WiSAR3D - Aerial LiDAR dataset for 3D object detection

Oren Shrout Ori Nizan Yizhak Ben-Shabat Ayellet Tal

Technion, Israel Institute of Technology

{shrout.oren@campus.,snizori@campus.,sitzikbs@,ayellet@ee.}technion.ac.il

https://wisar3d.github.io/

Abstract

Wilderness Search and Rescue (WiSAR) operations aim to locate and rescue individuals in remote, rugged environments. We introduce WiSAR3D, the first aerial LiDAR 3D dataset specifically tailored for 3D object detection in rural scenes. WiSAR3D comprises 2633 *strips—contiguous point clouds collected along the flight path, all fully annotated with* $67.5K$ *3D bounding boxes for* 22 *categories. As the data is captured with multiple returns (echoes), it enables detection under foliage, which is impossible with RGB cameras. Additionally, as this is the first-of-its-kind dataset, we provide benchmark evaluations for state-of-the-art methods developed for autonomous cars on our aerial dataset. The results suggest that specialized models are needed in this domain, as finding individuals remains challenging for current 3D detectors. The code and dataset are publicly available at https://github.com/oshrout/WiSAR3D.*

1. Introduction

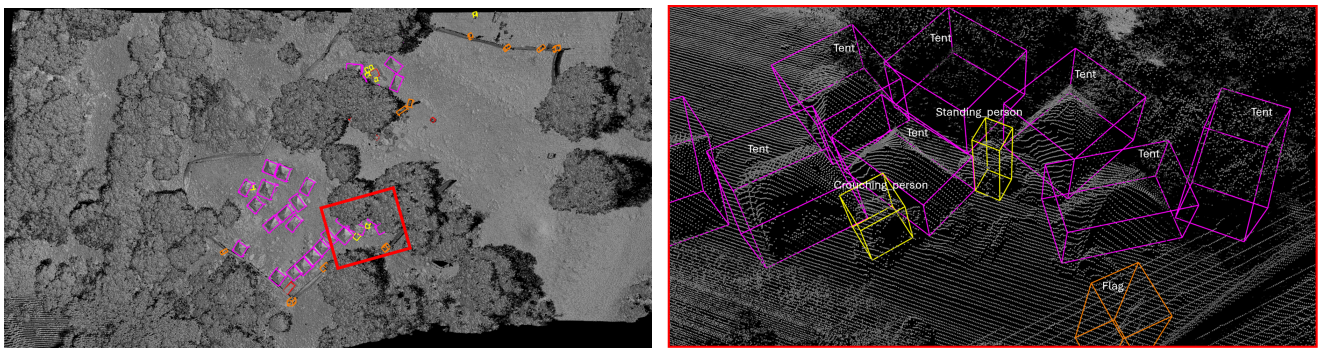
Wilderness Search and Rescue (WiSAR) refers to the specialized field of search and rescue operations conducted in remote, rugged, and often uninhabited wilderness areas. These operations typically involve locating and rescuing individuals who are lost or incapacitated. Since minimizing search time can be critical for survival, autonomous systems, such as drones equipped with advanced sensors, can significantly accelerate search efforts by rapidly covering large areas, collecting real-time data, and helping to prioritize search zones.

While several 2D search and rescue datasets exist [3–6, 11, 18, 20, 31], they rely on RGB images of people in varied poses, environments, and lighting conditions. However, RGB sensors struggle in challenging WiSAR scenarios [6], such as low light and occlusions in forested landscapes. LiDAR offers a robust alternative, operating reliably at all times of day—including dawn and dusk—in diverse weather conditions, and penetrating foliage to detect obscured objects in forests. This capability is crucial for WiSAR, which demands rapid, large-area coverage, often from high altitudes and under varied conditions where RGB may fall short.

We introduce *WiSAR3D*, the first aerial LiDAR 3D dataset for WiSAR applications. Our dataset differs from existing LiDAR datasets, which primarily focus on urban and rural scenarios involving either autonomous vehicles on roadways [7, 8, 12, 17, 23, 27] or aerial data for semantic segmentation [13, 15, 25, 32]. While these datasets have been invaluable for advancing object detection and scene recognition in urban environments, their applicability to WiSAR scenarios is limited due to differences in environmental conditions, object diversity, mobility requirements, and the availability of relevant training data. Instead, we propose a dataset tailored to wilderness search and rescue operations, facilitating the development of algorithms that address the unique challenges.

WiSAR3D comprises records from a LiDAR scanner mounted on a drone. Two major properties distinguish it from existing datasets. First, the scenes are captured in off-road, natural wilderness environments, which are absent from existing 3D datasets. These recordings include natural elements commonly found in nature, such as trees, bushes, and rocks, as well as man-made objects indicating human presence, including human figures. Second, unlike datasets for autonomous driving, which typically consist of single-return frames, our data consists of *strips*—contiguous point clouds collected along the flight path, featuring multiple returns (echoes). An example strip is shown in Fig. 1.

Our dataset comprises 2633 strips (1–10 seconds each), totaling 5.1 hours—comparable to nuScenes [7], which spans 5.5 hours over $40K$ frames. It includes $67.5K$ high-quality, manually annotated 3D boxes across 22 categories, focusing on humans and man-made objects, which are rare in natural environments. To simulate indirect cues of human presence, which have been shown to support search ef-



(a) A typical scene

(b) Zoom in on the red rectangle

Figure 1. **Input.** A typical strip is shown in (a), alongside a zoomed-in perspective in (b), revealing multiple tents (magenta boxes) and two individuals (yellow boxes) assembling tents in a forest. Some tents are concealed beneath the foliage.

forts [16], we strategically placed objects in diverse scenes. These range from small items like *cones*, *balls*, and *pipes* to larger structures like *tents*, *vehicles*, and *canopies*.

