Multi-Space Alignments Towards Universal LiDAR Segmentation

Supplementary Material

Table of Contents

A. Multi-Dataset Configuration

In this section, we elaborate on the details of combining multiple heterogeneous LiDAR segmentation datasets to train a universal LiDAR segmentation model.

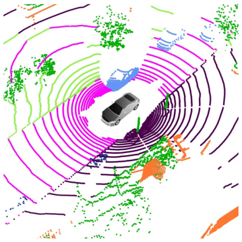
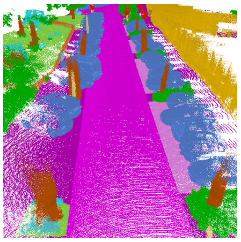
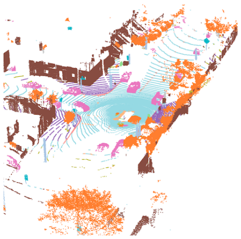
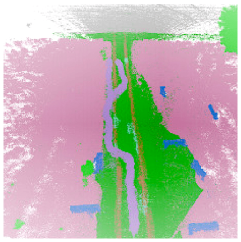
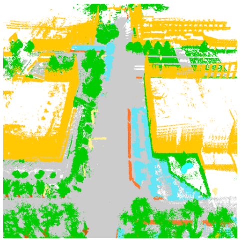
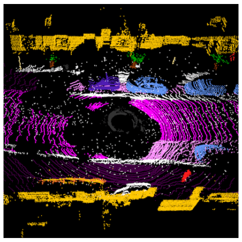
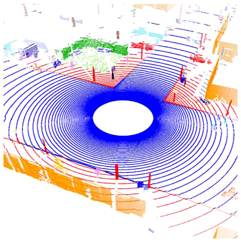
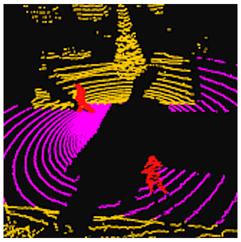
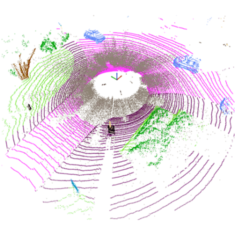
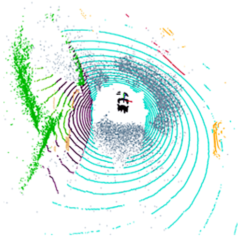
A.1. Overview

In this work, we resort to ten driving datasets for achieving *i)* multi-dataset training and evaluations, *ii)* knowledge transfer and generalization, and *iii)* out-of-distribution generalization. A summary of the datasets used in this work is shown in Tab. A. For multi-dataset training and evaluations, we use the LiDAR and camera data from the *nuScenes* [4, 9], *SemanticKITTI* [1], and *Waymo Open* [28] datasets.

- **nuScenes** is a large-scale public dataset for autonomous driving, created by Motional (formerly nuTonomy). It is widely used in the research and development of autonomous vehicles and related technologies. The dataset includes a comprehensive range of sensor data crucial for autonomous driving. It typically contains data from multiple cameras, LiDAR, RADAR, GPS, IMU, and other sensors. This multimodal data collection is essential for developing and testing algorithms for perception, prediction, and motion planning in autonomous vehicles. One of the strengths of the nuScenes dataset is its diversity. The data encompasses various driving conditions, including different times of day, weather conditions, and urban environments. This diversity is crucial for training robust algorithms that can handle real-world driving scenarios. In this work, we use the LiDAR semantic and panoptic segmentation data from the *lidarseg*[1] subset in the nuScenes dataset, which includes segmentation labels for the entire nuScenes dataset, encompassing thousands of scenes, each a 20-second clip captured from a driving vehicle in various urban settings. 32 classes are manually labeled, covering a wide range of objects and elements in urban scenes, where 16 of them are typically adopted in evaluating the segmentation performance. More details of this dataset can be found at `https://www.nuscenes.org/nuscenes`.
- **SemanticKITTI** is a well-known dataset in the field of autonomous driving and robotics, specifically tailored for the task of semantic and panoptic segmentation using LiDAR point clouds. It is an extension of the original KITTI Vision Benchmark Suite[2] [10], with annotations for over 20 sequences of driving scenarios, each containing tens of thousands of LiDAR scans. The dataset covers a variety of urban and rural scenes. This includes city streets, residential areas, highways, and country roads, providing a diverse set of environments for testing algorithms. The dataset provides labels for 28 different semantic classes, including cars, pedestrians, bicycles,

[1] `https://www.nuscenes.org/lidar-segmentation`.
[2] `https://www.cvlibs.net/datasets/kitti`.

Table A. **Summary of the datasets** used in this work. We split different datasets into three categories: **i)** The *nuScenes* [9], *SemanticKITTI* [1], and *Waymo Open* [28] datasets are used for multi-dataset training and evaluations. **ii)** The *RELLIS-3D* [14], *SemanticPOSS* [25], SemanticSTF [32], SynLiDAR [31], and DAPS-3D [15] datasets are used for knowledge transfer and generalization (*w/* fine-tuning). **iii)** The SemanticKITTI-C [17] and nuScenes-C [17] datasets are used for out-of-distribution generalization (*w/o* fine-tuning).

Dataset Summary				
				
nuScenes [9] [Link]	SemanticKITTI [1] [Link]	Waymo Open [28] [Link]	RELLIS-3D [14] [Link]	SemanticPOSS [25] [Link]
				
SemanticSTF [32] [Link]	SynLiDAR [31] [Link]	DAPS-3D [15] [Link]	SemanticKITTI-C [17] [Link]	nuScenes-C [17] [Link]

various types of vegetation, buildings, roads, and so on. 19 classes are typically adopted for evaluation. In total, around 4549 million points are annotated, and such extensive labeling provides a dense coverage for each LiDAR scan. More details of this dataset can be found at http://semantic-kitti.org.

