axis distribution of LiDAR point clouds, leading to inconsistent object representations and spatial misalignments across platforms. These discrepancies pose considerable challenges for developing robust cross-platform 3D detection methods.

A.5. Comparisons with Other Datasets

In our experiments, we leverage two widely recognized datasets: nuScenes [2] and KITTI [7] to evaluate cross-platform and cross-dataset 3D object detection. Both datasets have distinct characteristics that contribute to the domain gap. Below is a summary of their key attributes:

- **nuScenes** [2] is a large-scale autonomous driving dataset collected from urban environments in Boston and Singapore. It employs a 32-beam LiDAR (Velodyne HDL-32E) alongside high-resolution cameras and radar to provide a comprehensive, multimodal view of complex urban scenes. The dataset encompasses approximately 1,000 scenes, with each scene lasting around 20 seconds, and includes roughly 28,130 training frames, 6,019 validation frames, and 6,008 test frames. These frames capture a wide variety of weather conditions, traffic densities, and dynamic urban scenarios, making nuScenes a challenging benchmark for 3D object detection and tracking tasks.
- **KITTI** [7] is one of the pioneering datasets for autonomous driving research, widely recognized for its high-quality 3D annotations and real-world driving scenarios. Captured using a 64-beam LiDAR (Velodyne HDL-64E) mounted on a vehicle, KITTI provides precise 3D point clouds over suburban and urban landscapes under relatively consistent weather conditions. The dataset is divided into roughly 7,481 training frames and 7,518 test frames, with detailed labels for objects such as vehicles, pedestrians, and cyclists. The comprehensive sensor data and annotations have established it as a fundamental benchmark for evaluating 3D object detection algorithms, despite its smaller scale compared to more recent datasets.

Tab. B provides an overview of key discrepancies across datasets and platforms. The nuScenes dataset, collected using a 32-beam LiDAR, offers a balanced set of urban road scenes with both daytime and nighttime data. In contrast, KITTI, captured with a 64-beam sensor, presents higher point density per scene but lacks nighttime data.

Pi3DET spans three platforms, each utilizing a 64-beam LiDAR with a uniform angular range of $[-22.5^\circ, 22.5^\circ]$. The Vehicle subset focuses on road environments with abundant training and validation frames, while the Quadruped subset captures more diverse terrains, including roads, stairs, and under bridges. The Drone subset, acquired in aerial environments, offers a comparable point density to the Vehicle subset.

Table B. Summary of the **cross-platform** and **cross-dataset** discrepancies in existing 3D detection datasets (nuScenes, KITTI, and ours).

Dataset	Beam Ways	Beam Angles	Points per Scene	Training Frames	Validation Frames	Night	Condition
nuScenes [2]	32	[−30.0, 10.0]	~ 25K	28, 130	6, 019	Yes	Road
KITTI [7]	64	[−23.6, 3.20]	~ 118K	3, 712	3, 769	No	Road
Pi3DET (Ours)	Vehicle	64	~ 110K	16, 888	15, 305	Yes	Road
	Quadruped		~ 87K	7, 204	5, 096	Yes	Road, Stair, Under Bridge
	Drone		~ 110K	3, 584	3, 468	Yes	Air

These differences highlight the diverse sensor configurations and environmental conditions, underscoring the challenges inherent in cross-dataset and cross-platform 3D detection.

A.6. Cross-Platform Annotation Toolkit

Our annotation process for Pi3DET is executed through a streamlined three-stage pipeline, which is described below.

A.6.1. Pseudo-Label Generation

We pre-trained a diverse set of state-of-the-art 3D object detectors (PV-RCNN [14], PV-RCNN++ [17], Voxel-RCNN [5], IA-SSD [28], CenterPoint [26], and SECOND [21]) on external datasets such as Waymo [18], nuScenes [2], and Lyft [8], and then used these models to infer initial pseudo-labels on the **Pi3DET** data.

A.6.2. Pseudo-Label Optimization and Filtering

We applied a kernel density estimation (KDE) algorithm to fuse predictions from multiple 3D object detectors and used the 3D multi-object tracking algorithm CTRL [6] to ensure temporal consistency and to interpolate missed detections.

In addition, we employed the vision foundation model Tokenize Anything (TA) [11] to project pseudo-labels onto corresponding RGB images and verify object categories within an open vocabulary. This step maps the TA outputs to the Pi3DET classes (Vehicle, Pedestrian), with mismatches flagged for manual review.

A.6.3. Manual Refinement

Using the open-source 3D annotation platform Xtreme1¹, three annotators manually refined each frame on a per-box basis. This process, which included cross-validation among multiple annotators, ensured that the final annotations are both precise and consistent.

This comprehensive annotation toolkit integrates modules for data visualization, model pre-training, multi-object tracking, 3D bounding box editing, and vision model inference. Although our automated framework greatly reduced the manual workload, the inherent sparsity and irregularity of point

cloud data required an average of over 30 seconds of manual intervention per frame, culminating in **more than 500 hours of annotation effort** for the entire **Pi3DET** dataset.

Our annotation pipeline is further illustrated by several figures. Fig. A depicts the pseudo-label generation process, where multiple pre-trained 3D detectors infer initial labels from the raw Pi3DET data.

Fig. B and Fig. C demonstrate the pseudo-label optimization and filtering stage, highlighting how kernel density estimation and the CTRL tracking algorithm fuse detector outputs and maintain temporal consistency, while the Tokenize Anything model [11] verifies the projected labels on RGB images.

Finally, Fig. D showcases the manual refinement interface provided by the Xtreme1 platform, where annotators conduct frame-by-frame corrections and cross-validation to ensure high annotation accuracy. These visualizations underscore the comprehensive and multi-faceted nature of our annotation toolkit, which has been instrumental in achieving a high-quality and consistent Pi3DET dataset.

A.7. License

The **Pi3DET** dataset and the associated benchmark are released under the Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)² license.

B. Additional Implementation Details

In this section, we provide additional implementation details to facilitate a thorough understanding and reproducibility of our work. We begin by describing the construction of our benchmark, which leverages data from three platforms in Pi3DET, as well as two widely used datasets (nuScenes [2] and KITTI [7]).

Based on these sources, we construct a total of **eight** cross-platform and cross-dataset adaptation tasks. The cross-platform adaptation tasks involve various combinations of Vehicle, Drone, and Quadruped subsets from **Pi3DET**, while the cross-dataset tasks evaluate the domain gap between nuScenes and other vehicle data (Pi3DET and KITTI [7]).

¹<https://github.com/xtreme1-io/xtreme1>.

²<https://creativecommons.org/licenses/by-sa/4.0/legalcode>.

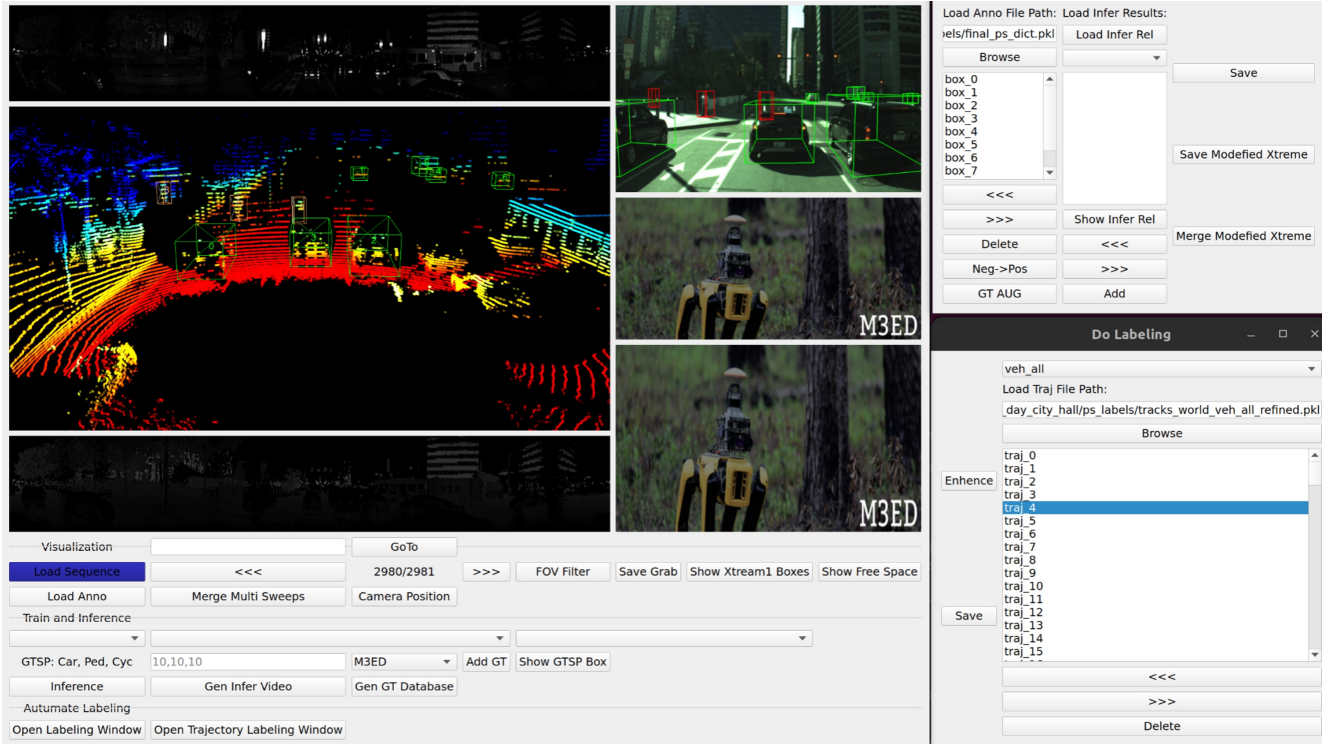


Figure A. **Model Pre-Training Interface:** This interface enables the pre-training of various 3D detection models to generate initial pseudo labels for subsequent processing.

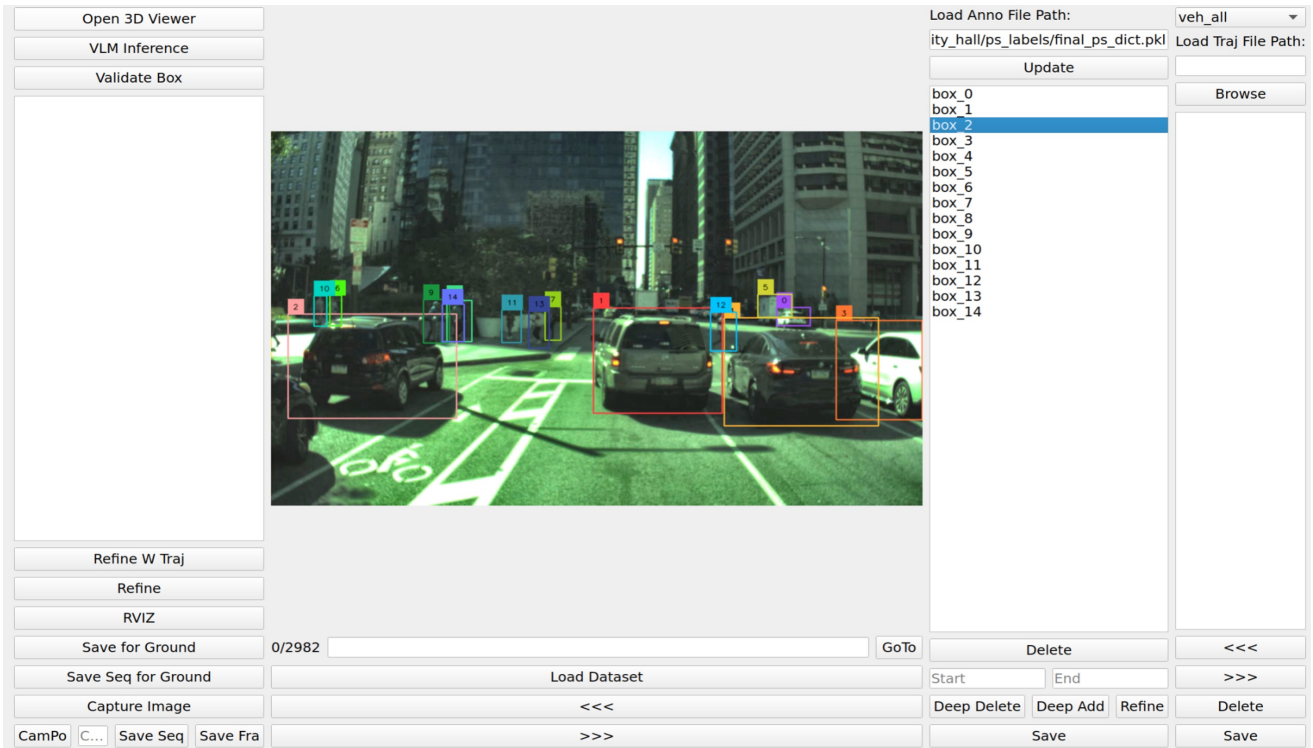


Figure B. **Pseudo-Label Filtering Interface:** In this view, 3D bounding boxes are projected onto corresponding RGB images, facilitating efficient and convenient filtering of pseudo labels.