For robustness and realistic scenarios, we varied the drone's altitude and tested three velocities. Another unique aspect of our dataset is that we scanned the scenes with four different sampling rates. This feature enables us to investigate the impact of sampling rates on detection. As far as we know, this dataset is the first to provide multiple sampling rates and analyze their importance. We show that incorporating low rates alongside high rates into training improves detection accuracy, especially when validated on low-rate datasets.

In our benchmark, the dataset is partitioned into training and validation sets, ensuring comprehensive coverage of height altitudes, flight velocities, and sampling rates in both sets. Evaluation employs the widely used Average Precision (AP) metric, and four adapted metrics based on those from nuScenes [7], modified for our dataset.

Given the lack of domain-specific models, we evaluated SoTA 3D object detectors developed for autonomous driving, adapting them to our dataset by tuning parameters such as input point count, voxel size, point cloud range, and model capacity. The results highlight the need for specialized models: existing methods achieve only 66.54% mAP while requiring 28.69 GB of GPU memory at inference—rendering them less practical for real-world deployment—or 60.76% mAP with a more manageable memory footprint of 5.16 GB. Performance is especially limited for humans, with mAP scores of just 54.80% and 45.01% for *standing* and *sitting person*, due to their deformable nature. Small objects like *cone* and *pipe* also show reduced performance due to sparse point representations.

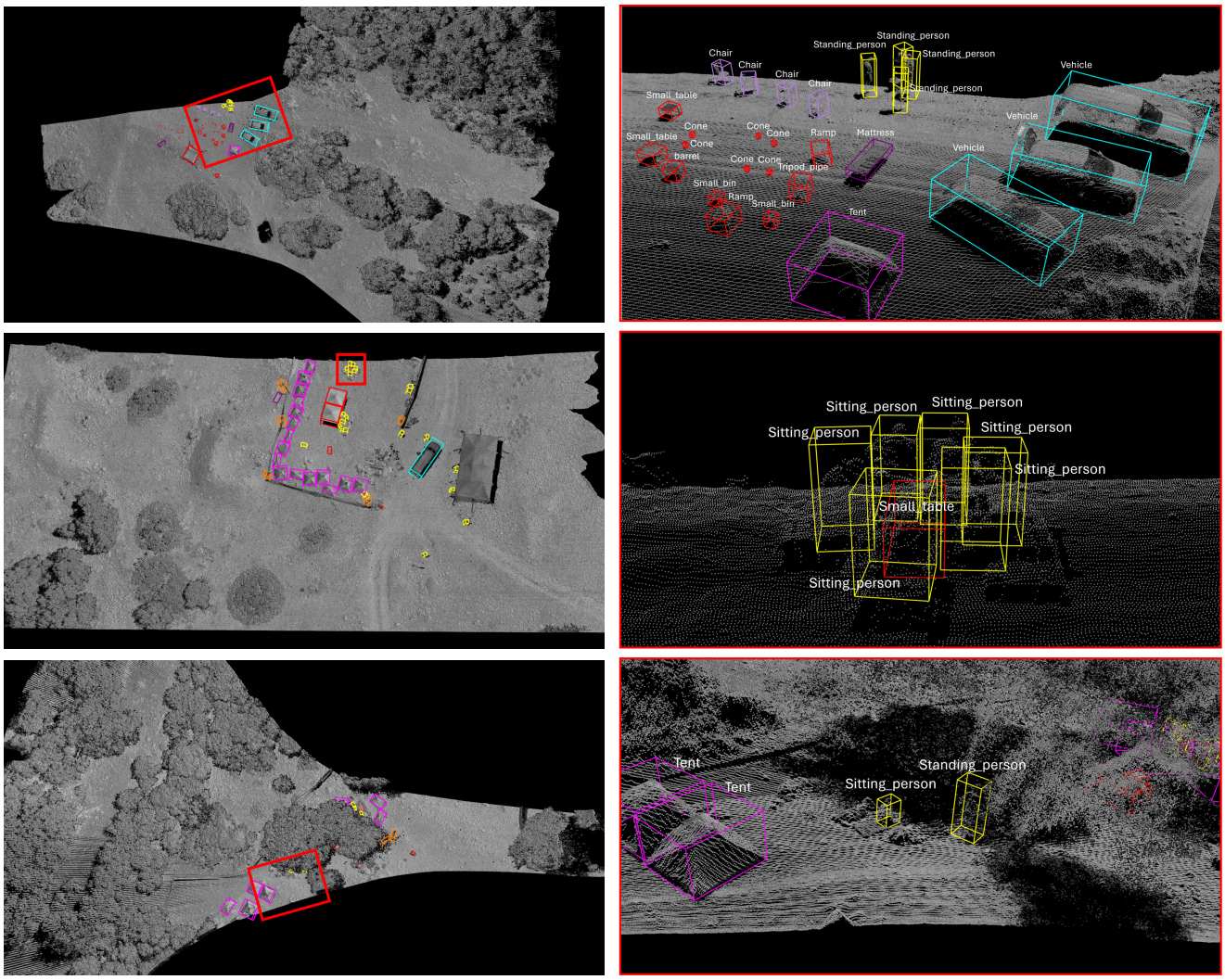
Our main contributions can be summarized as follows:

- We introduce *WiSAR3D*, the first aerial LiDAR 3D object detection dataset, specifically designed for *Wilderness Search and Rescue* applications. The dataset features multiple LiDAR returns, demonstrating the importance of detecting objects concealed beneath foliage, and it captures a range of altitudes, velocities, and sampling rates (see Fig. 2).
- We introduce a benchmark for 3D detection in WiSAR scenarios and adapt several state-of-the-art detectors commonly used in autonomous driving to our dataset.
- We provide insights into the significance of the sensor's sampling rates for accurate detection.