- **Waymo Open** is a large dataset for autonomous driving, provided by Waymo LLC, a company that specializes in the development of self-driving technology. This dataset is particularly notable for its comprehensive coverage of various scenarios encountered in autonomous driving. The data is collected using Waymo's self-driving vehicles, which are equipped with an array of sensors, including high-resolution LiDARs, cameras, and radars. This multimodal data collection allows for comprehensive perception modeling. The dataset includes a wide range of driving environments and conditions, such as city streets, highways, and suburban areas, captured at different times of day and in various weather conditions. This variety is crucial for developing robust autonomous driving systems. In this work, we use its 3D Semantic Segmentation subset, which specifically provides point-level annotations for 3D point clouds generated by LiDAR sensors. 22 semantic classes are used during evaluation, encompassing a wide range of object classes, such as vehicles, pedestrians, and cyclists, as well as static objects like road signs, buildings, and vegetation. More details of this dataset can be found at https://waymo.com/open.

To validate that the learned features from our multi-dataset training setup are superior to that of the singe-dataset training in knowledge transfer and generalization, we conduct fine-tuning experiments on the following five datasets: *RELLIS-3D* [14], *SemanticPOSS* [25], *SemanticSTF* [32], *SynLiDAR* [31], and *DAPS-3D* [15].

- **RELLIS-3D** is a dataset focusing on off-road environments for autonomous navigation and perception, developed by Texas A&M University. It contains multimodal sensor data, including high-resolution LiDAR, RGB imagery, and GPS/IMU data, providing a comprehensive set for developing and evaluating algorithms for off-road autonomous driving. The dataset features diverse terrain types, such as grasslands, forests, and trails, offering unique challenges compared to urban scenarios. RELLIS-3D includes annotations for 13 semantic classes, including natural elements and man-made objects, crucial for navigation in off-road settings. More details of this dataset can be found at http://www.unmannedlab.org/research/RELLIS-3D.
- **SemanticPOSS** focuses on panoramic LiDAR scans, which include urban scenes, highways, and rural areas.

The dataset contains annotations for 14 semantic classes, covering vehicles, pedestrians, cyclists, and various road elements. Its panoramic view provides a 360-degree understanding of the vehicle's surroundings, which is beneficial for comprehensive scene analysis. More details of this dataset can be found at https://www.poss.pku.edu.cn/semanticposs.

- **SemanticSTF** studies the 3D semantic segmentation of LiDAR point clouds under adverse weather conditions, including snow, rain, and fog. It is built from the real-world STF [3] dataset with point-wise annotations of 21 semantic categories. The original LiDAR data in STF was captured by a Velodyne HDL64 S3D LiDAR sensor. In total, SemanticSTF selected 2076 scans for dense annotations, including 694 snowy, 637 dense-foggy, 631 light-foggy, and 114 rainy scans. More details of this dataset can be found at https://github.com/xiaoaoran/SemanticSTF.
- **SynLiDAR** is a synthetic dataset for LiDAR-based semantic segmentation. It is generated using advanced simulation techniques to create realistic urban, suburban, and rural environments. SynLiDAR offers an extensive range of annotations for a variety of classes, including dynamic objects like vehicles and pedestrians, as well as static objects like buildings and trees. This dataset is useful for algorithm development and testing in simulated environments where real-world data collection is challenging. More details of this dataset can be found at https://github.com/xiaoaoran/SynLiDAR.
- **DAPS-3D** is a dataset focusing on dynamic and static point cloud segmentation. It includes LiDAR scans from diverse urban environments, providing detailed annotations for dynamic objects such as vehicles, pedestrians, and cyclists, as well as static objects like buildings, roads, and vegetation. DAPS-3D is designed to advance research in dynamic scene understanding and prediction in autonomous driving, addressing the challenges posed by moving objects in complex urban settings. More details of this dataset can be found at https://github.com/subake/DAPS3D.

Meanwhile, we leverage the *SemanticKITTI-C* and *nuScenes-C* datasets in the Robo3D benchmark [17] to probe the out-of-training-distribution robustness of

- **SemanticKITTI-C** is built upon the validation set of the *SemanticKITTI* [1] dataset. It is designed to cover out-of-distribution corruptions that tend to occur in the real world. A total of eight corruption types are benchmarked, including fog, wet ground, snow, motion blur, beam missing, crosstalk, incomplete echo, and cross-sensor cases. For each corruption, three subsets that cover different levels of corruption severity are created, *i.e.* easy, moderate, and hard. The LiDAR segmentation models are expected to be trained on the clean sets while tested on these eight corruption sets. The performance degradation under corruption scenarios is used to measure the model's robustness. Two metrics are designed for such measurements, namely mean Corruption Error (mCE) and mean Resilience Rate (mRR). mCE calculates the relative robustness of a candidate model compared to the baseline model, while mRR computes the absolute performance degradation of a candidate model when it is tested on clean and corruption sets, respectively. In total, there are 97704 LiDAR scans in *SemanticKITTI-C*, which follow the original dense annotations in *SemanticKITTI*. More details of this dataset can be found at https://github.com/ldkong1205/Robo3D.
- **nuScenes-C** shares the same corruption and severity level definitions with *SemanticKITTI-C* and is built upon the validation set of the *nuScenes* [9] dataset. In total, there are 144456 LiDAR scans in *nuScenes-C*, which follow the original dense annotations in *nuScenes*. More details of this dataset can be found at https://github.com/ldkong1205/Robo3D.

A.2. Statistical Analysis

In this section, we conduct a comprehensive statistical analysis of the *nuScenes* [4, 9], *SemanticKITTI* [1], and *Waymo Open* [28] datasets to showcase the difficulties in merging heterogeneous LiDAR and camera data.