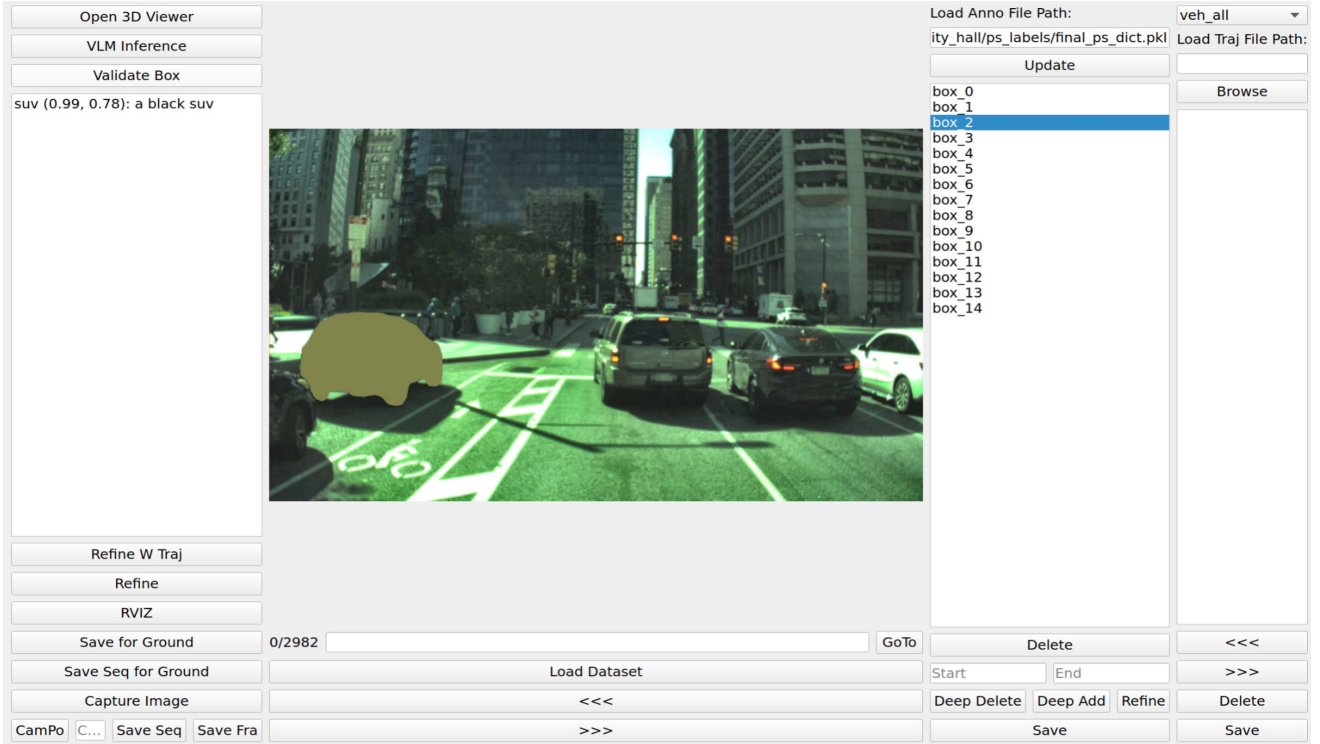


Figure C. **Automatic Pseudo-Label Screening** with TA [11]. This interface employs a vision foundation model (Tokenize Anything) to automatically filter pseudo labels by verifying alignment with image content, with mismatched frames flagged for manual review.

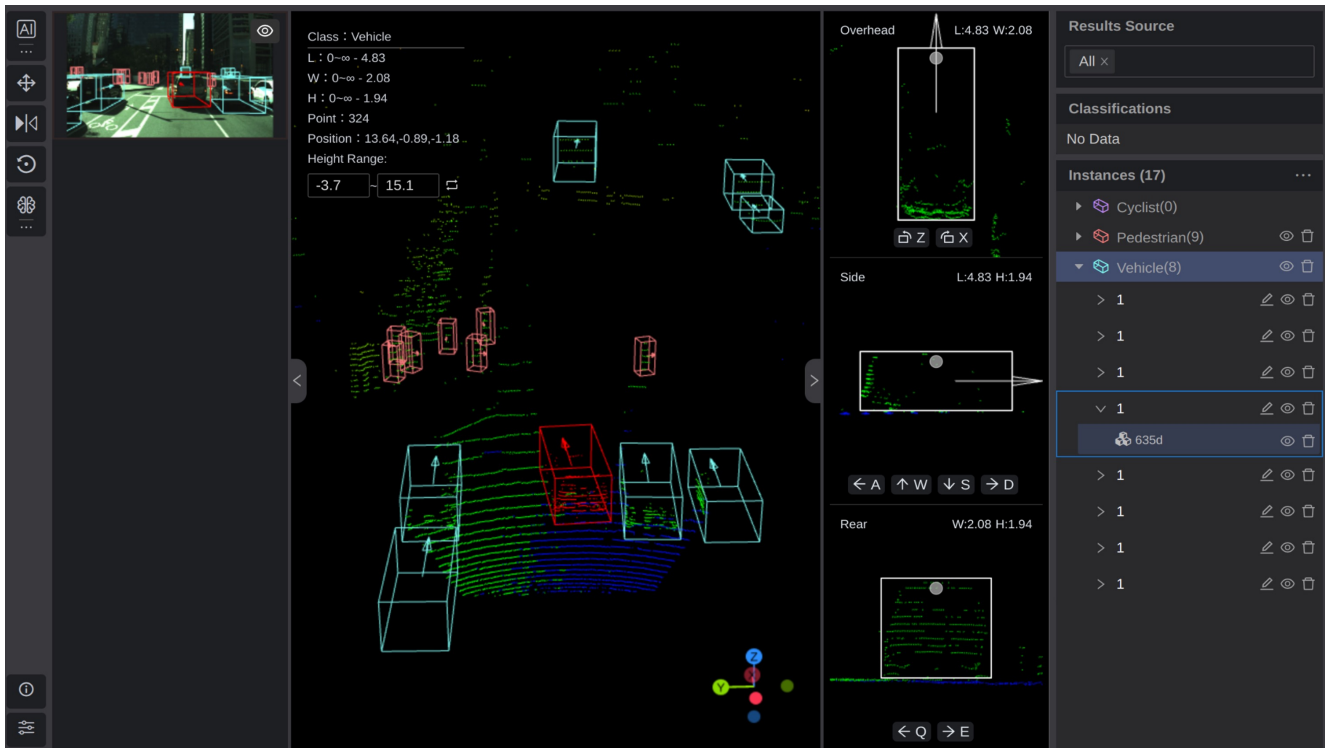


Figure D. **Manual Refinement Interface:** Utilizing the open-source Xtreme1 platform, this interface allows annotators to perform detailed frame-by-frame and box-by-box corrections, ensuring high-quality final annotations.

Following the benchmark construction, we summarize the notations used throughout our work in Tab. C for better clarity. We then detail our training configurations and evaluation protocols, which include specific settings used for both detection and adaptation baselines. Finally, we provide an overview of the detection baselines and adaptation baselines employed in our experiments.

The subsequent subsections elaborate on these aspects in detail, ensuring that all experimental and implementation choices are clearly documented.

B.1. Benchmark Construction

Building upon three platforms from **Pi3DET**, as well as the other two datasets (nuScenes [2] and KITTI [7]), we construct a total of **eight** cross-platform and cross-dataset adaptation tasks. These tasks are summarized as follows.

- **Cross-Platform Adaptation:**
 - Pi3DET (Vehicle) → Pi3DET (Drone)
 - nuScenes (Vehicle) → Pi3DET (Drone)
 - Pi3DET (Vehicle) → Pi3DET (Quadruped)
 - nuScenes (Vehicle) → Pi3DET (Quadruped)
 - Pi3DET (Quadruped) → Pi3DET (Drone)
 - Pi3DET (Drone) → Pi3DET (Quadruped)
- **Cross-Dataset Adaptation:**
 - nuScenes → Pi3DET (Vehicle)
 - nuScenes → KITTI

For the cross-platform adaptation tasks, we adopt PV-RCNN [14] and Voxel-RCNN [5] as the base 3D detectors. These state-of-the-art detectors utilize anchor-based and center-based detection heads, respectively, thereby covering the most popular 3D detection settings and demonstrating the generality of our approach.

For the cross-dataset adaptation tasks, we essentially employ the same configuration as in the cross-platform tasks; however, when KITTI [7] serves as the target dataset, we use the SECOND-IOU [21, 21] model, which is widely used in current cross-dataset methods to facilitate direct comparisons with reported results and highlight the effectiveness of our method. The data splits for each platform and dataset are summarized in Tab. B.

B.2. Summary of Notations

For better readability, the notations used in this work have been summarized in Tab. C.

B.3. Training Configurations

For all datasets, the detection range is fixed to $[-75.2 \text{ m}, 75.2 \text{ m}]$ along the X and Y axes and $[-2 \text{ m}, 4 \text{ m}]$ along the Z axis, with coordinate origins shifted to the ground plane. The voxel size is consistently set to $(0.1 \text{ m}, 0.1 \text{ m}, 0.15 \text{ m})$ across datasets. Data augmentation is widely adopted during both pre-training and self-training;

Table C. Summary of notations defined in this work.

Notation	Definition
β	Platform type
\mathcal{S}	Symbol denoting the source platform
\mathcal{T}	Symbol denoting the target platform
\mathcal{P}	LiDAR point cloud
\mathcal{B}	3D bounding box
N	Total number of LiDAR point clouds
M	Total number of 3D bounding boxes
(p^x, p^y, p^z)	Point coordinates in X, Y, Z directions
(c^x, c^y, c^z)	Center position of the 3D bounding box
l	Length of the 3D bounding box
w	Width of the 3D bounding box
h	Height of the 3D bounding box
φ	Heading angle of the 3D bounding box
ϕ	Roll angle of the ego platform
θ	pitch angle of the ego platform
ψ	Yaw angle of the ego platform
\mathbf{T}	Ego pose
\mathbf{R}	Ego rotation
$\Delta\phi$	Random jitter added to the roll angle
$\Delta\theta$	Random jitter added to the pitch angle
$\mathbf{F}^{\mathcal{S}}$	RoI feature from source platform
$\mathbf{F}^{\mathcal{T}}$	RoI feature from target platform

this includes random world flipping, scaling, and rotation, as well as random object rotation. In addition, Pi3DET-Net incorporates Random Platform Jitter, where rotations around the x and y axes ($\Delta\phi$) are uniformly sampled from $[-5^\circ, +5^\circ]$.

Our pre-training framework is built upon the open-source OpenPCDet project³ and is executed using two NVIDIA Titan RTX GPUs. For cross-platform tasks, 3D detectors are initially pre-trained on nuScenes with optimization settings that include a batch size of 4 per GPU for 20 epochs, use of the Adam optimizer with an initial learning rate of 0.01, weight decay of 0.001, and momentum of 0.9.

For cross-platform tasks on Pi3DET, we extend the pre-training to 40 epochs. In the cross-dataset adaptation tasks, we use the detector weights pre-trained on nuScenes from the ST3D++ framework⁴ to ensure fairness in comparisons.

B.4. Evaluation Protocols

We follow [22] and adopt the KITTI evaluation metric for the common category *Vehicle* (referred to as *Car* in the KITTI and nuScenes dataset). Our evaluation protocol uses the official KITTI criteria, reporting average precision (AP) in both bird’s-eye view (BEV) and 3D over 40 recall positions.

³<https://github.com/open-mmlab/OpenPCDet>.

⁴<https://github.com/CVMI-Lab/ST3D>.

Mean average precision is computed with an IoU threshold of 0.7 for cars and 0.5 for pedestrians and cyclists. For all tasks and datasets, the prediction confidence threshold for 3D detectors is set to 0.2.

For 3D IoU, given a predicted 3D box B_p and its corresponding ground truth B_{gt} , the IoU is calculated as:

$$\text{IoU} = \frac{\text{Vol}(B_p \cap B_{gt})}{\text{Vol}(B_p \cup B_{gt})}, \quad (1)$$

For BEV, the IoU is computed similarly using the 2D projections of the 3D boxes onto the ground plane.

The average precision (AP) is computed as follows:

$$\text{AP} = \frac{1}{40} \sum_{i=1}^{40} p_{\text{interp}}(r_i), \quad (2)$$

where r_i represents the i -th recall threshold (typically evenly spaced over the recall range), and $p_{\text{interp}}(r_i)$ is the interpolated precision defined as follows:

$$p_{\text{interp}}(r_i) = \max_{\tilde{r} \geq r_i} p(\tilde{r}), \quad (3)$$

with $p(\tilde{r})$ denoting the precision at recall \tilde{r} .

B.5. Summary of Detection Baselines

The following 3D object detection methods are used as baselines in our **Pi3DET** benchmark.

- **PV-RCNN** [14] is a two-stage 3D detection framework that effectively combines voxel-based and point-based representations. In the first stage, the model aggregates voxel features into keypoints via a voxel set abstraction module, which enables efficient proposal generation. In the second stage, PV-RCNN employs a RoI grid pooling module that leverages point-wise features to refine the candidate proposals, thereby achieving high localization accuracy and robust performance.
- **Voxel-RCNN** [5] is another two-stage detector that primarily relies on voxel representations. It integrates a voxel feature encoder for both proposal generation and refinement, enabling precise region proposal extraction from high-dimensional sparse data. The design emphasizes efficient voxel-based processing, reducing computational overhead while maintaining competitive accuracy in 3D object detection.
- **SECOND** [21], also termed as Sparsely Embedded Convolutional Detection, is a one-stage 3D detector that capitalizes on sparse convolutional networks to process voxelized point clouds. By converting irregular point cloud data into a structured voxel representation, SECOND applies sparse convolution operations to efficiently extract features and directly predict object classes and bounding boxes in a single forward pass. This design achieves a favorable

trade-off between detection speed and accuracy, making it a popular baseline in many 3D detection studies. Following the design proposed in ST3D++ [23], we improve the SECOND detector by incorporating an additional IoU head to estimate the IoU between object proposals and their corresponding ground truths, naming the modified detector **SECOND-IoU**.

In our experiments, PV-RCNN and Voxel-RCNN are two-stage detectors that respectively employ anchor-based and center-based detection heads, while SECOND is a one-stage detector. This comprehensive setting covers a broad range of popular 3D detection designs, thereby demonstrating the generality of our proposed approach.

B.6. Summary of Adaptation Baselines

The following cross-domain 3D object detection methods are used as baselines in our **Pi3DET** benchmark.