2. Related Work

Outdoor 3D datasets. Existing outdoor datasets are designed for object detection and semantic segmentation in autonomous driving or aerial semantic segmentation. Autonomous driving datasets, such as KITTI [12], Argoverse [8, 27], nuScenes [7], Waymo [23], and ONCE [17], provide extensive, diverse, and annotated data collected from sensors mounted on vehicles. These datasets have significantly advanced the development and assessment of algorithms capable of accurately detecting and categorizing objects in real-world driving scenarios. However, their applicability to WiSAR scenarios is limited due to differences in environmental conditions, object diversity, and mobility requirements. Aerial semantics segmentation datasets like DublinCity [32], DALES [25], Campus3D [15], and SensatUrban [13] offer aerial perspectives. These datasets focus on large spatial coverage from overlapping flight paths and include annotations for categories like ground, vegetation, buildings, and cars. However, they lack annotations relevant to WiSAR scenarios, such as humans.

WiSAR datasets. Existing datasets suitable for Wilderness Search and Rescue applications are scarce, and all of them are confined to the 2D domain [3, 5, 6, 20]. In Božić-Štulić et al. [5], cropped images of individuals in diverse poses, positions, environments, and under differing lighting conditions were collected, along with synthetic compositions cre-



(a) Zoom out

(b) Zoom in

Figure 2. **Dataset.** Several strips are shown in (a), with zoomed-in views of the red rectangles presented in (b). The strips highlight different spatial ranges and point counts, while the zoomed-in perspectives show various scanning patterns (the lines) produced by different flight paths. The top strip illustrates a variety of classes and object sizes. The middle and bottom strips emphasize human figures (in yellow), depicting them in different scenarios, such as a group seated around a table (middle) or taking shelter under foliage (bottom).

ated by merging elements from multiple images. Sambolek et al. [20] assembled a collection of images and videos featuring actors simulating various movements and scenarios, complemented by artificially generated imagery of different weather conditions. Broyles et al. [6]'s dataset contains visual and thermal imagery from a broad spectrum of wilderness landscapes, including various terrains, lighting, and weather conditions. Bernal et al. [3] assembled a dataset of videos that highlight occlusion scenarios. The lack of relevant 3D datasets limits the aid available to search and rescue teams, as they fail to account for important factors like operation time due to lighting and weather conditions, and occlusions in forested environments.

3. WiSAR3D Dataset

The main challenges encountered in WiSAR applications include the lack of annotations for identifying signs of human presence [16], lighting conditions [6], and obstructions created by foliage [3]. Our data consists of LiDAR strips, which can be effectively captured throughout the day and night, allowing search teams to conduct operations at any time. This sets it apart from most datasets, which are limited to daylight. This is important as the search duration is crucial in determining whether the rescue is successful. Furthermore, LiDAR technology offers the ability to partially penetrate foliage by capturing multiple returns/echoes per

(a) 1st return (b) 2nd return (c) 3rd return (d) All returns

Figure 3. **Tent under foliage.** 84% of the points are discovered in the high returns (> 1), indicating the multiple returns' importance. The first return represents points obtained from the open area, while the higher returns display points that penetrated the foliage once (sparse foliage), twice, and more.

shot. When the laser pulse encounters foliage, it can penetrate through gaps or spaces between leaves and branches, allowing it to interact with multiple surfaces within the canopy. This capability allows for the detection of objects obscured by foliage in vegetation-rich environments.

Fig. 2 illustrates typical strips in our dataset, demonstrating the variety of classes and object sizes. The humans are either standing in groups, seated around tables, or solitary. Furthermore, the strips exhibit various scanning patterns (generated by different flight paths), spatial ranges, point counts, and object quantities within each strip. Fig. 3 illustrates that increasing the number of LiDAR returns improves representation, as demonstrated by a tent largely obscured by foliage. In (a), with only one return, the right side and front of the tent are barely visible, whereas in (d), using all returns, the tent is much more clearly recognizable.

3.1. Sensor Specifications & Data Acquisition

The data was collected using an industrial-grade LiDAR sensor, the *RIEGL VUX-120*, which operates at a high measurement rate and supports multiple returns per laser pulse. This high rate enables the efficient scanning of complex objects, including small items and humans, while the multiple returns help detect objects obscured by foliage. Emitting its measuring beam (scan pattern) in three directions: nadir (directly below), $+10^\circ$ forward, and -10° backward, RIEGL VUX-120 enables data collection in challenging environments with vertical surfaces. The sensor offers a Pulse Repetition Rate (PRR), akin to a sampling rate, of up to 2400 kHz with an accuracy of 10 mm, generating up to 2 million points per second within a $\pm 50^\circ$ field of view.

The sensor was mounted on a *DJI Matrice 600* drone and flown over undeveloped wilderness areas. To capture diverse and realistic data, we conducted flights at 22 altitudes (30–160 meters), using three velocities (3, 4, and 5 m/s) and four Pulse Repetition Rates (PRRs): 300, 600, 1200, and 1800 kHz. For each combination of altitude, velocity, and PRR, we collected multiple *LiDAR strips*—contiguous point clouds acquired along the flight path. Flights followed a crisscross pattern, both vertically and horizontally, typically producing one strip per path. Each strip comprises consecutive scan lines captured continuously during flight, forming a dense point cloud.

The collected data is initially in the world coordinate system. However, to promote model generalization (rather than memorization of strip locations), we transform each strip into its own local coordinate system. In this local system, the X-axis follows the average flight trajectory, the Y-axis points to the left side of the drone, and the Z-axis extends upward from the ground. The flight trajectory is estimated based on GPS time, with the coordinate system anchored at the drone's first sampled point.