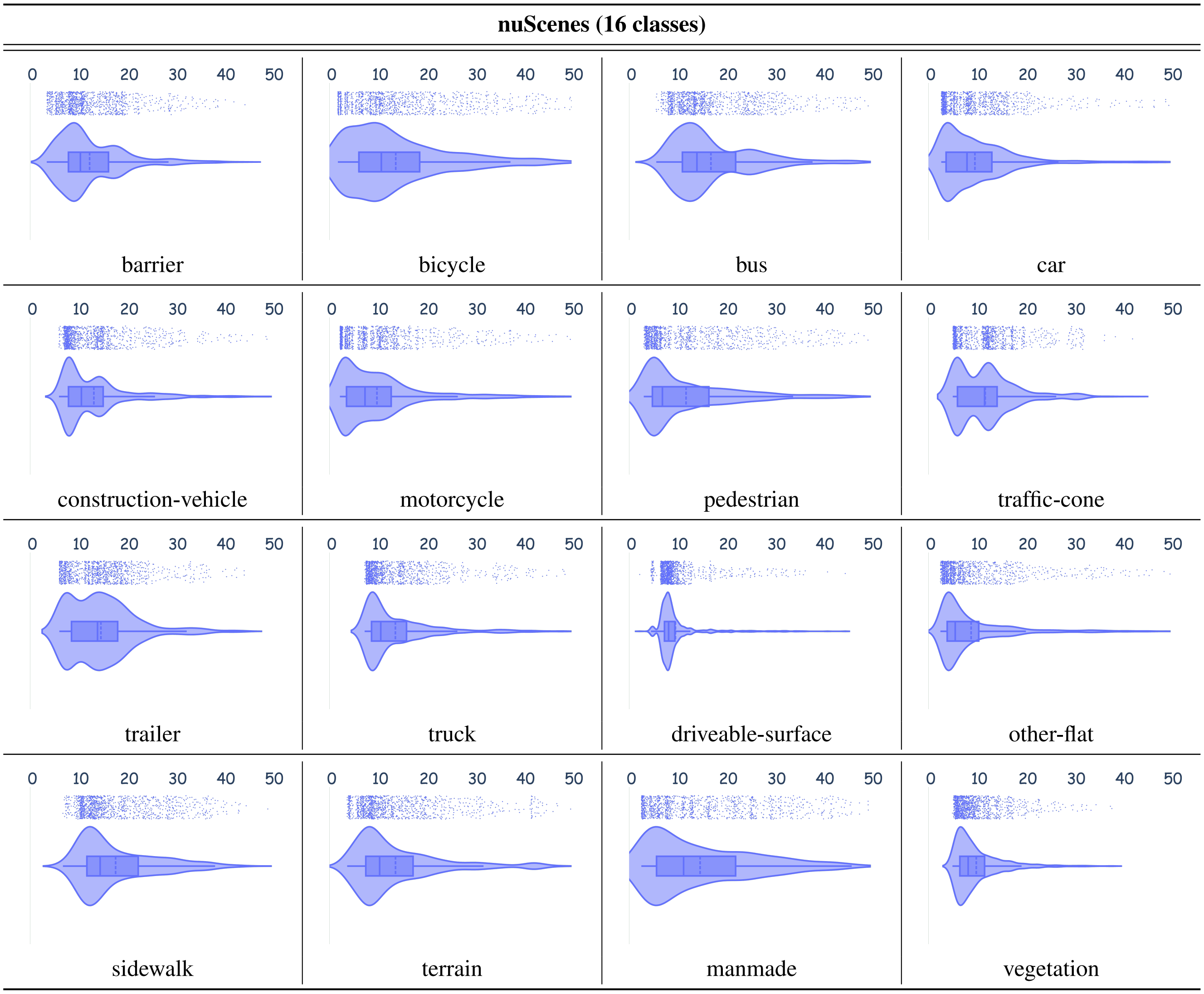
A.2.1 nuScenes

The LiDAR point clouds in the *nuScenes* [4, 9] dataset are acquired by a Velodyne HDL32E with 32 beams, 1080 (+/−10) points per ring, 20Hz capture frequency, 360-degree Horizontal FOV, +10-degree to −30-degree Vertical FOV, uniform azimuth angles, a 80m to 100m range, and up to around 1.39 million points per second. There are a total of 16 semantic classes in this dataset. The distributions of these classes across a 50 meters range are shown in Tab. B. As can be seen, most semantic classes distribute within the 20 meters range. The dynamic classes, such as `bicycle`, `motorcycle`, `bus`, `car`, and `pedestrian`, show a high possibility of occurrence at round 5 meters to 10 meters. The static classes, on the contrary, are often distributed across a wider range. Typically examples include `terrain` and `manmade`. Different semantic classes exhibit unique distribution patterns around the ego-vehicle.

A.2.2 SemanticKITTI

The LiDAR point clouds in the *SemanticKITTI* [1] dataset are acquired by a Velodyne HDL-64E with 64 beams, providing high-resolution data. The Velodyne HDL-64E features a 360-degree Horizontal Field of View (FOV), a Vertical FOV ranging from +2 to −24.33 degrees, and an

Table B. **The statistical analysis** of the 16 semantic classes in the ***nuScenes*** [9] dataset. Statistics are calculated from the *training* split of the dataset. Each violin plot shows the LiDAR point cloud density distribution in a 50 meters range. Best viewed in colors.