- **ST3D** [22] is a self-training pipeline designed for cross-dataset adaptation on 3D object detection from point clouds. ST3D consists of three key components: 1) Random Object Scaling (ROS), which mitigates source domain bias by randomly scaling 3D objects during pre-training; 2) Quality-Aware Triplet Memory Bank (QTMB), which generates high-quality pseudo labels by assessing localization quality and avoiding ambiguous examples; and 3) Curriculum Data Augmentation (CDA), which progressively increases the intensity of data augmentation to prevent overfitting to easy examples and improve the ability to handle hard cases. ST3D iteratively improves the detector on the target domain by alternating between pseudo label generation and model training, achieving state-of-the-art performance on multiple 3D object detection datasets, even surpassing fully supervised results in some cases.
- **ST3D++** [23] introduces a holistic pseudo-label denoising pipeline to reduce noise in pseudo-label generation and mitigate the negative impacts of noisy pseudo labels on model training. The pipeline consists of three key components: 1) Random Object Scaling (ROS), which reduces object scale bias during pre-training; 2) Hybrid Quality-Aware Triplet Memory (HQTMB), which improves the quality and stability of pseudo labels through a hybrid scoring criterion and memory ensemble; and 3) Source-Assisted Self-Denoised Training (SASD) and Curriculum Data Augmentation (CDA), which rectify noisy gradient directions and prevent overfitting to easy examples. ST3D++ achieves state-of-the-art performance on multiple 3D object detection datasets, even surpassing fully supervised results in some cases, and demonstrates robustness across various categories such as cars, pedestrians, and cyclists. The method is model-agnostic and can be integrated with different 3D detection architectures.
- **MS3D++** [19] is a multi-source self-training framework

Table D. Ablation study on the adverse effects of the random object scaling (ROS) operation on the pseudo label quality.

Method	ROS	AP@0.70	AP@0.50
PV-RCNN [14]	✗	37.84 / 30.20	39.83 / 39.28
	✓	17.96 / 12.02	29.26 / 24.58
SECOND-IOU [21]	✗	32.47 / 28.21	38.76 / 37.25
	✓	19.75 / 10.40	36.14 / 31.94

designed for cross-dataset 3D object detection. The method addresses the significant performance drop (70-90%) that occurs when 3D detectors are deployed in unfamiliar domains due to variations in lidar types, geography, or weather. MS3D++ generates high-quality pseudo-labels by leveraging an ensemble of pre-trained detectors from multiple source domains, which are then fused using Kernel-density estimation Box Fusion (KBF) to improve domain generalization. Temporal refinement is applied to ensure consistency in box localization and object classification. The framework also includes a multi-stage self-training process to iteratively improve pseudo-label quality, balancing precision and recall. Experimental results on datasets like Waymo [18], nuScenes [2], and Lyft [8] demonstrate that MS3D++ achieves state-of-the-art performance, comparable to training with human-annotated labels, particularly in Bird’s Eye View (BEV) evaluation for both low and high-density lidar. The approach is highly versatile, allowing easy integration with various 3D detector architectures and data augmentation techniques without modifying the inference runtime of the detector.

- **ReDB** [4] aims to generate reliable, diverse, and class-balanced pseudo labels to iteratively guide self-training on a target dataset with a different distribution. The framework includes a cross-domain examination (CDE) to assess pseudo label reliability, an overlapped boxes counting (OBC) metric to ensure geometric diversity, and a class-balanced self-training strategy to address inter-class imbalance.

C. Additional Experimental Analyses

In this section, we present additional results to complement the findings reported in the main paper. First, we provide further quantitative results that reinforce our evaluation of cross-platform and cross-dataset adaptation performance. Next, we offer qualitative results with visual examples that highlight both the strengths and potential weaknesses of our approach. Finally, we analyze failure cases to identify specific scenarios where our method struggles, thereby offering insights for future improvements.

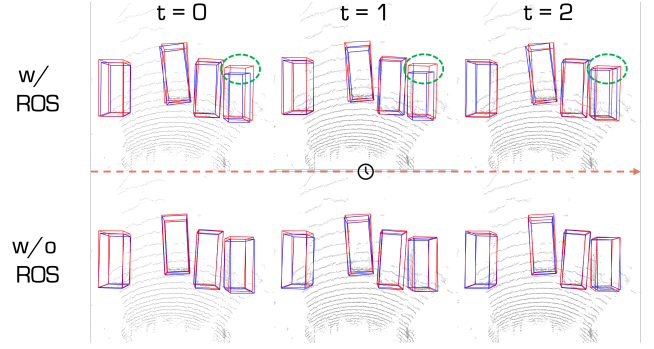


Figure E. Comparisons of inference results in a continuous static scene using PV-RCNN with and without ROS. The **red boxes** indicate ground truth, while the **blue boxes** denote predictions from the detector. Despite the ego vehicle and surrounding objects remaining static, the ROS-pretrained PV-RCNN yields variable predictions, whereas the model without ROS produces much more stable and consistent outputs.

C.1. Additional Quantitative Results

C.1.1. Adverse Effects of Random Object Scaling (ROS)

Random Object Scaling (ROS) is a data augmentation technique introduced in ST3D [22] for cross-dataset 3D object detection. The primary goal of ROS is to enhance the diversity of foreground objects in the source domain by randomly scaling the sizes of ground-truth bounding boxes. This augmentation strategy aims to mitigate the bias inherent in object size distributions, thereby improving the detector’s ability to extract robust foreground features.

In cross-dataset tasks such as nuScenes [2] → KITTI [7], Waymo [18] → KITTI [7], and Waymo [18] → nuScenes [2], ROS has demonstrated considerable benefits and has been adopted by subsequent methods, including ST3D++ [23] and ReDB [4].

However, our experiments on the Pi3DET dataset reveal that ROS has a deleterious effect on pseudo-label quality, particularly in high-frequency annotated data. Pi3DET is annotated at 10 Hz, meaning that in consecutive frames, although the LiDAR point clouds exhibit subtle variations due to sensor noise and slight motion, the positions and sizes of foreground objects remain essentially constant.

Under these conditions, ROS inadvertently exaggerates minor variations in object size, causing the detector to produce inconsistent predictions across similar frames. For instance, when evaluating a nuScenes → Pi3DET (Vehicle) cross-dataset task, we observed that PV-RCNN [14] and SECOND-IOU [21] models pre-trained with ROS experienced performance drops of approximately 60% and 63% respectively in AP_{3D} , as detailed in Tab. D.

Further analysis indicates that the adverse effects of ROS are mainly due to its sensitivity to high-frequency data. As illustrated in Fig. E, in a continuous scene where both the

Table E. Ablation study on the effect of different angle setups in the proposed Random Platform Jitter (RPJ).

Angle	Vehicle to Quadruiped		Vehicle to Drone	
	AP@0.70	AP@0.50	AP@0.70	AP@0.50
$\pm 0^\circ$	38.61 / 26.84	40.64 / 39.22	57.29 / 36.62	58.92 / 56.19
$\pm 3^\circ$	40.87 / 28.46	44.14 / 41.21	61.95 / 40.52	63.89 / 59.75
$\pm 5^\circ$	42.54 / 30.03	46.54 / 43.02	56.47 / 34.69	56.23 / 54.65
$\pm 8^\circ$	30.55 / 21.41	35.29 / 30.88	41.03 / 26.17	48.65 / 42.58

ego vehicle and surrounding objects are static, the ROS augmentation leads to varying outputs even though the actual scene remains unchanged, whereas detectors without ROS produce much more temporally stable pseudo labels. This inconsistency in the predictions results in a higher rate of false negatives and false positives during pseudo-label generation, thereby misleading the subsequent self-training process.

Consequently, to ensure a fair comparison and maintain stable pseudo label quality, we opted not to apply ROS during the pre-training phase for all our experiments on the Pi3DET dataset. For completeness, we also evaluated variants of ST3D [22] and ST3D++ [23] without ROS during self-training. Our findings underscore that, while ROS can be beneficial in datasets with lower annotation frequencies, its application in high-frequency scenarios like Pi3DET can be counterproductive, and thus must be carefully reconsidered for such settings.

C.1.2. Ablation Study on Random Platform Jitter (RPJ)

In our analysis of cross-platform LiDAR imaging discrepancies, we identified ego motion – specifically, sensor jitter – as a key factor induced by different platform dynamics. Vehicles typically travel on smooth, gently sloping roads, so their 6D ego poses (relative to the world coordinate system) exhibit minimal or gradual changes in pitch and roll.

In contrast, the Quadruiped platform, although also operating on the ground, experiences significant variations in pitch and roll due to mechanical vibrations and unique actions (such as crouching, standing, and turning). The Drone platform, with its greater degrees of freedom, exhibits an even broader distribution of view angles. This motivated our use of Random Platform Jitter during pre-training to simulate these dynamic variations.

To explore the impact of jitter augmentation, we experimented with three settings for randomly rotating the scene around the x and y axes: $\pm 3^\circ$, $\pm 5^\circ$, and $\pm 8^\circ$. Our experiments were conducted using the PV-RCNN model on two cross-platform tasks: Pi3DET (Vehicle) \rightarrow Pi3DET (Quadruiped) and Pi3DET (Vehicle) \rightarrow Pi3DET (Drone). The results are shown in Tab. E.

We observed that a jitter range of $\pm 5^\circ$ yields a 3.2 AP@0.7 gain for the Vehicle-to-Quadruiped task, while a smaller range of $\pm 3^\circ$ is more effective for the Vehicle-to-Drone task, resulting in a 4.3 AP@0.7 gain. We note that

larger jitter angles, such as $\pm 8^\circ$, can cause the point cloud to exceed the pre-defined detection range, limiting their practical utility.

These results indicate that the optimal jitter setting is task-specific and likely depends on the intrinsic sensor placement and motion characteristics of the platform. We believe that while the current settings are effective for the Pi3DET benchmark, other platforms may require tailored augmentation parameters. Moreover, our findings highlight a broader challenge: truly robust 3D detectors should ideally be invariant to viewpoint changes, yet current state-of-the-art models, due to their reliance on regularized point cloud representations, often lose genuine viewpoint robustness from the outset. Future research should continue to explore methods that overcome these limitations.

C.1.3. Ablation Results on Cross-Dataset Task

Tab. F summarizes our cross-dataset adaptation results for the nuScenes \rightarrow Pi3DET (Vehicle) task. In this setting, we compare several state-of-the-art methods using two base detectors, PVRCNN [14] and VoxelRCNN [5], and report AP in both BEV and 3D at IoU thresholds of 0.70 and 0.50. The table reveals several key observations: the Source Only model, trained solely on the nuScenes dataset, suffers from a considerable performance drop when directly applied to the Pi3DET (Vehicle) target dataset, underscoring the significant domain shift.

In contrast, adaptation methods such as ST3D and ST3D++ markedly improve performance by leveraging self-training strategies. Our proposed method, Pi3DET-Net, achieves the highest AP scores among the compared methods on both PVRCNN and VoxelRCNN settings. For instance, under the PV-RCNN configuration, Pi3DET-Net attains an AP of 64.29% in BEV and 54.76% in 3D (at an IoU of 0.70), which is substantially higher than the other methods, and it significantly narrows the gap to the fully supervised target performance. Overall, our method closes a large portion of the performance gap between the Source Only baseline and the Oracle (fully supervised) model.

C.1.4. Cross-Platform 3D Detection Benchmark

In this section, we detail our cross-platform detection benchmark built on the Pi3DET dataset and analyze the performance of several state-of-the-art 3D detection algorithms under the AP@0.5 metric. We evaluate detectors from three design paradigms – point-based, grid-based, and point-grid-based – to comprehensively assess their cross-platform performance.

Dataset Settings. For our experiments, we select the `penno big loop` sequence from the Vehicle platform as the training set, which contains a large number of Vehicle targets to ensure robust feature learning. The test set comprises three platforms: the Vehicle platform uses the `city hall` sequence, the Quadruiped platform uses the `penno short`

Table F. **Comparisons among state-of-the-art 3D detection algorithms for nuScenes \rightarrow Pi3DET (Vehicle) adaptation.** We report the average precision (AP) in “BEV / 3D” at the IoU thresholds of 0.70 and 0.50, respectively. Symbol ‡ denotes algorithms *w.o.* ROS. All scores are given in percentage (%). The **Best** and **Second Best** scores under each metric are highlighted in **Red** and **Blue**, respectively.

Setting	Method	PV-RCNN [15]		Voxel RCNN-C [5]	
		AP@0.70	AP@0.50	AP@0.70	AP@0.50
nuScenes [2] \rightarrow Pi3DET (Vehicle)	Source Dataset	37.84 / 30.20	39.83 / 39.28	45.13 / 34.14	53.27 / 51.20
	SN [20]	23.23 / 14.91	38.27 / 33.51	- / -	- / -
	ST3D [22]	55.40 / 37.92	63.26 / 57.67	50.89 / 39.10	56.83 / 55.32
	ST3D [‡] [22]	56.42 / 44.40	64.11 / 58.37	52.55 / 40.47	58.75 / 56.1
	ST3D++ [23]	58.55 / 47.19	60.23 / 59.72	54.48 / 43.99	60.03 / 57.46
	ST3D++ [‡] [23]	58.93 / 47.34	60.75 / 60.33	53.83 / 44.16	59.59 / 57.05
	REDB [4]	51.65 / 43.50	58.70 / 52.70	- / -	- / -
	MS3D++ [19]	59.48 / 50.83	65.92 / 64.89	56.14 / 47.61	62.58 / 61.50
	Pi3DET-Net	64.29 / 54.76	66.77 / 66.21	57.12 / 48.98	63.36 / 61.03
	Target Platform	70.48 / 62.77	75.28 / 70.13	68.47 / 58.44	73.29 / 68.56

Table G. **Cross-platform 3D detection benchmark.** We report the average precision (AP) in “BEV / 3D” at the IoU thresholds of 0.7. All scores are given in percentage (%). “-C”, “-A” mean detectors with Anchor-based or Center-based detection head.