Due to the scarcity of humans and man-made objects in natural habitats, we strategically positioned various objects of different sizes (from centimeters to meters) in diverse locations and orientations, in addition to capturing natural scenes. As shown in Table 1, this is the first aerial 3D dataset designed for object detection in WiSAR scenarios, where it is compared against other outdoor 3D datasets.

In our application, the scene is mostly static, with humans being the only expected source of motion. However, the sensor emits three rays per cycle, which can introduce motion artifacts, such as ghosting and smearing, when scanning moving people. For instance, a person in motion may appear duplicated, elongated, or distorted along their path of travel. To reduce these effects, we separate points from the three scan orientations into distinct strips.

3.2. The Dataset

Categories. We annotated 22 categories of humans and man-made objects found in wilderness settings, with the latter serving as markers of human activity, for a total of $67.5K$ cuboids encapsulating objects, as shown in Fig. 4. Regarding people, the primary category of interest, we annotated both standing and seated people, including people in seated or crouched positions. The other categories, which indicate the presence of people, include boxes, chairs, small bins, barrels, mattresses, vehicles, cones, small and big tables, tents, ramps, gazebos, pipes, balls, pipes on tripods that resemble telescopes, drone cases, play tunnels, flags, canopies, and portable potties. These objects vary significantly in size, ranging from smaller items like cones, balls, and pipes, with average dimensions of $[0.12, 0.13, 0.17]$, $[0.25, 0.25, 0.24]$, and $[1.06, 0.19, 0.21]$ meters, respectively, to larger items such as tents, vehicles, and canopies, with average dimensions of $[2.13, 2.18, 1.38]$, $[4.48, 1.91, 1.63]$, and $[3.32, 3.20, 2.07]$ meters.

Additionally, it is important to note that certain objects, such as standing and sitting persons, are deformable and can appear in various poses, adding complexity to the dataset. This is reflected in the mean and standard deviation of the bounding box dimensions and center point heights for these categories. Specifically, for the standing person class, the mean and standard deviation (in meters) for the bounding box dimensions are $[0.48 \pm 0.09, 0.58 \pm 0.10, 1.69 \pm 0.10]$,

Dataset	Venue	Classes†	Avg. points	Returns/shot	Frequencies	Environment	Detection	LiDAR	Aerial
Paris-Lille-3D [19]	IJRR'18	9 (50)	-	1	1	roads	✗	✓	✗
SemanticKITTI [2]	ICCV'19	25 (28)	121K	1	1	roads	✗	✓	✗
DublinCity [32]	BMVC'19	13	19M	-	1	urban	✗	✓	✓
DALES [25]	CVPRW'20	8 (9)	12M	4	1	urban/rural	✗	✓	✓
Campus3D [15]	ACM MM'20	24	-	-	-	urban	✗	✗	✓
SensatUrban [13]	IJCV'22	13 (31)	119M	-	-	urban	✗	✗	✓
KITTI [12]	CVPR'12	3 (8)	120K	1	1	roads	✓	✓	✗
Argoverse 1 [8]	CVPR'19	3 (15)	107K	1	1	roads	✓	✓	✗
NuScenes [7]	CVPR'20	10 (23)	34K	1	1	roads	✓	✓	✗
Waymo Open [23]	CVPR'20	4	177K	2	1	roads	✓	✓	✗
ONCE [17]	NeurIPS'21	3 (5)	65K	1	1	roads	✓	✓	✗
Argoverse 2 [27]	NeurIPS'21	26 (30)	97K	1	1	roads	✓	✓	✗
WiSAR3D (ours)	WACV'26	22	2.1M	7	4	wilderness	✓	✓	✓

Table 1. **Comparison to popular outdoor 3D datasets.** (top) 3D datasets are annotated for semantic segmentation tasks. (bottom) 3D detection datasets are captured from the perspective of vehicles. Our dataset is the first aerial dataset tailored for 3D detection in Wilderness Search and Rescue (WiSAR) scenarios. † The number of classes used for evaluation and the total annotated classes in brackets.

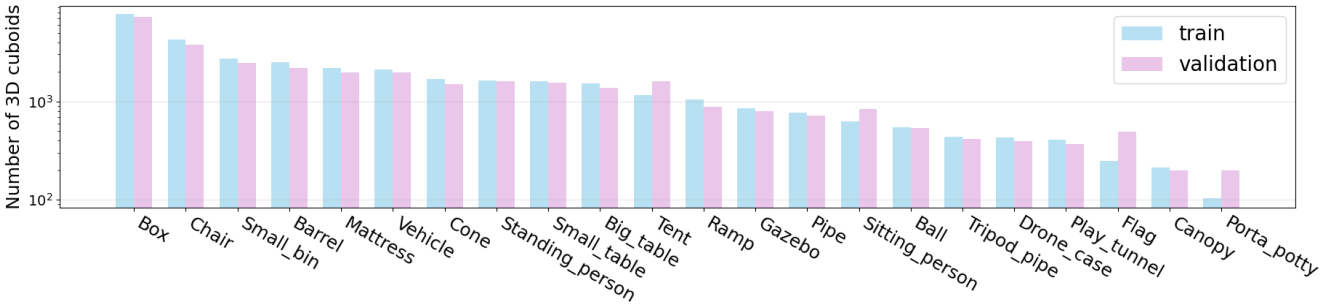


Figure 4. **Categories.** Our categories serve as potential indicators of human presence in wilderness environments. They vary greatly in size, shape, and deformation, with many being unique to our dataset. The figure also displays the number of annotated objects per category.

with a center point height of 0.67 ± 0.56. Similarly, for the sitting person class, the bounding box dimensions are $[0.72 \pm 0.14, 0.62 \pm 0.0, 1.20 \pm 0.13]$, and the center point height is 0.39 ± 0.56. These statistics show that distinguishing between a standing and sitting person based on height alone is unreliable.