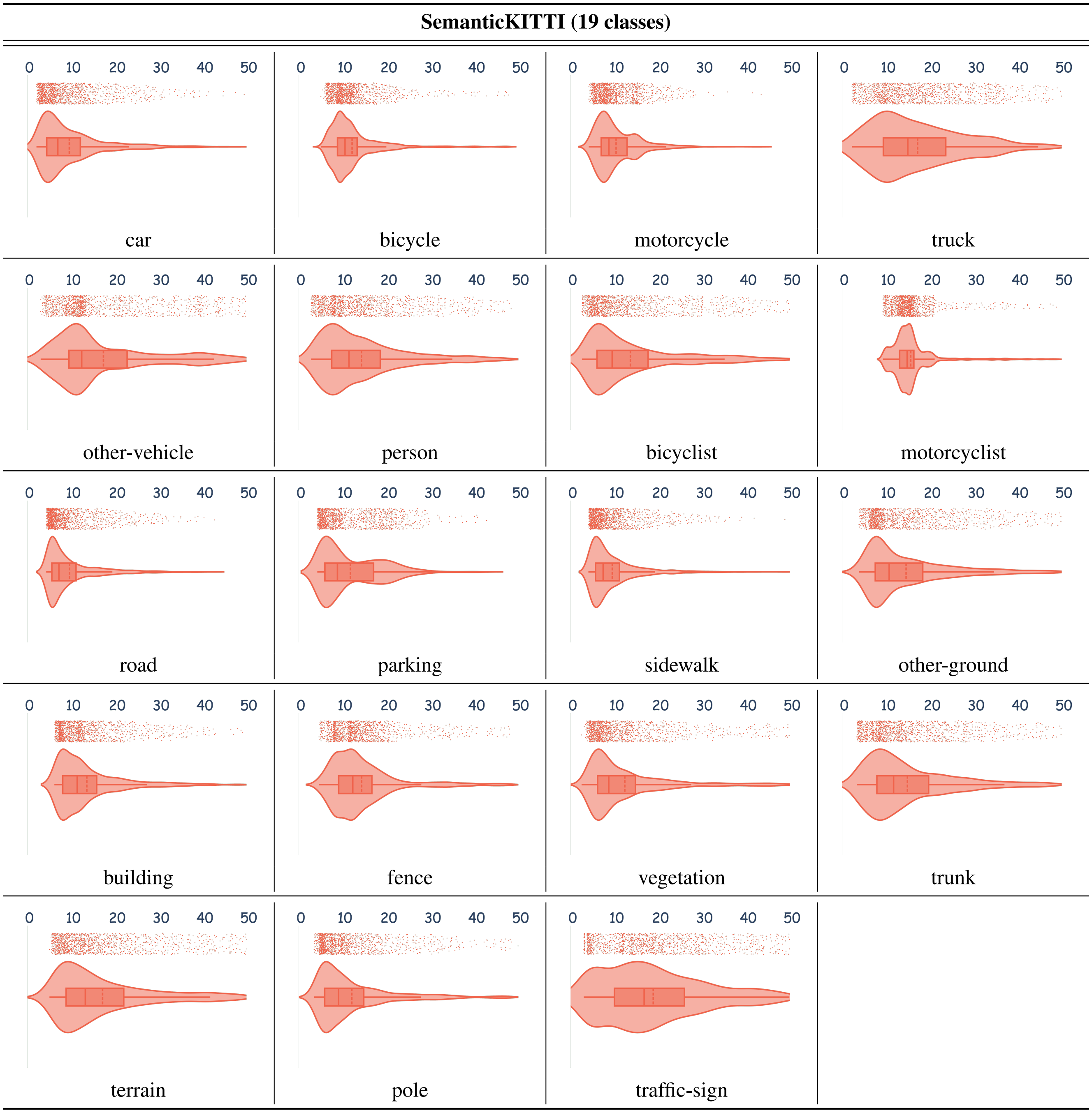


angular resolution of approximately $0.08 - 0.4$ degrees (vertically) and $0.08 - 0.35$ degrees (horizontally). The sensor operates at a 10Hz capture frequency and can detect objects within a range of up to 120m, delivering densely sampled, detailed point clouds with approximately 1.3 million points per second. There are a total of 19 semantic classes in this dataset. The distributions of these classes across a 50 meters range are shown in Tab. C. It can be seen from these statistical plots that the distributions are distinctly different from each other; points belonging to the `road` class are intensively distributed in between 5 meters to 10 meters around the ego-vehicle, while those dynamic classes like `bicyclist`, `motorcyclist`, `other-vehicle` and `truck`, tend to appear in a wider range. Similar to the *nuScenes* dataset, the 19 classes in *SemanticKITTI* also exhibit distinct patterns of occurrence in the driving scenes.

A.2.3 Waymo Open

The 3D semantic segmentation subset of the *Waymo Open* [28] dataset features LiDAR point clouds obtained using Waymo's proprietary LiDAR sensors, which include mid-range and short-range LiDARs. There are five LiDARs in total - one mid-range LiDAR (top) and four short-range LiDARs (front, side left, side right, and rear), where the mid-range LiDAR has a non-uniform inclination beam angle pattern. The range of the mid-range LiDAR is truncated to a maximum of 75 meters. The range of the short-range LiDARs is truncated to a maximum of 20 meters. The strongest two intensity returns are provided for

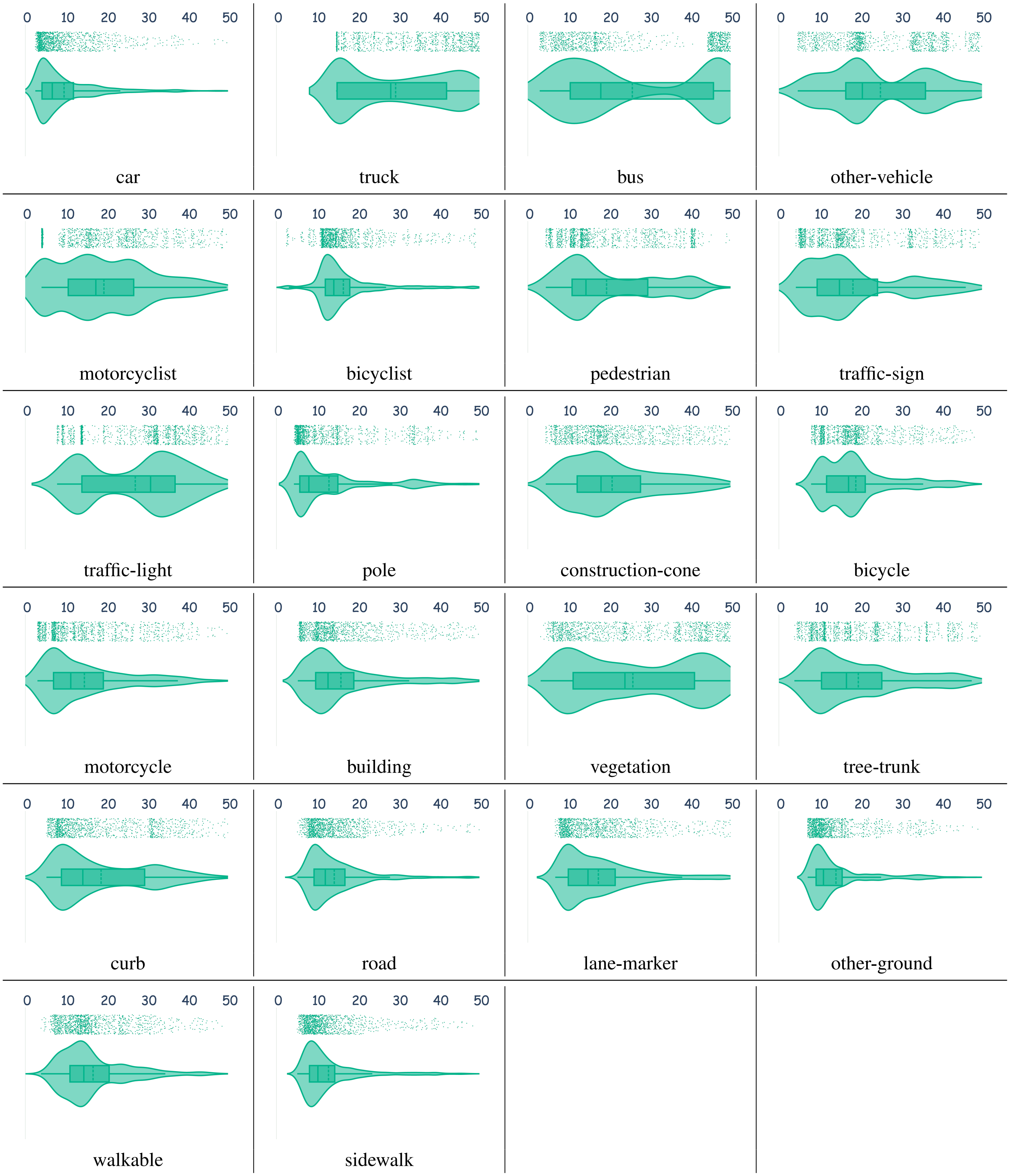
Table C. **The statistical analysis** of the 19 semantic classes in the ***SemanticKITTI*** [1] dataset. Statistics are calculated from the *training* split of the dataset. Each violin plot shows the LiDAR point cloud density distribution in a 50 meters range. Best viewed in colors.