Category	Method	Vehicle AP@0.50	Quadruped AP@0.50	Drone AP@0.50	Average
Grid-Based Detector	PointPillar [10]	61.39 / 59.86	46.81 / 37.46	56.13 / 49.03	54.78 / 48.78
	SECOND-IOU [21]	62.95 / 60.31	54.63 / 44.31	60.02 / 56.43	59.20 / 53.68
	CenterPoint [26]	62.48 / 60.91	52.79 / 40.88	60.90 / 53.38	58.72 / 51.72
	PillarNet [12]	60.12 / 58.57	46.88 / 36.36	53.82 / 46.29	53.61 / 47.07
	Part A* [16]	64.41 / 63.10	56.07 / 46.89	65.24 / 57.37	61.91 / 55.79
	Transfusion-L [1]	59.28 / 56.77	52.41 / 38.41	59.74 / 48.63	57.14 / 47.94
	HEDNet [27]	57.14 / 54.38	50.56 / 35.77	58.05 / 46.52	55.25 / 45.56
	SAFNet [9]	53.01 / 50.48	48.95 / 36.68	59.80 / 48.77	54.00 / 45.31
	Part A* + Ours	63.21 / 61.47	57.26 / 49.16	67.82 / 60.01	62.76 / 56.88
Point-Based Detector	PointRCNN [13]	51.71 / 51.04	48.45 / 41.50	59.10 / 52.31	53.09 / 48.28
	3DSSD [25]	52.72 / 51.98	52.68 / 43.07	62.32 / 54.63	55.91 / 49.89
	IA-SSD [28]	58.62 / 57.61	68.77 / 56.65	69.50 / 60.10	65.63 / 58.12
	DBQ-SSD [24]	54.28 / 53.87	62.89 / 54.77	65.63 / 58.74	60.93 / 55.79
	PointRCNN + Ours	57.80 / 57.23	49.76 / 45.83	62.53 / 59.25	56.70 / 54.10
Grid-Point Detector	PV-RCNN [14]	67.02 / 66.57	56.37 / 57.64	67.19 / 59.66	63.53 / 61.29
	PV-RCNN-C [14]	60.24 / 60.08	51.58 / 42.12	53.77 / 52.66	55.20 / 51.62
	PV-RCNN++-A [17]	67.59 / 67.20	57.91 / 47.95	67.78 / 60.14	64.43 / 58.43
	PV-RCNN++-C [17]	60.37 / 60.20	51.45 / 48.39	61.90 / 53.12	57.91 / 53.90
	VoxelRCNN-A [5]	70.32 / 66.27	57.31 / 51.50	67.66 / 59.62	65.10 / 59.13
	VoxelRCNN [5]	60.21 / 60.03	52.29 / 49.04	61.91 / 59.86	58.14 / 56.31
	PV-RCNN++ + Ours	66.33 / 65.90	68.15 / 59.20	70.47 / 67.43	68.32 / 64.18

loop sequence, and the Drone platform uses the penno parking 1 and penno parking 2 sequences. These test sequences were collected in scenes similar to those in the training set to provide a fair evaluation of cross-platform detection performance.

Implementation Details. Our training framework is also

built upon the open-source OpenPCDet project⁵ and is executed using two NVIDIA Titan RTX GPUs. For training, 3D detectors are optimized with a batch size of 4 per GPU over 40 epochs, using the Adam optimizer with an initial learning rate of 0.01, weight decay of 0.001, and momentum of

⁵<https://github.com/open-mmlab/OpenPCDet>

0.9. The data augmentation strategy remains consistent with that used for both cross-platform and cross-dataset tasks. In our experiments, we selected the best-performing detector for each category and further enhanced its performance by incorporating Random Platform Jitter. Specifically, we set the rotation range for the Quadruped platform to $\pm 5^\circ$ and for the Drone platform to $\pm 3^\circ$.

Tab. G presents the AP@0.5 results for various detectors. Our analysis yields several key findings:

- Under the AP@0.5 setting, all detectors show improved performance on the Quadruped and Drone platforms, sometimes approaching or even surpassing the results obtained on the Vehicle platform. This indicates that while these detectors have good recall, they still struggle to accurately regress the geometric parameters of the target bounding boxes.
- Detectors that combine grid-based and point-based representations continue to perform well under the AP@0.5 metric, suggesting that the hybrid approach of leveraging both regular (grid) and irregular (point cloud) representations is a highly effective strategy for building high-performance 3D detectors.
- Point-based detectors exhibit relatively balanced performance across platforms, with some even achieving higher AP@0.5 scores on Quadruped and Drone platforms than on the Vehicle platform. For example, IA-SSD achieves an AP@0.5 on the Drone platform that is approximately 2.5% higher than on the Vehicle platform, indicating that architectures based on raw point cloud inputs tend to be less sensitive to viewpoint changes.
- Although IA-SSD shows significantly lower AP@0.7 performance compared to PointRCNN on the Vehicle platform, its AP@0.5 performance is notably higher – especially on the Quadruped and Drone platforms. This suggests that the semantic feature extraction branch in IA-SSD plays a key role in overcoming viewpoint variations.
- We further evaluated the best-performing models across the different detector types by incorporating our proposed Random Platform Jitter (RPJ) data augmentation. Our experiments indicate that RPJ, while causing a slight decrease in performance on the Vehicle platform, significantly enhances cross-platform performance. Specifically, for the Part A* model, RPJ improved the average BEV/3D AP by 0.85% and 1.1%, respectively; PointRCNN saw gains of 3.6% and 5.8%, while PV-RCNN++ improved by 3.9% and 5.8%.

These results demonstrate that although RPJ may slightly reduce performance on the source domain, it effectively boosts cross-platform detection performance by enhancing the model’s robustness to diverse viewing conditions.

Overall, the experimental results under the AP@0.5 setting reveal that although current detectors exhibit strong recall, they often lack the precision needed to accurately

regress bounding box geometries across different platforms. The combination of diverse detector architectures and the RPJ augmentation provides a promising pathway for improving cross-platform 3D detection, offering valuable insights for future research in this challenging domain.

C.2. Additional Qualitative Results

In this section, we present qualitative visualizations for six cross-platform adaptation tasks to further analyze the effectiveness of our proposed method, Pi3DET-Net (see Fig. J through Fig. O).

We compare our results against two state-of-the-art cross-dataset approaches, ST3D++ [23] and MS3D++ [19]. Overall, Pi3DET-Net consistently delivers superior detection performance across all tasks. For example, in Fig. J, ST3D++ fails to detect a target in one scenario, whereas Pi3DET-Net successfully captures the target in its entirety; in contrast, MS3D++ tends to produce false positives.

Similarly, Fig. L illustrates that while both ST3D++ and MS3D++ generate numerous false positives, our method maintains high precision and recall. These qualitative observations, combined with our quantitative analyses, highlight the significant advantages of Pi3DET-Net in cross-platform detection tasks.

C.3. Failure Cases

Although Pi3DET-Net introduces effective strategies to enhance viewpoint robustness in cross-platform detection tasks, certain failure cases reveal limitations and challenges that remain to be addressed.

In some scenarios, when the platform viewpoint becomes excessively distorted, Pi3DET-Net tends to miss detections, as illustrated in Fig. L. This suggests that further improvements in aligning platform feature domains are necessary. Additionally, the method still struggles with long-distance detection; sparse targets at far ranges exhibit significant deviations in feature distribution under viewpoint transformations, leading to degraded performance.

Furthermore, Pi3DET-Net does not achieve true viewpoint invariance; it fundamentally relies on the underlying performance of the base detector. Current state-of-the-art detectors typically depend on regularizing point clouds, which involves pre-defining a sensing range. When significant viewpoint changes occur, for example, a 10° downward tilt can reduce the effective sensing range to under 20 meters due to increased vertical drop in the point cloud (As illustrated in our example Fig. H.), resulting in fewer points being captured within the detection range.

In future work, based on Pi3DET, we plan to develop more effective data augmentation strategies and leverage the intrinsic robustness of point-based approaches to design detectors that achieve true viewpoint invariance without relying on pre-defined sensing ranges.

D. Broader Impact

In this section, we discuss the broader impact of our proposed Pi3DET dataset and the Pi3DET-Net framework, highlighting its contributions to robot perception and beyond. Additionally, we outline potential limitations and areas for future improvements.

D.1. Potential Societal Impact

The Pi3DET dataset and Pi3DET-Net framework hold significant promise for advancing robotic perception and enhancing the safety and efficiency of autonomous systems. By providing a comprehensive benchmark for cross-platform 3D detection, our work can foster the development of detectors that perform robustly in diverse real-world environments.

This progress is critical for a wide array of applications, from autonomous driving and delivery robotics to search and rescue operations, ultimately contributing to improved safety, reduced operational risks, and more efficient resource utilization.

Moreover, the availability of a multi-platform dataset may accelerate innovation in related fields such as surveillance, environmental monitoring, and assistive technologies.

D.2. Potential Limitations

Despite the promising results, several limitations warrant consideration. First, the effectiveness of Pi3DET-Net is still largely dependent on the underlying performance of base detectors, which may constrain its applicability across various sensor types or operational conditions. Second, the current approach relies on predefined sensing ranges and data augmentation strategies (*e.g.*, Random Platform Jitter), which may not generalize optimally to platforms with significantly different sensor configurations or motion patterns.

D.3. Future Directions

Looking ahead, we plan to further enhance cross-platform robustness by exploring novel data augmentation techniques that reduce dependency on fixed sensing ranges and better capture the dynamics of varying platform motions.

In addition, future work will investigate more intrinsically viewpoint-invariant detection architectures, potentially leveraging advances in point-based feature extraction to overcome the limitations of regularized representations. We also aim to extend our framework to other modalities and domains, such as multi-modal sensor fusion detection, to further advance the state of autonomous perception.

Ultimately, we hope that the Pi3DET dataset and our findings will serve as a foundation for developing truly platform-agnostic 3D detection systems.

E. Public Resources Used

In this section, we acknowledge the use of the following public resources, during the course of this work.

E.1. Public Codebase Used

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

- MMEEngine⁶ Apache License 2.0
- MMCV⁷ Apache License 2.0
- MMDetection⁸ Apache License 2.0
- MMDetection3D⁹ Apache License 2.0
- OpenPCSeg¹⁰ Apache License 2.0
- OpenPCDet¹¹ Apache License 2.0
- xtreme1¹² Apache License 2.0

E.2. Public Datasets Used

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

- M3ED¹³ CC BY-SA 4.0
- nuScenes¹⁴ CC BY-NC-SA 4.0
- KITTI¹⁵ CC BY-NC-SA 3.0.

E.3. Public Implementations Used

- nuscenes-devkit¹⁶ Apache License 2.0
- waymo-open-dataset¹⁷ Apache License 2.0
- Open3D¹⁸ MIT License
- PyTorch¹⁹ BSD License
- ROS Humble²⁰ Apache License 2.0
- torchsparse²¹ MIT License

⁶<https://github.com/open-mmlab/mengine>.

⁷<https://github.com/open-mmlab/mmcv>.

⁸<https://github.com/open-mmlab/mmdetection>.

⁹<https://github.com/open-mmlab/mmdetection3d>.

¹⁰<https://github.com/PJLab-ADG/OpenPCSeg>.

¹¹<https://github.com/open-mmlab/OpenPCDet>.

¹²<https://github.com/xtreme1-io/xtreme1>.

¹³<https://m3ed.io>.

¹⁴<https://www.nuscenes.org/nuscenes>.

¹⁵<http://www.cvlibs.net/datasets/kitti>.

¹⁶<https://github.com/nutonomy/nuscenes-devkit>.

¹⁷<https://github.com/waymo-research/waymo-open-dataset>.

¹⁸<http://www.open3d.org>.

¹⁹<https://pytorch.org>.

²⁰<https://docs.ros.org/en/humble>.

²¹<https://github.com/mit-han-lab/torchsparse>.

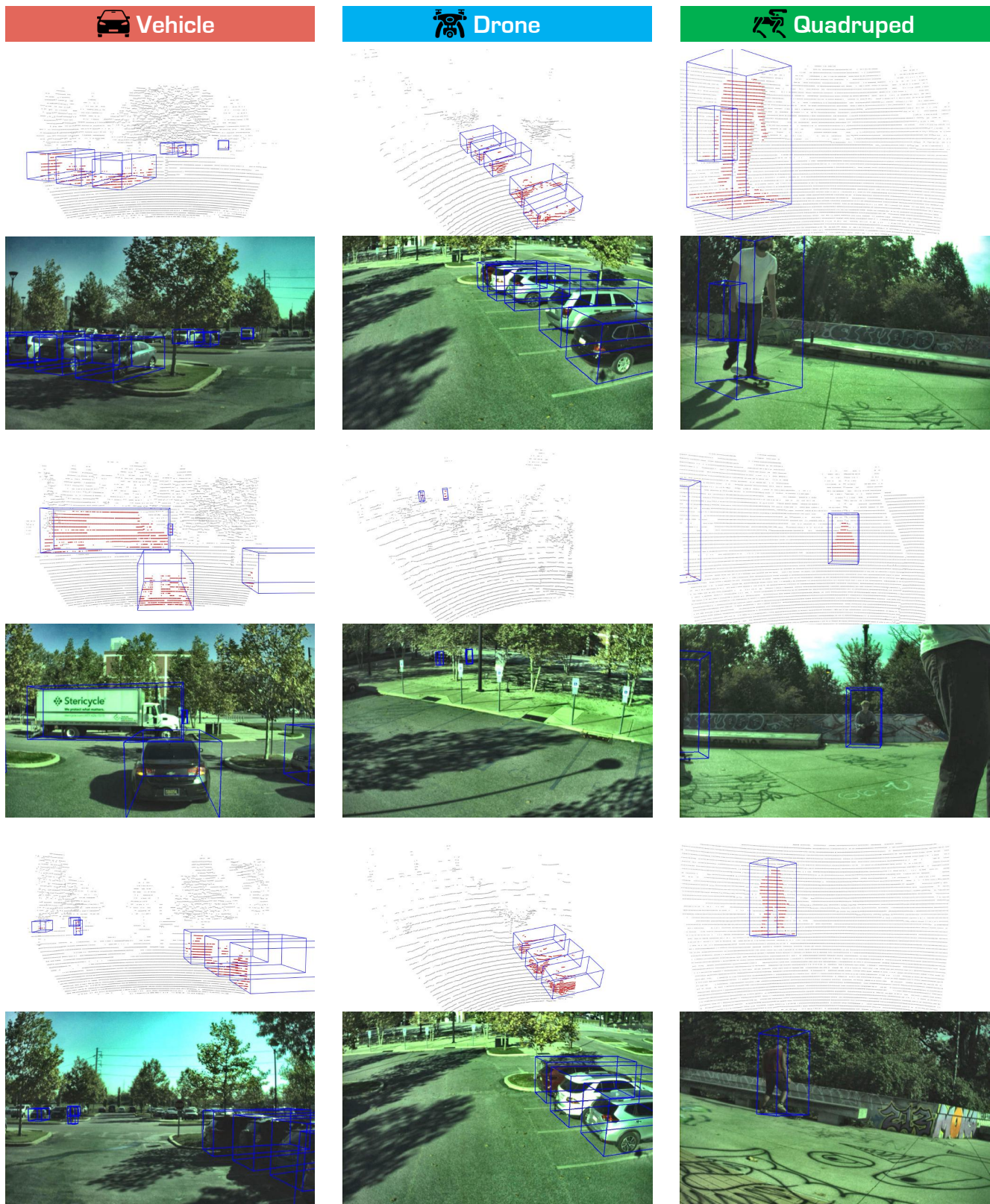


Figure F. Examples of **3D object detection annotations** from 3D (LiDAR point cloud) and 2D (RGB image) in our **Pi3DET** dataset. We provide data from **three robot platforms**: **Vehicle**, **Drone**, and **Quadruped**. Best viewed in colors.