Data format. Our *WiSAR3D* dataset includes 2633 strips, each between 1 to 10 seconds, for a total of 5.1 hours. Every flight path generates a point cloud strip saved as a *las* file. To optimize computational efficiency and simplify training, strips exceeding 10 seconds were divided into shorter segments, with non-meaningful points removed by truncating longer strips and excluding data collected during the drone's turnaround maneuvers. These strips are structured as $N \times 5$, with N denoting the number of points, and each point is characterized by five features: the coordinates (x, y, z), the reflection intensity, and the point's return number (ranging from 1 to 7). They are converted into *npy* files and stored separately, simplifying user reading access.

3.3. Labeling Process

To simplify the labeling process for each scan setup, we combine strips from different flight paths to form a dense point cloud scene. For this purpose, we use the original point coordinates, which are provided in the world coordinate system. Two experts, a junior and a senior, annotated the data using the CVAT annotation tool [1]. Following the common protocol [12, 17, 23], each object was annotated using a 7-degrees-of-freedom (7-DOF) bounding box, defined by the center coordinates (cx, cy, cz), length (l), width (w), height (h), and yaw angle (α).

Afterward, each strip was separated and transformed into its local coordinate system along with the corresponding labels. A subsequent validation phase was performed on the separated strips to remove bounding boxes that lacked objects, specifically those without sampled points within the given strip. Such empty boxes can occur due to moving objects, like vehicles or people, that were present in other strips but not captured in the current one.

4. Benchmark & Metrics

WiSAR3D includes 2633 contiguous point clouds collected along the flight path, referred to as *strips*, divided into 1372 for training and 1261 for validation. The partition ensures a diverse representation of heights, velocities, and PRRs in each set while keeping all strips of the same scene together in one set (either training or validation). See Table 2 for

	Heights [m]			Velocities [m/s]			PRRs [kHz]			
	30-60	65-95	100-160	3	4	5	300	600	1200	1800
Train	632	376	364	223	697	452	134	316	822	100
Validation	639	304	318	213	622	426	115	276	802	68

Table 2. **Data statistics.** Number of strips used for training and testing, categorized by flight altitudes, velocities, and pulse repetition rates (PRRs).

the strip statistics. For each strip, the 3D detection task involves predicting 3D boxes for objects from 22 pre-defined categories presented in Fig. 4.

We provide five evaluation metrics, as suggested in [7]: mAP, ATE, ASE, AOE, and NDS, with modifications as necessary to suit our data, as described below.

mAP. Average Precision (mAP) assesses the precision of object localization within scenes by computing the area under the precision-recall curve. It determines the match between predicted and ground truth boxes by evaluating the distance between their centers. The mAP is the mean AP across all categories and a list of predefined matching thresholds. In [12, 17, 23], true positives are determined using the intersection over union (IoU) metric. In nuScenes [7], where object classes may exhibit rotational symmetry, the detection is decoupled from object orientation and size. We adopt orientation decoupling due to rotationally symmetric classes in our dataset (e.g., *ball*, *cone*) and modify the nuScenes [7] size decoupling approach as follows: Instead of employing a single predefined matching set (i.e., $0.5, 1, 2, 4$ meters) for all categories, we propose a specific matching set for each category based on the class's average size. This adjustment accounts for the diverse range of sizes within our dataset, spanning from objects smaller than 20 centimeters to those over 3 meters. Thus, our approach ensures sufficient overlap between predicted and ground truth boxes.

Specifically, for a category $c \in \mathbb{C}$ with average 2D dimensions (dx, dy) along the X and Y axes, we define the matching threshold set for this category as:

$$\mathbb{D} = \{0.1, 0.5, 0.9\} \times min(dx, dy). \tag{1}$$

Consequently, the mean Average Precision (mAP) is the average AP over all categories and thresholds:

$$\text{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} \text{AP} \tag{2}$$

ATE, ASE, AOE. Average Translation Error (ATE) quantifies the Euclidean center distance, with a true positive identified based on a single predefined threshold ($0.5 \times min(dx, dy)$). Average Scale Error (ASE) measures the error in the intersection over union (IoU) between predicted and ground truth boxes. Average Orientation Error (AOE) represents the minimal yaw angle difference between the prediction and the ground truth.

For ATE we set the matching threshold to $0.5 \times min(dx, dy)$. The ASE and AOE are similar to nuScenes. We exclude AOE measurements for the categories *Ball*, *Barrel*, *Small bin*, *Cone*, and *Flag* due to their lack of clear orientation. Angles for all other categories are measured across a full $360°$ period, except for *Mattress*, *Gazebo*, *Canopy*, *Small table*, *Big table*, *Box*, *Drone case*, *Pipe*, and *Play tunnel*, where they are measured within a $180°$ period because these items exhibit bilateral symmetry.

NDS. The NuScenes Detection Score (NDS) is a weighted sum of all the aforementioned metrics. Let $\mathbb{TP} = \{\text{ATE}, \text{ASE}, \text{AOE}\}$ be the set of the True Positive metrics. The NDS is defined as:

$$\text{NDS} = \frac{1}{2}\text{mAP} + \frac{1}{6} \sum_{\text{mTP} \in \mathbb{TP}} (1 - \min(1, \text{mTP})). \tag{3}$$

5. Experiments

We adapted and tested state-of-the-art autonomous driving detectors on our benchmark, and analyzed the impact of different sampling rates.