all five LiDARs. An extrinsic calibration matrix transforms the LiDAR frame to the vehicle frame. The point clouds of each LiDAR in *Waymo Open* are encoded as a range image. Two range images are provided for each LiDAR, one for each of the two strongest returns. There are four channels in the range image, including range, intensity, elongation, and occupancy. The distributions of these classes across a 50 meters range are shown in Tab. D. As can be seen. the class distributions of *Waymo Open* are more diverse than those from *nuScenes* and *SemanticKITTI*. Some semantic classes, including `motorcyclist`, `pedestrian`, `construction-cone`, `vegetation`, and `tree-trunk`, are distributed across almost the entire driving scenes captured by the five LiDAR sensors.

Table D. **The statistical analysis** of the 22 semantic classes in the ***Waymo Open*** [28] dataset. Statistics are calculated from the *training* split of the dataset. Each violin plot shows the LiDAR point cloud density distribution in a 50 meters range. Best viewed in colors.

Waymo Open (22 classes)



B. Multi-Task Configuration

In this section, we supplement more details of our design and implementation toward multi-task (semantic and panoptic) LiDAR segmentation.

B.1. Overview

A proper pipeline design could enable the model to generate suitable predictions to fulfill multiple tasks simultaneously. In the context of LiDAR segmentation, we are especially interested in unifying semantic and panoptic segmentation of LiDAR point clouds. Such a holistic way of 3D scene understanding is crucial for the safe perception in autonomous vehicles.

B.2. Mean Shift

In this work, we enhance the versatility of our framework in an end-to-end fashion through the integration of a multi-tasking approach. This adaptation involves the modification of the instance extractor on top of the semantic predictions, which enables a dual output for both LiDAR semantic and panoptic segmentation. Specifically, drawing inspiration from DS-Net [11, 12], our instance extractor comprises an instance head, succeeded by a point clustering step. The instance head encompasses a sequence of multilayer perceptrons designed to predict the offsets between instance centers. This point clustering step strategically employs semantic predictions to filter out *stuff* points, thereby retaining only those associated with *thing* instances, such as `pedestrian`, `car`, and `bicyclist`. Subsequently, the remaining points undergo mean-shift clustering [6], utilizing features from the instance head to discern distinct instances. This meticulous process enhances the framework's capacity for accurate instance segmentation. The bandwidth for mean-shift in the *SemanticKITTI* and *Panoptic-nuScenes* datasets is set to 1.2 and 2.5, respectively.

C. Additional Implementation Details

In this section, we provide additional details to assist the implementation and reproduction of the approach proposed in the main body of this paper.

C.1. Datasets

In our multi-dataset training pipeline, we train our M3Net framework on the three most popular large-scale driving datasets, *i.e.*, the *SemanticKITTI* [1], *nuScenes* [9], and *Waymo Open* [28] datasets. These three datasets consist of 19130, 29130, and 23691 training LiDAR scans, and 4071, 6019, and 5976 validation LiDAR scans, respectively. Besides, we leverage the synchronized camera images from the corresponding datasets as our 2D inputs in the M3Net training pipeline for cross-modality alignments. The *SemanticKITTI*, *nuScenes*, and *Waymo Open* datasets contain 19130, 174780, and 118455 camera images in the train set, respectively, where *SemanticKITTI* has single-camera (front-view) data, *nuScenes* is with a six-camera (three front-view and three back-view) systems, and *Waymo Open* has five camera views in total.

For multi-task experiments on *SemanticKITTI* [1] and *Panoptic-nuScenes* [9], we follow the official data preparation procedures to set up the training and evaluations. Specifically, these two datasets share the same amount of data with their semantic segmentation subsets, *i.e.*, 19130 and 29130 training LiDAR scans, and 4071 and 6019 validation LiDAR scans, respectively. Each LiDAR scan is associated with a panoptic segmentation map which indicates the instance IDs. For additional details, kindly refer to the original papers.

For the knowledge transfer fine-tuning experiments on the *RELLIS-3D* [14], *SemanticPOSS* [25], *SemanticSTF* [32], *SynLiDAR* [31] and *DAPS-3D* [15] datasets, we follow the same procedure as Seal [21] to prepare the training and validation sets. Kindly refer to the original paper for more details on this aspect.

For the out-of-training-distribution generalization experiments on *SemanticKITTI-C* and *nuScenes-C*, we follow the same data preparation procedure in Robo3D [17]. There are eight different corruption types in each dataset, including fog, wet ground, snow, motion blur, beam missing, crosstalk, incomplete echo, and cross-sensor cases, where each corruption type contains corrupted data from three severity levels. In total, there are 97704 LiDAR scans in *SemanticKITTI-C* and 144456 LiDAR scans in *nuScenes-C*. For additional details, kindly refer to the original paper.