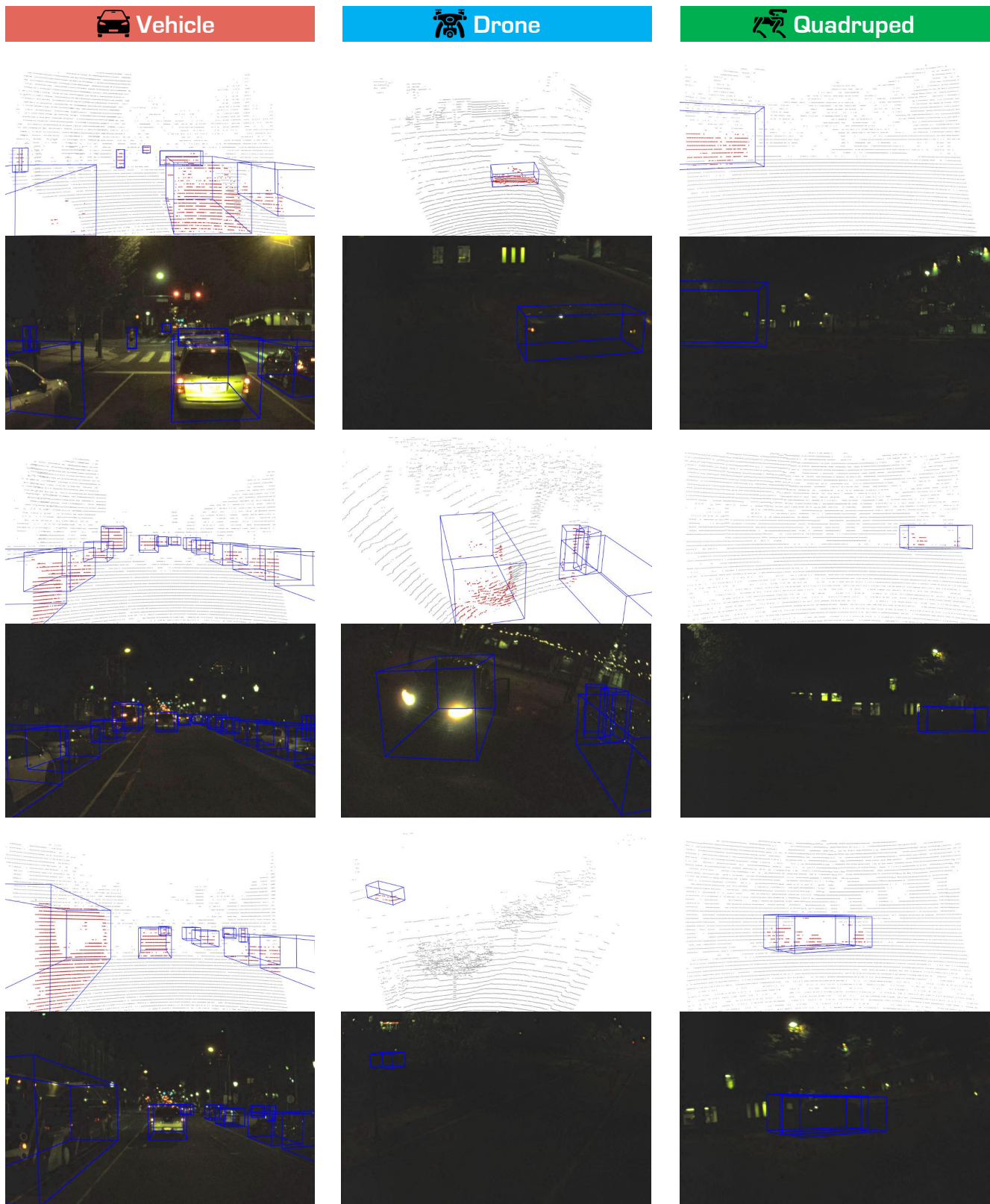


Figure G. Examples of **3D object detection annotations** from 3D (LiDAR point cloud) and 2D (RGB image) in our **Pi3DET** dataset. We provide data from **three robot platforms**: **Vehicle**, **Drone**, and **Quadruped**. Best viewed in colors.

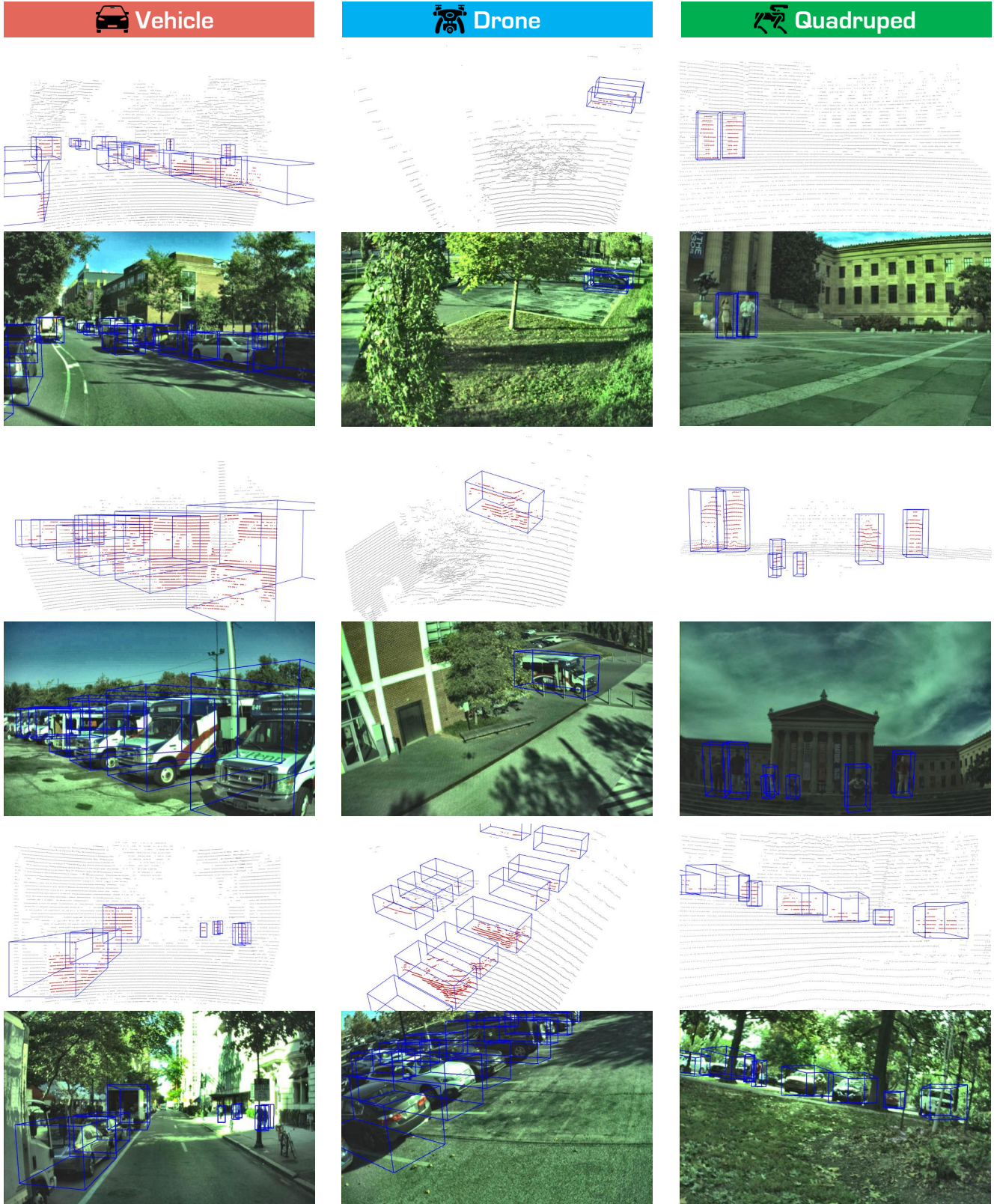



Figure H. Examples of **3D object detection annotations** from 3D (LiDAR point cloud) and 2D (RGB image) in our **Pi3DET** dataset. We provide data from **three robot platforms**: **Vehicle**, **Drone**, and **Quadruped**. Best viewed in colors.



Figure I. Examples of **3D object detection annotations** from 3D (LiDAR point cloud) and 2D (RGB image) in our **Pi3DET** dataset. We provide data from **three robot platforms**: **Vehicle**, **Drone**, and **Quadruped**. Best viewed in colors.

Table H. Summary of **point cloud distribution statistics** (x , y , z , and intensity) of the  **Vehicle** data from the **Pi3DET** dataset.

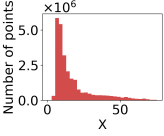
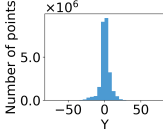
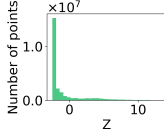
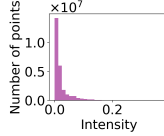
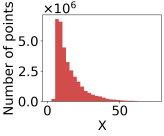
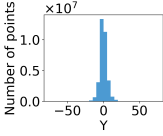
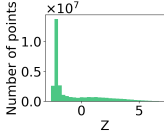
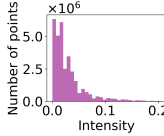
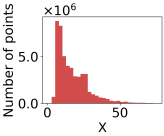
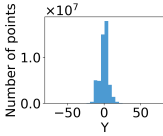
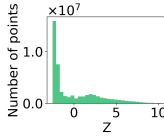
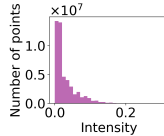
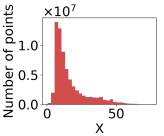
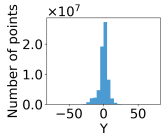
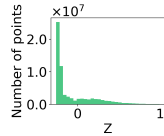
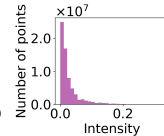
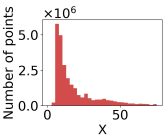
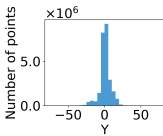
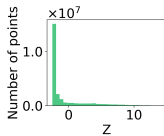
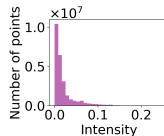
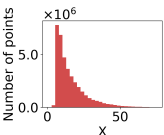
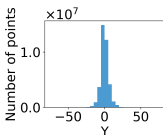
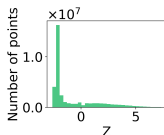
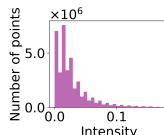
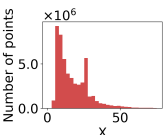
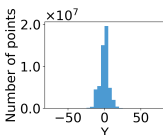
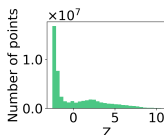
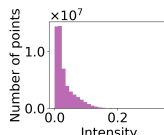
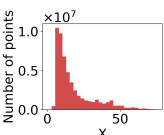
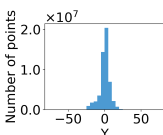
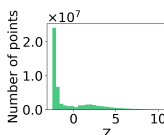
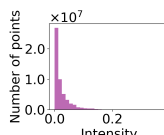
Platform	Condition	Sequence	Point Cloud Distributions (X, Y, Z, Intensity)			
Vehicle (8)	Daytime (4)	city_hall				
		penno_big_loop				
		rittenhouse				
		ucity_small_loop				
	Nighttime (4)	city_hall				
		penno_big_loop				
		rittenhouse				
		ucity_small_loop				

Table I. Summary of **point cloud distribution statistics** (x , y , z , and intensity) of the **Drone** data from the **Pi3DET** dataset.