5.1. Implementation Details

We adapted state-of-the-art and widely used 3D detectors: PointPillars [14], PointRCNN [21], PV-RCNN (Centerhead) [22], Voxel-RCNN (Centerhead) [10], CenterPoint [29], VoxelNeXt2d [9], VoxelNeXt [9], DSVT (Voxel) [26], and HEDNet [30], using implementations from [24]. All models were trained for 80 epochs on 8 A100-SXM4-80GB GPUs (total batch size: 16), with standard augmentations including rotation, flipping, scaling, and object copy/paste. These augmentations include random flipping around the X-axis, random scaling drawn from the range $[0.95, 1.05]$, and random rotation around the Z-axis with angles drawn from $[-45°, 45°]$. Additionally, we copy/paste GT objects from different training strips and inserted them into strips during training, as done in [28].

We used the version of [22] with the center head, and the version of [10] with the center head and dynamic voxels, as they have been shown to outperform their original counterparts [24]. For CenterPoint [29], we modified the 2D BEV backbone to have a single level instead of two, as this configuration demonstrated better results, as shown in the supplemental. For PointRCNN [21], a point-based detector, we increased the number of points to better handle our denser point clouds. The supplemental shows that increasing the number of points for the input sampling phase, as well as the number of points used in the backbone's hierarchies, improves results, albeit with longer training times and higher memory consumption. For example, with the original KITTI setting ($16K$ points), the model achieves an

Method	mAP↑	NDS↑	ATE↓	ASE↓	AOE↓	Training ↓ Time [h]	Inference ↑ FPS	Memory [GB] ↓ Training	Inference
PointPillars [14] [CVPR'19]	0.3592	0.5079	0.1547	0.1982	0.6774	1.92	9.43	38.39	8.64
PointRCNN [21] [CVPR'19]	0.4163	0.5276	0.2903	0.3355	0.4577	11.63	0.23	35.18	**4.04**
PV-RCNN [22] [CVPR'20]	0.5043	0.6254	0.1782	0.2345	0.3478	24.10	0.24	**17.45**	6.62
Voxel-RCNN [10] [AAAI'21]	0.5062	0.6382	0.1721	0.2297	0.2880	**1.63**	**10.22**	17.67	6.09
CenterPoint [29] [CVPR'21]	0.6076	0.7146	0.0853	0.1570	0.2928	1.97	7.43	20.17	5.16
VoxelNeXt2d [9] [CVPR'23]	0.5805	0.7127	0.1036	0.1476	**0.2141**	2.13	7.22	51.13	11.82
VoxelNeXt [9] [CVPR'23]	0.6394	0.7426	**0.0779**	0.1468	0.2378	3.37	4.55	47.61	16.13
DSVT (Voxel) [26] [CVPR'23]	0.6527	**0.7475**	0.0814	**0.1454**	0.2466	2.90	4.90	79.04	35.88
HEDNet [30] [NeurIPS'23]	**0.6654**	0.7421	0.0909	0.1540	0.2985	1.98	7.41	52.05	28.69

Table 3. **Detection results on WiSAR3D dataset.** We benchmarked state-of-the-art detectors from autonomous driving on WiSAR3D. The table reports five accuracy metrics, training time (on 8×A100 GPUs), inference FPS (on a single A100, batch size 1), and GPU memory usage during training and inference. [26, 30] yield the best accuracy but are resource-intensive and slow at inference. In contrast, [10] is faster and more efficient, but less accurate.

Method	Vehicle	Mattress	Ball	Gazebo	Canopy	Chair	Small table	Big table	Tripod pipe	Barrel	Tent	Box	Drone case	Small bin	Ramp	Sitting person	Standing person	Cone	Flag	Pipe	Porta potty	Play tunnel	mAP
PointPillars	.8223	.6384	.1784	.9367	.9234	.3443	.2861	.4310	.1098	.2960	.7683	.3726	.2838	.2615	.3676	.0126	.3219	.0052	.0097	.1414	.1313	.2607	.3592
PointRCNN	.8948	.6716	.0000	.9555	.9319	.4412	.5933	.4549	.3057	.4198	.7877	.4071	.4885	.1729	.5762	.0372	.2988	.0000	.0075	.0000	**.7086**	.0054	.4163
PV-RCNN	.9065	.6633	.2026	.9502	.9365	.5373	.6676	.5091	.6084	.5796	.8945	.4993	.5623	.4196	.6054	.1510	.4255	.0000	.3118	.0000	.3894	.2744	.5043
Voxel-RCNN	.9102	.5537	.3582	.9477	.9305	.5693	.6874	.5098	.6297	.5922	.8952	.5520	.5877	.4908	.6264	.1758	.4478	.0000	.3145	.0000	.2633	.0952	.5062
CenterPoint	.9204	.7799	.4597	.9435	.9302	.6356	.7172	.6070	.6710	.6287	.9025	.6292	.6647	.5436	.7046	.3888	.5026	.0271	.3901	.3068	.4373	.5761	.6076
VoxelNeXt2d	.8762	.7101	.5401	.9425	.9262	.5640	.6441	.5445	.6294	.6019	.8294	.5976	.6454	.5426	.7119	.2601	.4592	.0761	.2952	.3709	.4011	.6025	.5805
VoxelNeXt	.9261	**.8215**	.5508	.9601	.9370	**.6655**	**.7739**	.6151	**.6989**	**.6764**	.9230	**.6720**	**.7366**	.5772	.7121	.3929	.5174	.0559	**.4275**	.2954	.5279	.6031	.6394
DSVT (Voxel)	.9223	.7594	**.5659**	.9431	**.9406**	.6566	.7270	**.6328**	.6515	.6444	.9195	.6623	.7231	**.6017**	**.7442**	**.4501**	**.5480**	.2073	.3382	.4585	.6295	**.6343**	.6527
HEDNet	**.9264**	.7805	.5536	**.9626**	.9335	.6233	.7665	.6101	.6792	.6646	**.9238**	.6542	.6693	.5851	.7075	.3415	.4842	**.3992**	.3853	**.8695**	.4933	.6246	**.6654**

Table 4. **Detection results by class.** The challenging classes across all methods include small objects (e.g., cones, pipes, balls, and small bins) and deformable ones (e.g., persons and flags).

mAP of 0.1117 while training takes 1.38 hours on 8 GPUs, each using 8.72 GB. Increasing the input to $300K$ points and adjusting the hierarchy substantially improves mAP to 0.4163, but training time rises to 11.63 hours and memory usage to 35.18 GB per GPU.