C.2. Text Prompts

In this work, we adopt the standard templates along with specified class text prompts to generate the CLIP text embedding for the three datasets used in our multi-dataset training pipeline. Specifically, the text prompts associated with the semantic classes in the *nuScenes* [9], *SemanticKITTI* [1], and *Waymo Open* [28] datasets are shown in Tab. E, Tab. F, and Tab. G, respectively.

C.3. Backbones

In this work, we adopt two models to serve as the backbone of our proposed M3Net, *i.e.*, the classical MinkUNet [5] and the more recent PTv2+ [29].

C.3.1 MinkUNet

The primary contribution of MinkUNet [5] is the introduction of a neural network architecture capable of processing 4D spatiotemporal data (3D space + time). This is particularly relevant for applications that involve dynamic

Table E. **Text prompts** defined for the *nuScenes* [9] dataset (16 classes) in our proposed M3Net framework.

nuScenes (16 classes)		
#	**class**	**text prompt**
1	`barrier`	'barrier', 'barricade'
2	`bicycle`	'bicycle'
3	`bus`	'bus'
4	`car`	'car'
5	`construction-vehicle`	'bulldozer', 'excavator', 'concrete mixer', 'crane', 'dump truck'
6	`motorcycle`	'motorcycle'
7	`pedestrian`	'pedestrian', 'person'
8	`traffic-cone`	'traffic-cone'
9	`trailer`	'trailer', 'semi-trailer', 'cargo container', 'shipping container', 'freight container'
10	`truck`	'truck'
11	`driveable-surface`	'road'
12	`other-flat`	'curb', 'traffic island', 'traffic median'
13	`sidewalk`	'sidewalk'
14	`terrain`	'grass', 'grassland', 'lawn', 'meadow', 'turf', 'sod'
15	`manmade`	'building', 'wall', 'pole', 'awning'
16	`vegetation`	'tree', 'trunk', 'tree trunk', 'bush', 'shrub', 'plant', 'flower', 'woods'

environments, like autonomous driving, where understanding the temporal evolution of the scene is crucial. A key feature of the Minkowski convolution, and by extension MinkUNet, is its ability to perform convolutional operations on sparse data. This is achieved through the use of a generalized sparse convolution operation that can handle data in high-dimensional spaces while maintaining computational efficiency. The implementation of Minkowski convolutions is facilitated by the Minkowski Engine, a framework for high-dimensional sparse tensor operations. This engine enables the efficient implementation of the MinkUNet and other similar architectures. In this work, we resort to the Pointcept [8] implementation of MinkUNet and adopt the base version as our backbone network in M3Net. More details of this used backbone can be found at `https://github.com/Pointcept/Pointcept`.

C.3.2 PTv2+

PTv2+ [29] introduces an effective grouped vector attention (GVA) mechanism. GVA facilitates efficient information exchange both within and among attention groups, significantly enhancing the model's ability to process complex point cloud data. PTv2+ also introduces an improved position encoding scheme. This enhancement allows for better utilization of point cloud coordinates, thereby bolstering the spatial reasoning capabilities of the model. The additional position encoding multiplier strengthens the position information for attention, allowing for more accurate and detailed data processing. Extensive experiments demonstrate that PTv2+ achieves state-of-the-art performance on several challenging 3D point cloud understanding benchmarks. In this work, we resort to the Pointcept [8] implementation of PTv2+ implementation of MinkUNet and adopt the base version as our backbone network in M3Net. More details of this used backbone can be found at `https://github.com/Pointcept/Pointcept`.

C.4. Training Configuration

In this work, we implement the proposed M3Net framework based on Pointcept [8] and MMDetection3D [7]. We trained our baselines and M3Net on four A100 GPUs each with 80 GB memory. We adopt the AdamW optimizer [23] with a weight decay of 0.005 and a learning rate of 0.002. The learning rate scheduler utilized is cosine decay and the batch size is set to 6 for each GPU.

In the data-specific rasterization process, we rasterize the point clouds with voxel sizes tailored to the dataset characteristics. Specifically, we set the voxel sizes to [0.05m, 0.05m, 0.05m], [0.1m, 0.1m, 0.1m], and [0.05m, 0.05m, 0.05m] for the *SemanticKITTI* [1], *nuScenes* [9], and *Waymo Open* [28] datasets, respectively.

For data augmentation, we leverage several techniques, including random flips along the X, Y, and XY axes, and

Table F. **Text prompts** defined for the *SemanticKITTI* [1] dataset (19 classes) in our proposed M3Net framework.

SemanticKITTI (19 classes)		
#	**class**	**text prompt**
1	car	'car'
2	bicycle	'bicycle'
3	motorcycle	'motorcycle'
4	truck	'truck'
5	other-vehicle	'other vehicle', 'bulldozer', 'excavator', 'concrete mixer', 'crane', 'dump truck', 'bus', 'trailer', 'semi-trailer', 'cargo container', 'shipping container', 'freight container'
6	person	'person'
7	bicyclist	'bicyclist'
8	motorcyclist	'motorcyclist'
9	road	'road'
10	parking	'parking'
11	sidewalk	'sidewalk'
12	other-ground	'other ground', 'curb', 'traffic island', 'traffic median'
13	building	'building'
14	fence	'fence'
15	vegetation	'tree'
16	trunk	'tree trunk', 'trunk'
17	terrain	'grass', 'grassland', 'lawn', 'meadow', 'turf', 'sod'
18	pole	'pole'
19	traffic sign	'traffic sign'

random jittering within the range of [-0.02m, 0.02m]. Additionally, we incorporate global scaling and rotation, choosing scaling factors and rotation angles randomly from the intervals [0.9, 1.1] and [0, 2π], respectively. Furthermore, we integrate Mix3D [24] into our augmentation strategy during the training. There also exists some other augmentation techniques, such as LaserMix [19], PolarMix [30], RangeMix [16, 18], and FrustumMix [35].

For the network backbones, we have opted for MinkUNet [5] and PTv2+ [29]. In the case of MinkUNet, the encoder channels are set as $\{32, 64, 128, 256\}$, and the decoder channels are $\{256, 128, 64, 64\}$, each with a kernel size of 3. Meanwhile, for the PTv2+, the encoder channels are $\{32, 64, 128, 256, 512\}$, and the decoder channels are $\{64, 64, 128, 256\}$. For additional details, kindly refer to the original papers.