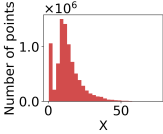
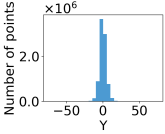
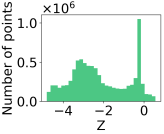
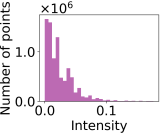
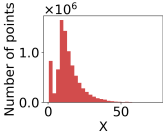
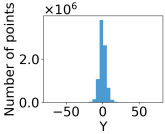
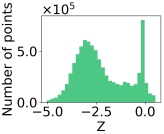
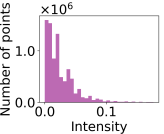
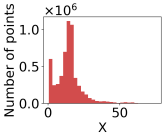
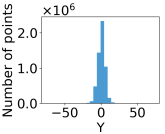
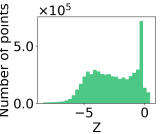
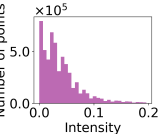
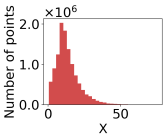
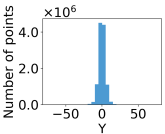
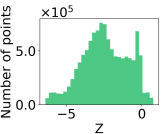
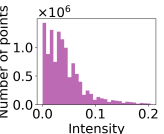
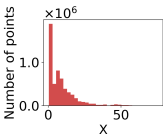
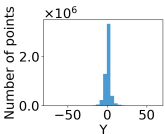
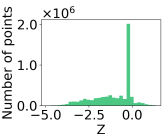
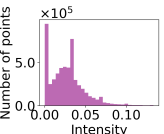
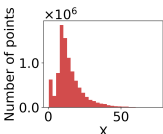
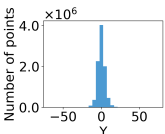
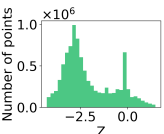
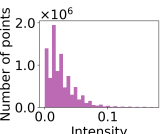
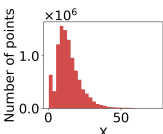
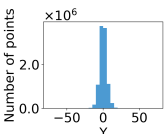
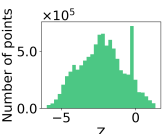
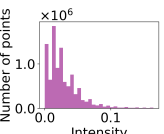

Platform	Condition	Sequence	Point Cloud Distributions (X, Y, Z, Intensity)
Drone (7)	Daytime (4)	penno_parking_1	   
		penno_parking_2	   
		penno_plaza	   
		penno_trees	   
	Nighttime (3)	high_beams	   
		penno_parking_1	   
		penno_parking_2	   

Table J. Summary of **point cloud distribution statistics** (x , y , z , and intensity) of the **Quadruped** data from the **Pi3DET** dataset.

Platform	Condition	Sequence	Point Cloud Distributions (X, Y, Z, Intensity)			
Quadruped (10)	Daytime (8)	art_plaza_loop				
		penno_short_loop				
		rocky_steps				
		skatepark_1				
		skatepark_2				
		srt_green_loop				
		srt_under_bridge_1				
		srt_under_bridge_2				
	Nighttime (2)	penno_plaza_lights				
		penno_short_loop				

Table K. Summary of 3D bounding box statistics (length L , width W , height H) of the  Vehicle data from the Pi3DET dataset.

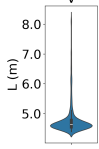
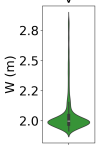
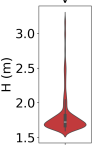
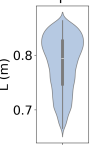
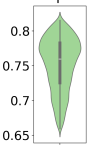
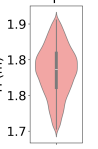
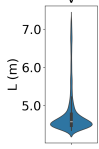
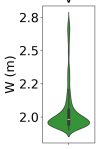
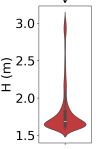
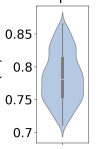
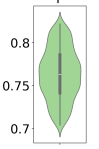
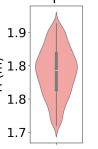
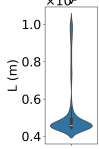
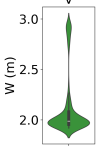
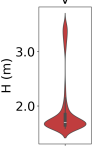
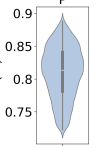
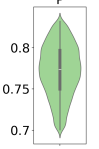
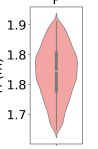
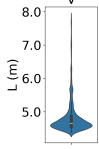
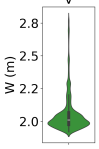
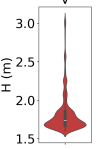
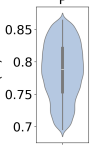
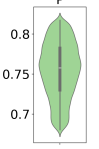
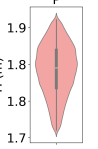
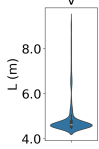
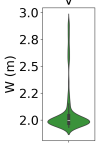
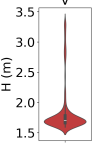
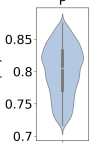
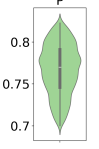
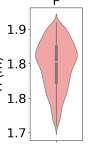
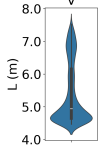
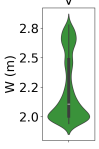
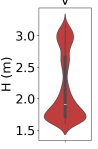
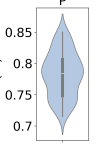
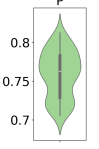
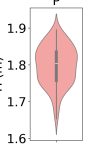
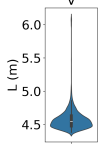
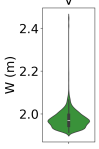
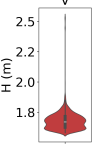
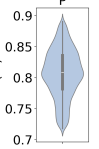
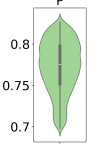
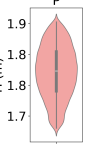
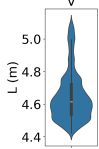
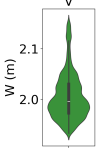
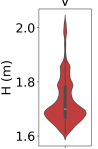
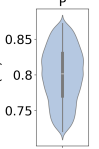
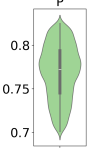
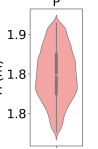
Platform	Condition	Sequence	3D Box Statistics of Veh. (Left) and Ped. (Right)					
Vehicle (8)	Daytime (4)	city_hall						
		penno_big_loop						
		rittenhouse						
		ucity_small_loop						
	Nighttime (4)	city_hall						
		penno_big_loop						
		rittenhouse						
		ucity_small_loop						

Table L. Summary of **3D bounding box statistics** (length L , width W , height H) of the **Drone** data from the **Pi3DET** dataset.

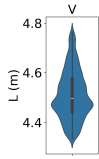
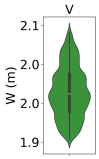
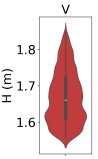
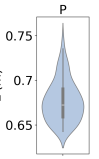
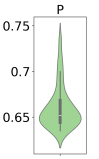
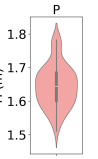
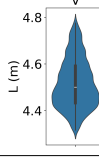
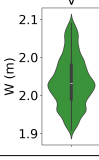
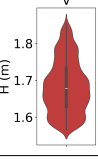
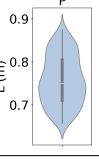
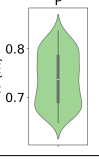
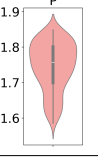
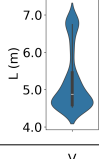
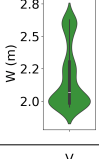
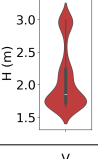
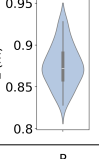
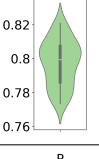
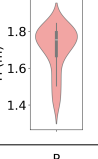
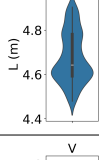
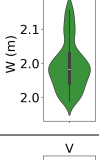
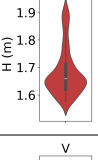
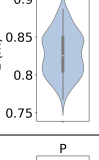
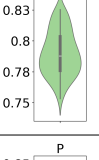
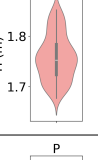
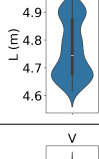
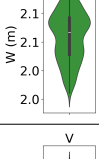
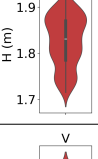
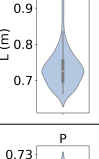
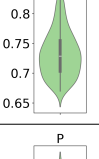
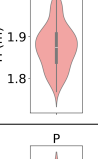
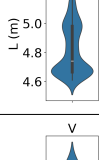
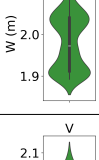
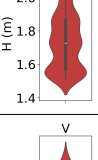
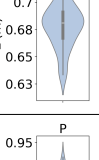
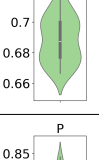
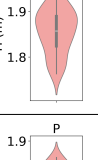
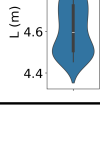
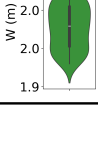
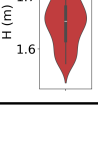
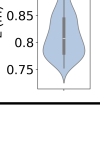
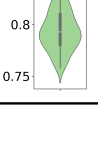
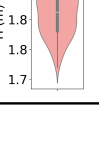

Platform	Condition	Sequence	3D Box Statistics of Veh. (Left) and Ped. (Right)					
Drone (7)	Daytime (4)	penno_parking_1						
		penno_parking_2						
		penno_plaza						
		penno_trees						
	Nighttime (3)	high_beams						
		penno_parking_1						
		penno_parking_2						

Table M. Summary of **3D bounding box statistics** (length L , width W , height H) of the **Quadruped** data from the **Pi3DET** dataset.

Platform	Condition	Sequence	3D Box Statistics of Veh. (Left) and Ped. (Right)					
Quadruped (10)	Daytime (8)	art_plaza_loop						
		penno_short_loop						
		rocky_steps						
		skatepark_1						
		skatepark_2						
		srt_green_loop						
		srt_under_bridge_1						
		srt_under_bridge_2						
	Nighttime (2)	penno_plaza_lights						
		penno_short_loop						

Table N. Summary of **3D object statistics** (objects per frame and points per box) of the  **Vehicle** data from the **Pi3DET** dataset.

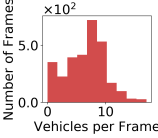
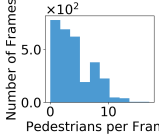
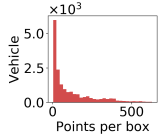
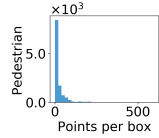
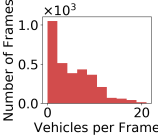
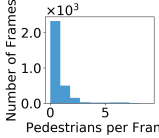
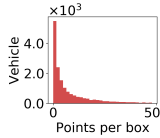
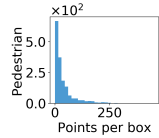
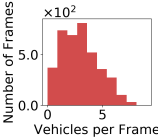
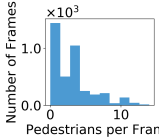
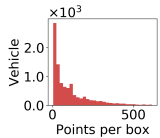
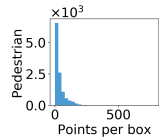
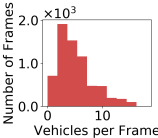
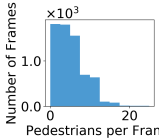
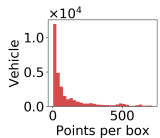
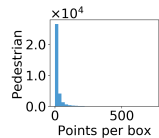
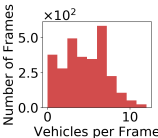
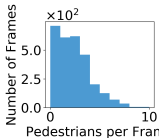
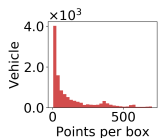
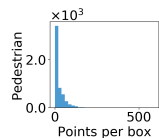
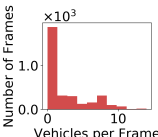
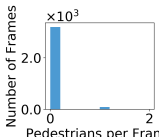
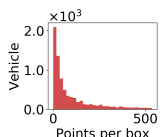
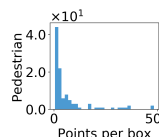
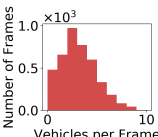
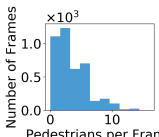
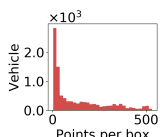
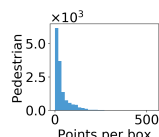
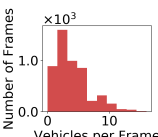
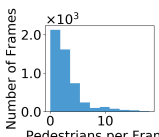
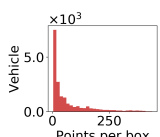
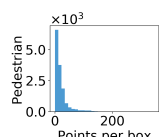
Platform	Condition	Sequence	Objects Per Frame (Left) and Points Per Box (Right)			
Vehicle (8)	Daytime (4)	city_hall				
		penno_big_loop				
		rittenhouse				
		ucity_small_loop				
	Nighttime (4)	city_hall				
		penno_big_loop				
		rittenhouse				
		ucity_small_loop				

Table O. Summary of **3D object statistics** (objects per frame and points per box) of the **Drone** data from the **Pi3DET** dataset.