Additionally, we utilized four features for the input strips, the coordinates (x, y, z) and the reflectance intensity. For [9, 10, 21, 22, 29, 30] we set the voxel size to $0.1m, 0.1m, 0.15m$ and the grid range to $[-40m, 110m], [-75m, 75m], [-2m, 4m]$ along the X, Y, and Z axes respectively. This results in a 1500×1500 pixel Birds Eye View (BEV) pseudo-image for the voxel-based detectors. For [14] and [26] we set the voxel size to $0.32m, 0.32m, 6.0m$ and $0.32m, 0.32m, 0.1875m$ respectively, both with the respective grid range of $[-40.88m, 108.88m], [-74.88m, 74.88m], [-2m, 4m]$ along the X, Y, and Z axes, which gives a BEV pseudo-image of 468×468 pixels. To accommodate our denser point clouds, we raised the maximum voxel count limitation from $150k$, as employed in Waymo, to $900k$.

5.2. Detection Benchmark

Table 3 shows that voxel-based detectors outperform point-based ones (e.g., PointRCNN), benefiting from the high point density in our scenes. DSVT (Voxel) and HEDNet achieve the best performance, but at the cost of increased memory consumption, as they leverage a larger effective receptive field (ERF) to capture voxel relationships within the object and its immediate surroundings. Given the dataset's millions of points per strip, a larger ERF increases computational demands. These results underline the need for models better suited to WiSAR3D.

Table 4 presents the detection results across all categories. Generally, performance varies across categories. For example, DSVT performs better in person detection (mAP of 45.01% for sitting and 54.80% for standing persons), while HEDNet performs better at detecting smaller objects like cones and pipes. It is observed that larger objects such as *Gazebo*, *Tent*, *Canopy*, and *Vehicle* are detected effectively, whereas smaller objects like *Cone*, *Pipe*, *Small bin*, and *Ball*, are more challenging to detect. This is because smaller objects have fewer sampled points and occupy only a small portion of the detector's receptive field. Furthermore, categories such as *Sitting person*, *Standing person*, and *Flag* are particularly challenging due to their deformable characteristics. Notably, these classes are crucial, as missing persons often lack large objects (e.g., vehicles, gazebos) but may have smaller ones (e.g., boxes) that

serve as indicators of their presence in the wild. These results underscores the need for annotated datasets containing such objects to support search and rescue efforts better.

5.3. Sampling Rate Analysis

In this experiment, we examine how different sampling rates affect the *WiSAR* detection task. Higher rates produce denser point clouds, increasing storage needs and processing time, raising the question of whether such rates are necessary. Additionally, varying sampling rates across sensors may hinder generalization across devices. This experiment reveals two key insights: (1) Scanning at the highest rate is unnecessary; models trained on lower rates generalize well to higher ones. (2) Conversely, models trained on higher rates generalize poorly to lower ones, though incorporating lower-rate data during training improves performance.

For all the experiments we divided the data into four sampling rates: 300kHz, 600kHz, 1200kHz, and 1800kHz. We used the CenterPoint detector as it provides a good balance between speed, memory, and accuracy. We evaluated on 16 categories, which are present in all the sampling rates.

First, we examine the effectiveness of training on a specific sampling rate and evaluate others. We designated the 1200kHz rate as our default Pulse Repetition Rate (PRR) and utilized it as a reference point. Its intermediate position between the highest and lowest sampling rates, along with its majority of example strips as detailed in Table 2, makes it the optimal choice. We utilized the 1200kHz rate training set, while evaluation was conducted on the 1200kHz validation set and on both the training and validation sets of alternate rates. Rows highlighted in soft blue in Table 5 indicate that lower sampling rates present greater challenges for accurate object detection. This is expected, as lower rates produce fewer sampling points, resulting in reduced object recognition performance. Interestingly, when evaluated at a higher rate such as 1800kHz, the mAP exceeds that of the training rate (1200kHz), suggesting that training with the highest available sampling rate may not be necessary. These results indicate that models trained on lower sampling rates can generalize to higher rates.

Next, we examine how using various sampling rates during training affects performance when evaluated on other rates. We designate the 1200kHz rate as the default PRR and augment the training data with additional rates, while keeping the test set fixed. Each block in Table 5 shows that enriching the training set with samples from multiple PRRs improves detection accuracy compared to training solely on the 1200kHz subset. Moreover, the greater the diversity of PRRs in the training data, the more substantial the improvement. For instance, when evaluated on the 1800kHz dataset, adding either the 300kHz or 600kHz datasets increases the baseline mAP by 1.5%. Incorporating both yields a complementary effect, boosting the mAP by 3.35%.