For the loss function, we incorporate the conventional cross-entropy loss and the Lovasz-softmax [2] loss to provide optimization for the LiDAR semantic and panoptic segmentation task. Additionally, we employ the L1 loss to optimize the instance head, aiding in the regression of precise instance offsets.

C.5. Evaluation Configuration

In this work, we follow the conventional reporting and employ the Intersection-over-Union (IoU) for individual classes and the mean Intersection-over-Union (mIoU) across all classes as our evaluation metrics for LiDAR semantic segmentation. Specifically, the IoU score for semantic class c is computed as follows:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}~. \tag{1}$$

Here, TP_c, FP_c, and FN_c represent the true positive, false positive, and false negative of class c, respectively. The mIoU score on each dataset is calculated by averaging the IoU scores across every semantic class. Notably, following recent works [29, 36], we report mIoU with Test Time Augmentation (TTA). For additional details, kindly refer to the original papers.

For panoptic LiDAR segmentation, we follow conventional reporting and utilize the Panoptic Quality (PQ) as our primary metric. The definition and calculation of the Panoptic Quality (PQ), Segmentation Quality (SQ), and Recogni-

Table G. **Text prompts** defined for the *Waymo Open* [28] dataset (22 classes) in our proposed M3Net framework.

Waymo Open (22 classes)		
#	**class**	**text prompt**
1	`car`	'car'
2	`truck`	'truck'
3	`bus`	'bus'
4	`other-vehicle`	'other vehicle', 'pedicab', 'construction vehicle', 'recreational vehicle', 'limo', 'tram', 'trailer', 'semi-trailer', 'cargo container', 'shipping container', 'freight container', 'bulldozer', 'excavator', 'concrete mixer', 'crane', 'dump truck'
5	`motorcyclist`	'motorcyclist'
6	`bicyclist`	'bicyclist'
7	`pedestrian`	'person', 'pedestrian'
8	`traffic-sign`	'traffic sign', 'parking sign', 'direction sign', 'traffic sign without pole', 'traffic light box'
9	`traffic-light`	'traffic light'
10	`pole`	'lamp post', 'traffic sign pole'
11	`construction-cone`	'construction cone'
12	`bicycle`	'bicycle'
13	`motorcycle`	'motorcycle'
14	`building`	'building'
15	`vegetation`	'bushes', 'tree branches', 'tall grasses', 'flowers', 'grass', 'grassland', 'lawn', 'meadow', 'turf', 'sod'
16	`tree-trunk`	'tree trunk', 'trunk'
17	`curb`	'curb'
18	`road`	'road'
19	`lane-marker`	'lane marker'
20	`other-ground`	'other ground', 'bumps', 'cateyes', 'railtracks'
21	`walkable`	'walkable', 'grassy hill', 'pedestrian walkway stairs'
22	`sidewalk`	'sidewalk'

tion Quality (RQ) scores are given as follows:

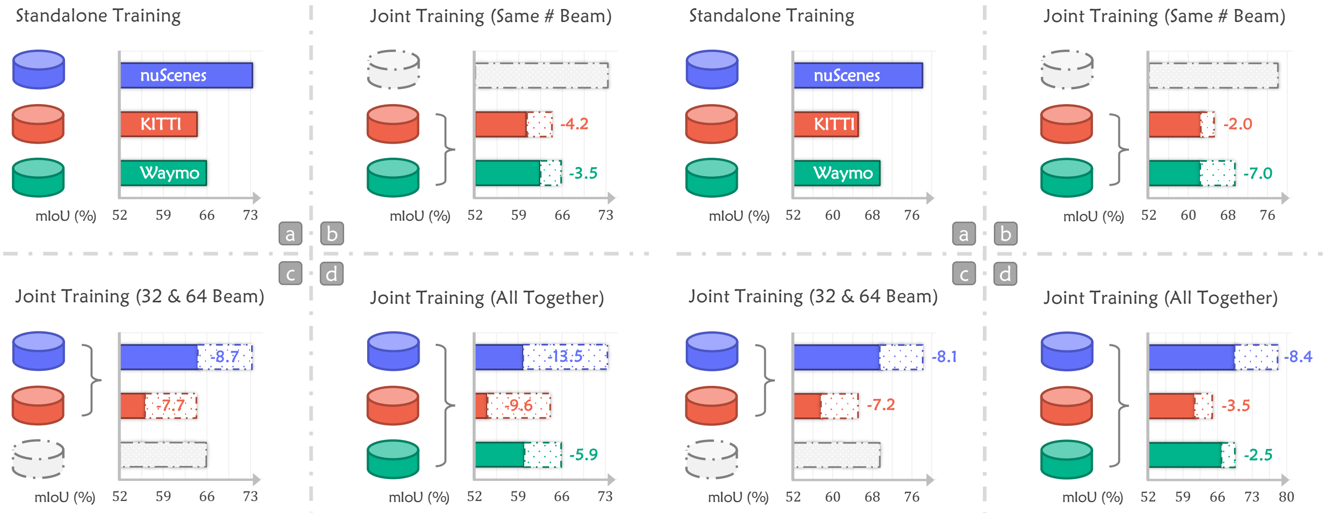
$$\text{PQ} = \underbrace{\frac{\sum_{(i,j)\in TP}\text{IoU}(i,j)}{|TP|}}_{\text{SQ}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{RQ}} . \tag{2}$$

The three aforementioned metrics are also calculated individually for *things* and *stuff* classes, resulting in PQth, SQth, RQth, and PQst, SQst, RQst. Additionally, we also report the PQ† score as widely used in many prior works [11, 20, 27, 37]. This metric is defined by exchanging the PQ of each *stuff* class with its IoU and then averaging across all semantic classes. For additional details, kindly refer to the original papers.