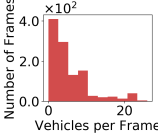
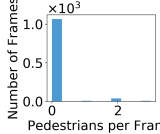
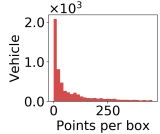
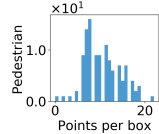
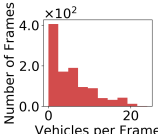
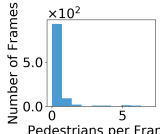
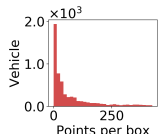
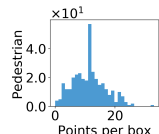
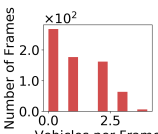
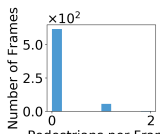
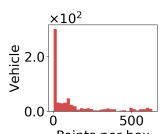
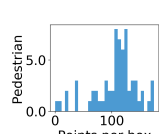
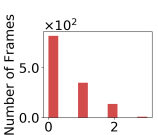
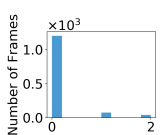
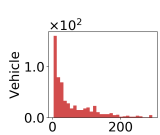
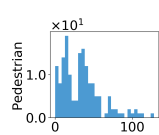
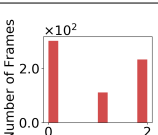
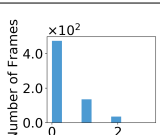
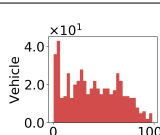
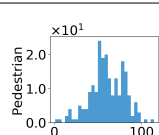
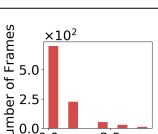
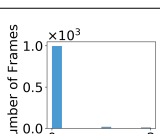
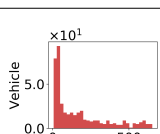
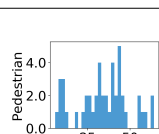
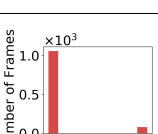
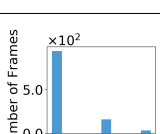
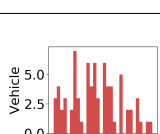
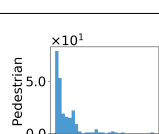
Platform	Condition	Sequence	Point Cloud Distributions (X, Y, Z, Intensity)			
Drone (7)	Daytime (4)	penno_parking_1				
		penno_parking_2				
		penno_plaza				
		penno_trees				
	Nighttime (3)	high_beams				
		penno_parking_1				
		penno_parking_2				

Table P. Summary of **3D object statistics** (objects per frame and points per box) of the **Quadruped** data from the **Pi3DET** dataset.

Platform	Condition	Sequence	Point Cloud Distributions (X, Y, Z, Intensity)			
Quadruped (10)	Daytime (8)	art_plaza_loop				
		penno_short_loop				
		rocky_steps				
		skatepark_1				
		skatepark_2				
		srt_green_loop				
		srt_under_bridge_1				
		srt_under_bridge_2				
	Nighttime (2)	penno_plaza_lights				
		penno_short_loop				

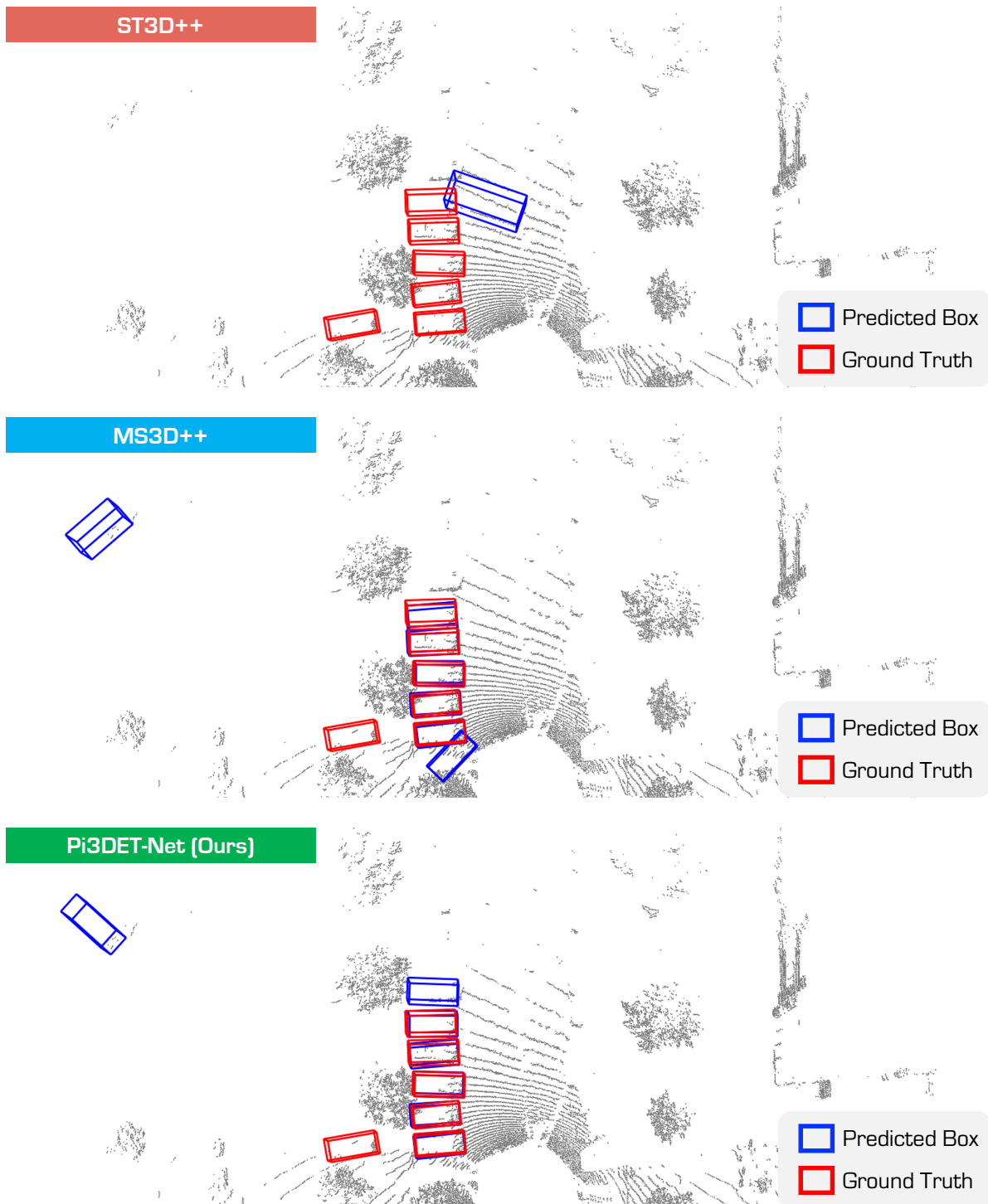


Figure J. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Vehicle)** to **Pi3DET (Drone)**. Best viewed in colors.

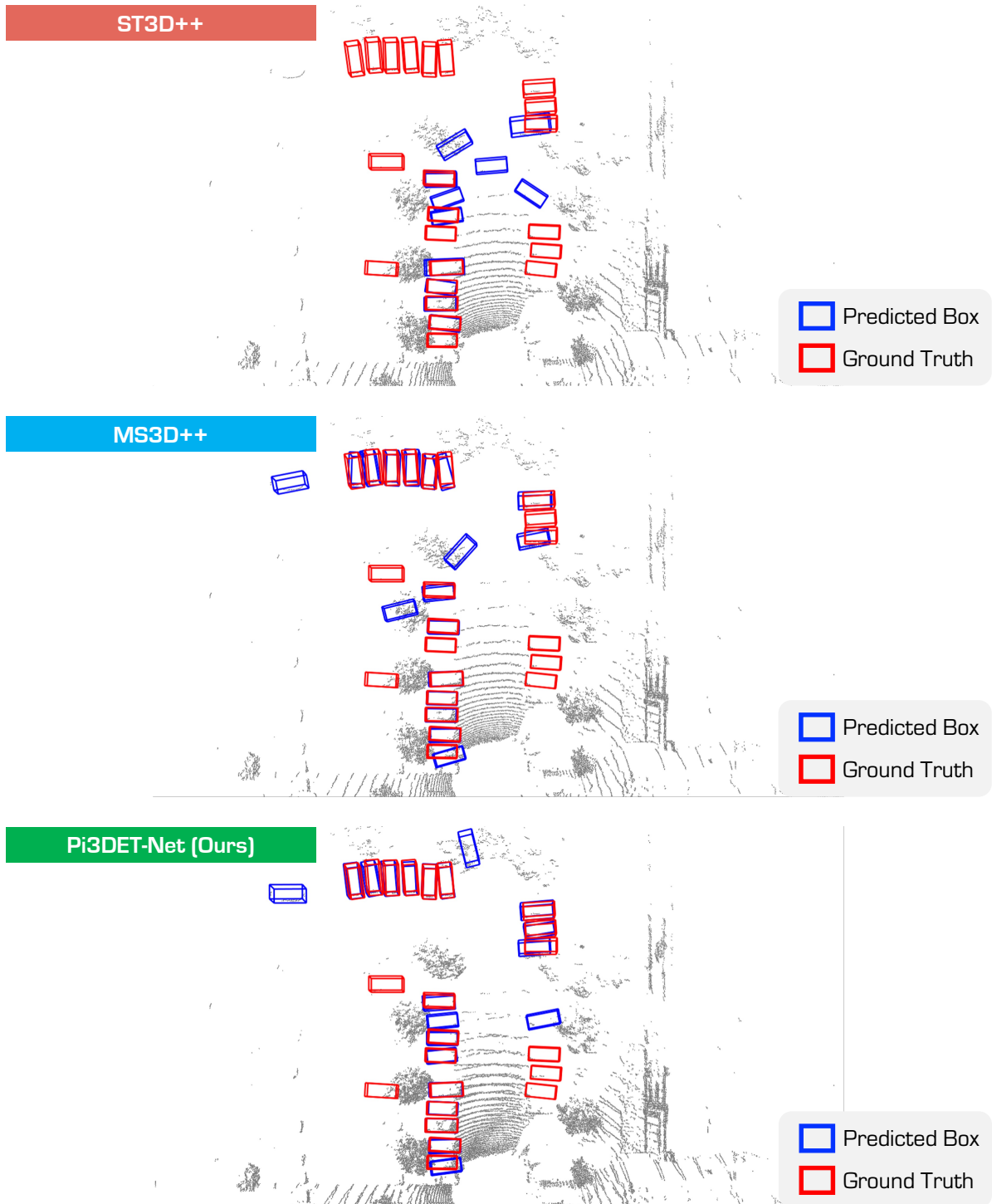


Figure K. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Vehicle)** to **Pi3DET (Drone)**. Best viewed in colors.

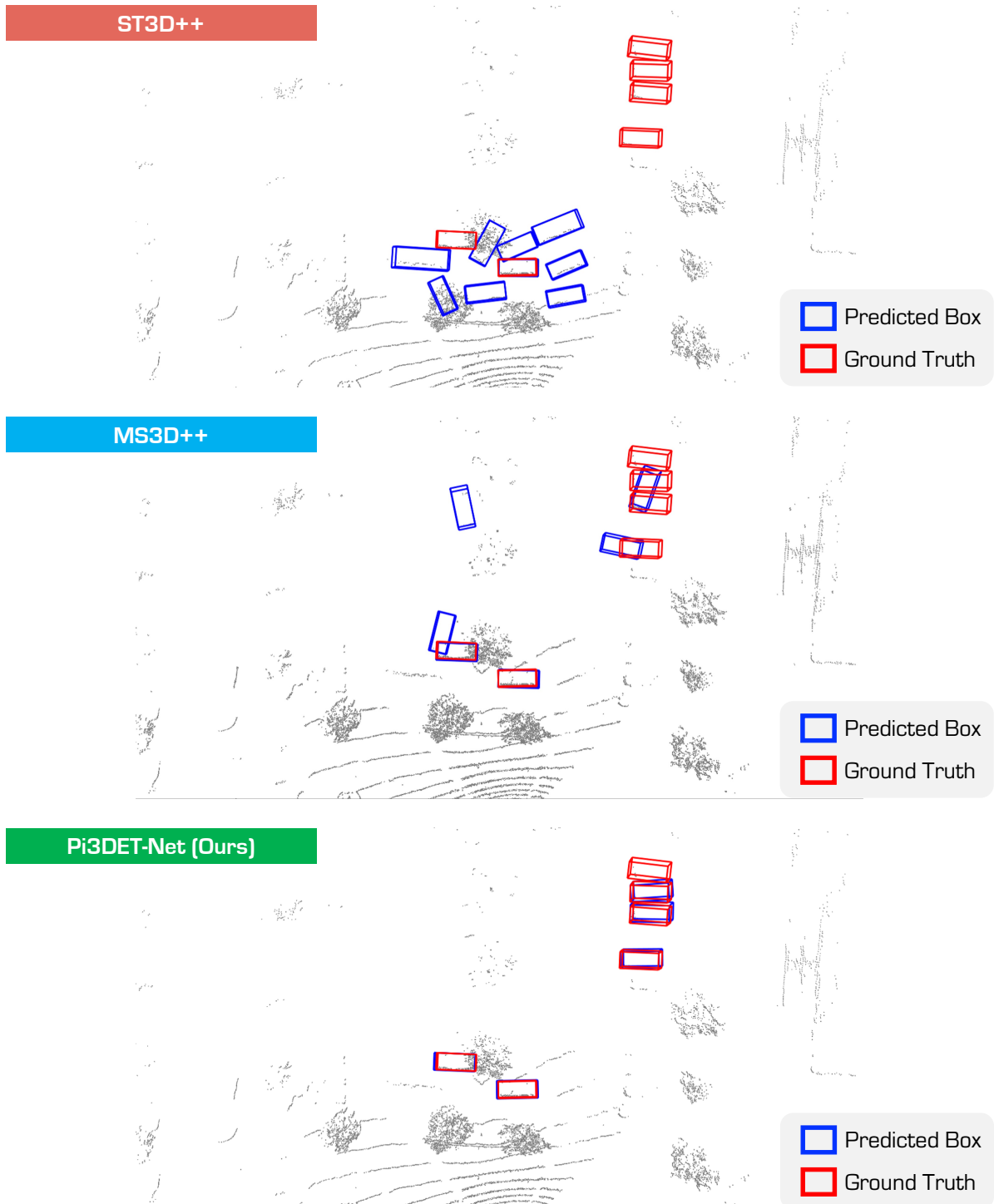


Figure L. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Vehicle)** to **Pi3DET (Quadruped)**. Best viewed in colors.