Train	Test	mAP↑	NDS↑	ATE↓	ASE↓	AOE↓
1200	1200	0.6881	0.7655	0.0573	0.1388	0.2752
1200	300	0.4108	0.5658	0.1393	0.2218	0.4765
1200 & 1800	300	0.4377	0.6189	0.0743	0.1743	0.3514
1200 & 600	300	0.5093	0.6216	0.1857	0.2445	0.3683
1200 & 1800 & 600	300	0.5257	0.6753	0.0739	0.1571	0.2944
1200	600	0.4973	0.6065	0.0799	0.1863	0.5867
1200 & 1800	600	0.5060	0.6241	0.0795	0.1892	0.5045
1200 & 300	600	0.5272	0.6398	0.0812	0.1891	0.4730
1200 & 1800 & 300	600	0.5411	0.6522	0.0775	0.1739	0.4584
1200	1800	0.7295	0.7579	0.1052	0.2165	0.3197
1200 & 600	1800	0.7440	0.7745	0.1042	0.2060	0.2746
1200 & 300	1800	0.7443	0.8011	0.0474	0.1592	0.2196
1200 & 600 & 300	1800	0.7630	0.8183	0.0485	0.1513	0.1791

Table 5. **Impact of various rates used for training.** Integrating diverse sampling rates proves advantageous for precise detection. In particular, adding low sampling rates (e.g. 600kHz in the second block) to high rates (e.g., 1200) substantially improves the results when tested on the 300.

Finally, the benefits of incorporating additional data vary across PRRs. For example, on the challenging 300kHz subset, adding samples from the 1800kHz subset improves the baseline's mAP by 2.69%, while including the 600kHz subset boosts it by 9.85%. This emphasizes the importance of low rates, particularly for detecting objects in scenes sampled at low rates.

Limitations and future work. Our dataset includes strips with millions of points (see supplemental for details), posing challenges for some detectors due to high GPU memory demands and slow inference. Future work will focus on improving detection of difficult classes like persons while optimizing memory efficiency to better support WiSAR on standard hardware.

6. Conclusion

This paper presents a new dataset, *WiSAR3D*, along with benchmarks, metrics, baselines, and detection task results. *WiSAR3D* is the first aerial LiDAR 3D dataset specifically designed for object detection in Wilderness Search and Rescue scenarios. The dataset comprises continuous point clouds, referred to as strips, with multiple returns, demonstrating the ability to penetrate foliage and detect obscured objects. Additionally, the data was collected at various altitudes, velocities, and sampling rates, providing realistic scenarios. The different sampling rates allow for an analysis of their effect on the detection task. We provide a comprehensive evaluation of state-of-the-art detectors as a benchmark, highlighting the need for further research, particularly for detecting challenging objects such as humans and small objects. In the future, we plan to expand the dataset to include more data, possibly from other landscapes.

Acknowledgments. This work was supported by the Israel Science Foundation (ISF) under grant 2329/22.

References

[1] Computer vision annotation tool (cvat). https://github.com/cvat-ai/cvat. 5

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 5

[3] Arturo Miguel Russell Bernal, Walter Scheirer, and Jane Cleland-Huang. Nomad: A natural, occluded, multi-scale aerial dataset, for emergency response scenarios. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8584–8595, 2024. 1, 2, 3

[4] Elizabeth Bondi, Raghav Jain, Palash Aggrawal, Saket Anand, Robert Hannaford, Ashish Kapoor, Jim Piavis, Shital Shah, Lucas Joppa, Bistra Dilkina, et al. Birdsai: A dataset for detection and tracking in aerial thermal infrared videos. In *Proceedings of the IEEE/CVF Winter conference on applications of computer vision*, pages 1747–1756, 2020.

[5] Dunja Božić-Štulić, Željko Marušić, and Sven Gotovac. Deep learning approach in aerial imagery for supporting land search and rescue missions. *International Journal of Computer Vision*, 127(9):1256–1278, 2019. 2

[6] Daniel Broyles, Christopher R Hayner, and Karen Leung. Wisard: A labeled visual and thermal image dataset for wilderness search and rescue. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9467–9474. IEEE, 2022. 1, 2, 3

[7] 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 multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1, 2, 5, 6

[8] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019. 1, 2, 5

[9] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023. 6, 7

[10] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv preprint arXiv:2012.15712*, 1(2):4, 2020. 6, 7

[11] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018. 1

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 2, 5, 6

[13] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. *International Journal of Computer Vision*, 130(2):316–343, 2022. 1, 2, 5

[14] 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. 6, 7

[15] Xinke Li, Chongshou Li, Zekun Tong, Andrew Lim, Junsong Yuan, Yuwei Wu, Jing Tang, and Raymond Huang. Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 238–246, 2020. 1, 2, 5

[16] Thomas Manzini and Robin Murphy. Open problems in computer vision for wilderness sar and the search for patricia wu-murad. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3784–3789, 2023. 2, 3

[17] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021. 1, 2, 5, 6

[18] Jesús Morales, Ricardo Vázquez-Martín, Anthony Mandow, David Morilla-Cabello, and Alfonso García-Cerezo. The uma-sar dataset: Multimodal data collection from a ground vehicle during outdoor disaster response training exercises. *The International Journal of Robotics Research*, 40(6-7): 835–847, 2021. 1

[19] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, 2018. 5

[20] Sasa Sambolek and Marina Ivasic-Kos. Automatic person detection in search and rescue operations using deep cnn detectors. *Ieee Access*, 9:37905–37922, 2021. 1, 2, 3

[21] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 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. 6, 7

[22] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 6, 7

[23] 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. 1, 2, 5, 6

[24] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6

[25] Nina Varney, Vijayan K Asari, and Quinn Graehling. Dales: A large-scale aerial lidar data set for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 186–187, 2020. 1, 2, 5

[26] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13520–13529, 2023. 6, 7

[27] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 1, 2, 5

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

[29] 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. 6, 7

[30] 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, 2024. 6, 7

[31] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021. 1

[32] SM Zolanvari, Susana Ruano, Aakanksha Rana, Alan Cummins, Rogerio Eduardo Da Silva, Morteza Rahbar, and Aljosa Smolic. Dublincity: Annotated lidar point cloud and its applications. *arXiv preprint arXiv:1909.03613*, 2019. 1, 2, 5