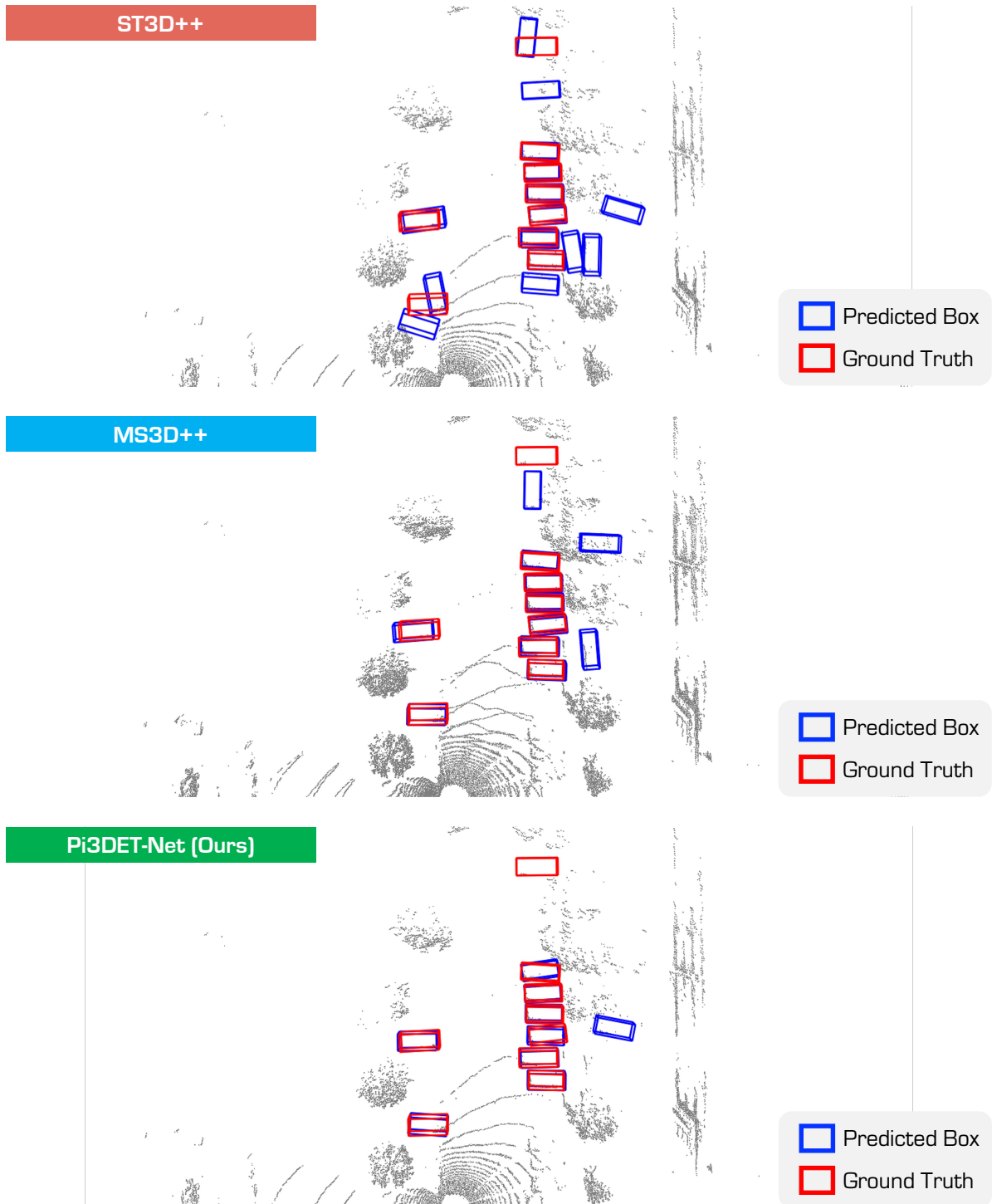


Figure M. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Vehicle)** to **Pi3DET (Quadruped)**. Best viewed in colors.

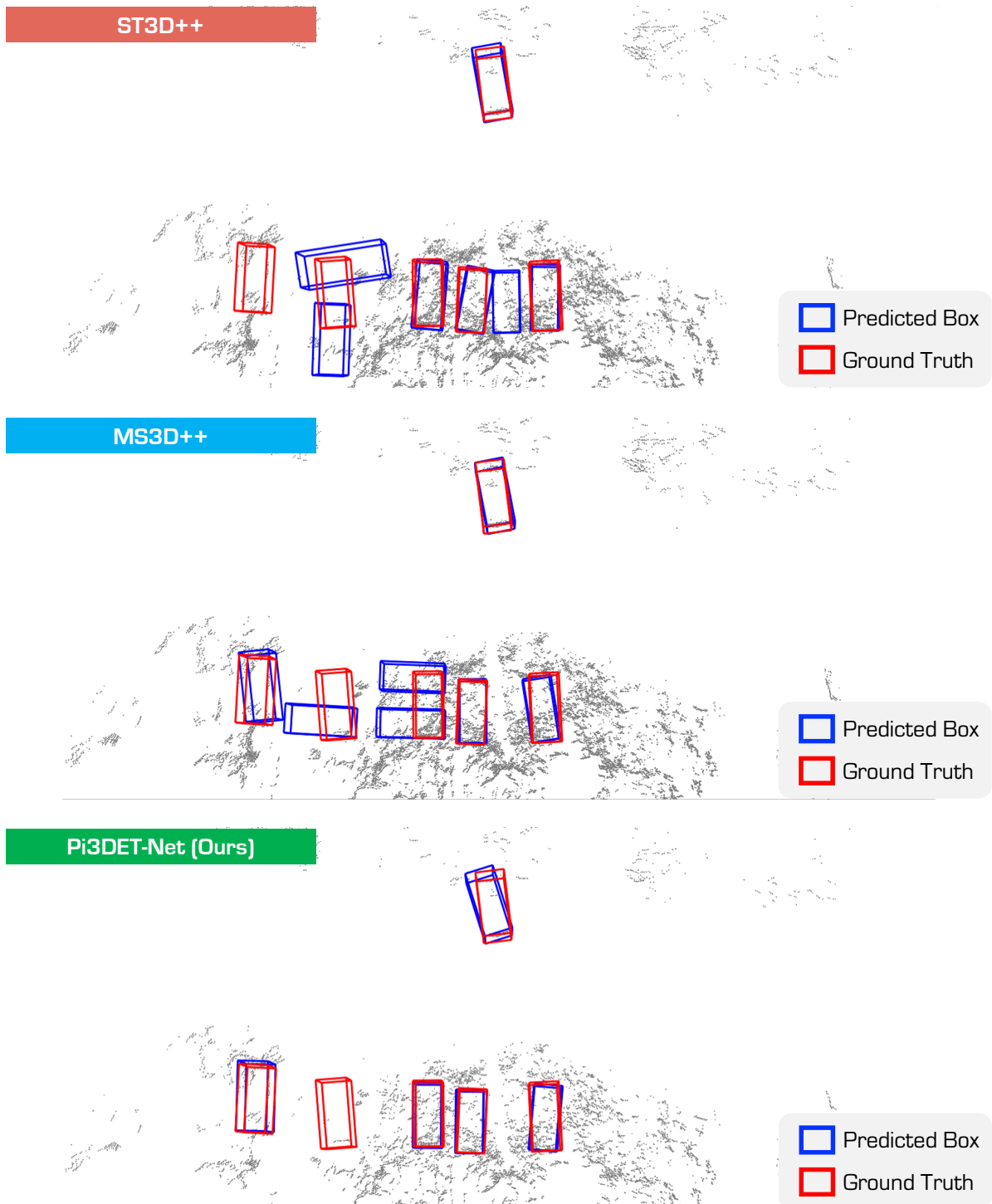


Figure N. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Drone)** to **Pi3DET (Quadruped)**. Best viewed in colors.

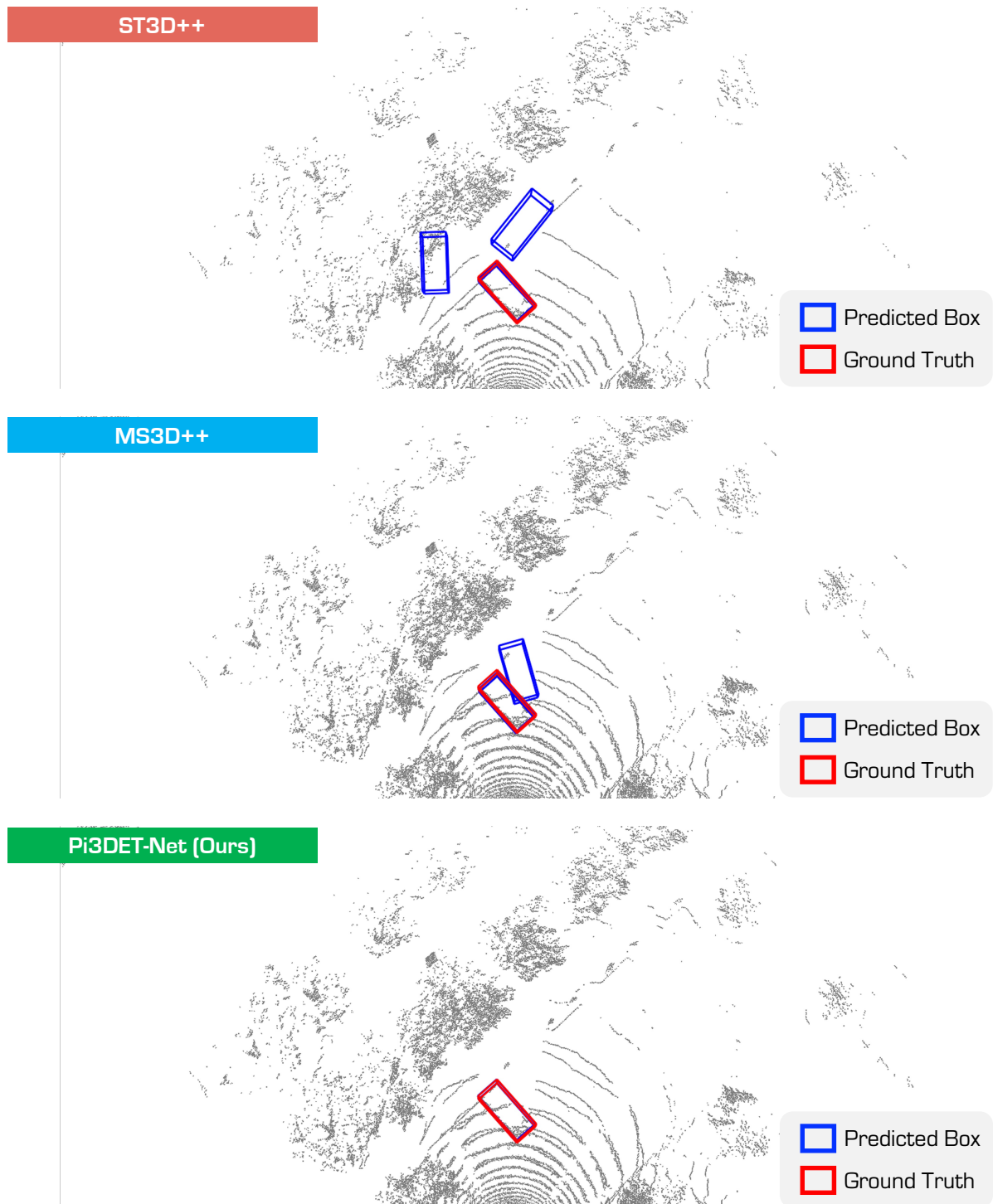


Figure O. Qualitative results from state-of-the-art methods. We compare **Pi3DET-Net** with ST3D++ [23] and MS3D++ [19]. The figure illustrates predictions from methods that are adapted from **Pi3DET (Drone)** to **Pi3DET (Quadruped)**. Best viewed in colors.

References

- [1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 11
- [2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 3, 4, 7, 9, 11
- [3] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo J Taylor, and Kostas Daniilidis. M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 4016–4023, 2023. 1
- [4] Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi Huang. Revisiting domain-adaptive 3d object detection by reliable, diverse and class-balanced pseudo-labeling. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3714–3726, 2023. 9, 11
- [5] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, pages 1201–1209, 2021. 4, 7, 8, 10, 11
- [6] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19820–19829, 2023. 4
- [7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 3, 4, 7, 9
- [8] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021. 4, 9
- [9] Lingtong Kong, Bo Li, Yike Xiong, Hao Zhang, Hong Gu, and Jinwei Chen. Safnet: Selective alignment fusion network for efficient hdr imaging. In *European Conference on Computer Vision*, pages 256–273. Springer, 2024. 11
- [10] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 11
- [11] Ting Pan, Lulu Tang, Xinlong Wang, and Shiguang Shan. Tokenize anything via prompting. In *European Conference on Computer Vision*, pages 330–348. Springer, 2024. 4, 6
- [12] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *European Conference on Computer Vision*, pages 35–52. Springer, 2022. 11
- [13] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointnet-cnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 11
- [14] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 4, 7, 8, 9, 10, 11
- [15] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 11
- [16] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2647–2664, 2020. 11
- [17] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023. 4, 11
- [18] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 4, 9
- [19] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Ms3d++: Ensemble of experts for multi-source unsupervised domain adaptation in 3d object detection. *IEEE Transactions on Intelligent Vehicles*, pages 1–16, 2024. 8, 11, 12, 27, 28, 29, 30, 31, 32
- [20] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Weilun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 11
- [21] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 4, 7, 8, 9, 11
- [22] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021. 7, 8, 9, 10, 11
- [23] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsupervised

- domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6354–6371, 2022. [8](#), [9](#), [10](#), [11](#), [12](#), [27](#), [28](#), [29](#), [30](#), [31](#), [32](#)
- [24] Jinrong Yang, Lin Song, Songtao Liu, Weixin Mao, Zeming Li, Xiaoping Li, Hongbin Sun, Jian Sun, and Nanning Zheng. Dbq-ssd: Dynamic ball query for efficient 3d object detection. *arXiv preprint arXiv:2207.10909*, 2022. [11](#)
- [25] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. [11](#)
- [26] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. [4](#), [11](#)
- [27] Gang Zhang, Chen Junnan, Guohuan Gao, Jianmin Li, and Xiaolin Hu. Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. *Advances in Neural Information Processing Systems*, 36:53076–53089, 2023. [11](#)
- [28] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. [4](#), [11](#)