To further assess the capability of a LiDAR segmentation model for out-of-training-distribution generalization, we follow Robo3D [17] and adopt the corruption error (CE) and resilience rate (RR), as well as the mean corruption error (mCE) and mean resilience rate (mRR) as the evaluation metrics in comparing the robustness. To normalize the severity effects, we chose MinkUNet [5] as the baseline model. The CE and mCE scores are calculated as follows:

$$\text{CE}_k = \frac{\sum_{l=1}^{3}(1-\text{Acc}_{k,l})}{\sum_{l=1}^{3}(1-\text{Acc}_{k,l}^{\text{baseline}})} , \quad \text{mCE} = \frac{1}{N}\sum_{k=1}^{N}\text{CE}_k , \tag{3}$$

where $\text{Acc}_{k,l}$ denotes mIoU scores on corruption type k at severity level l. $N = 8$ is the total number of corruption types. The mRR serves as the relative robustness indicator for measuring how much accuracy a model can retain when evaluated on the corruption sets. The RR and mRR scores



(a) M3Net (*w/* MinkUNet [5] backbone)

(b) M3Net (*w/* PTv2+ [29] backbone)

Figure A. **A pilot study** of naïvely merging different datasets for training the MinkUNet [5] model. Compared to the standalone training in **(a)**, either jointly training with **(b)** the same, **(c)** different, or **(d)** all sensor-acquired data will cause severe degradation. Subfigure (a): M3Net *w/* a MinkUNet [5] backbone. Subfigure (b): M3Net *w/* a PTv2+ [29] backbone.

are calculated as follows:

$$\text{RR}_k = \frac{\sum_{l=1}^{3} \text{Acc}_{k,l}}{3 \times \text{Acc}_{\text{clean}}}~, \qquad \text{mRR} = \frac{1}{N}\sum_{k=1}^{N} \text{RR}_k~, \qquad (4)$$

where $\text{Acc}_{\text{clean}}$ denotes the mIoU score on the clean validation set of each dataset. Kindly refer to the original paper for additional details.

D. Additional Experimental Results

In this section, we present the complete experimental results as a supplement to the findings and conclusions drawn in the main body of this paper.

D.1. Pilot Study

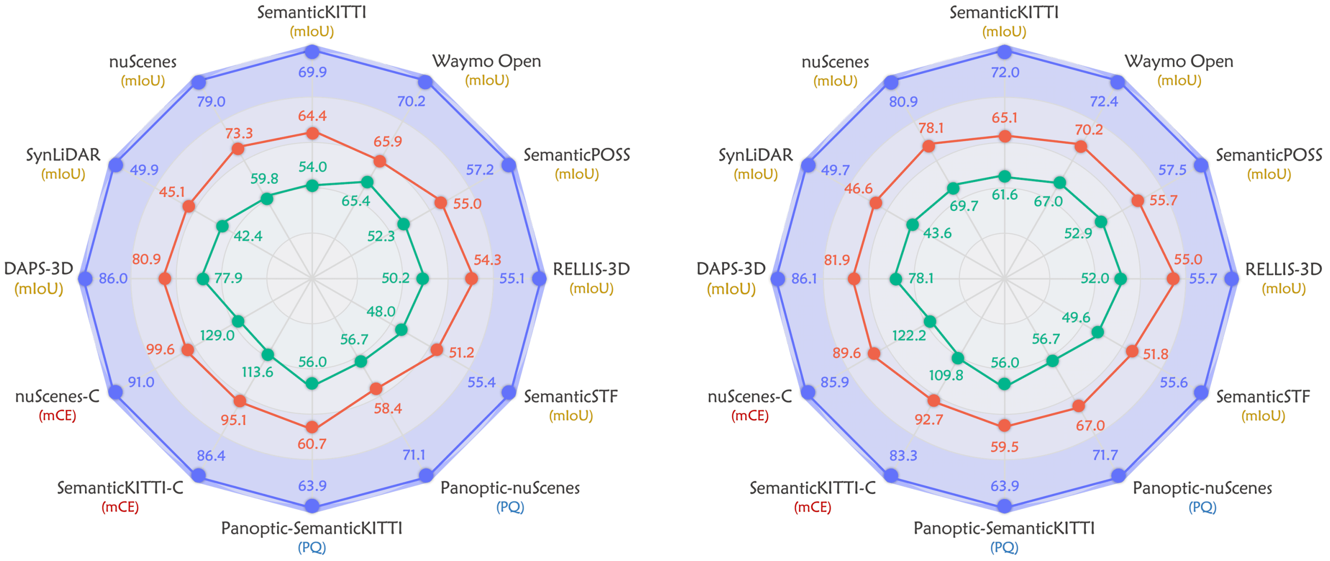
In the main body of this paper, we conduct a pilot study to showcase the potential problems in the Single-Dataset Training and Naïve Joint Training pipelines. Specifically, we observe that it is non-trivial to naïvely combine heterogeneous data from different driving datasets with large data distribution and sensor configuration gaps to train a universal LiDAR segmentation model.

We show in Fig. A our pilot study with the MinkUNet [5] backbone in subfigure (a) and the PTv2+ [29] backbone in subfigure (b), for both standalone and joint training setups. As can be seen, using either the classical MinkUNet or the most recent PTv2+ as the backbone, the brutal combination will undermine the segmentation performance. Due to large discrepancies in aspects like sensor configurations, data acquisitions, label mappings, and domain shifts, the jointly trained representations tend to be disruptive instead of being more general. Such degradation is particularly overt using naïvely combining LiDAR data acquired by different sensor setups, such as the direct merge of *nuScenes* [9] (Velodyne HDL32E with 32 laser beams) and *SemanticKITTI* [1] (Velodyne HDL-64E with 64 laser beams).

Meanwhile, we also supplement the complete comparison results among the Single-Dataset Training, Naïve Joint Training, and our proposed M3Net pipelines and show the results in Fig. B. As can be seen, compared to the Single-Dataset Training baselines, a naïve merging of heterogeneous LiDAR data will cause severe performance degradation. This observation holds true for both the MinkUNet [5] backbone as in Fig. Ba and the PTv2+ [29] backbone as in Fig. Bb, which highlights again the importance of conducting alignments when merging multiple driving datasets for training. Notably, after proper data, feature, and label space alignments, we are able to combine the advantage of leveraging the diverse training data sources and achieve better performance than the Single-Dataset Training baselines. Such improvements are holistic, as shown in the radar charts, our proposed M3Net achieves superior performance gains over the baselines under all the tested scenarios across all twelve LiDAR segmentation datasets.

D.2. Ablation Study

In this section, we supplement more fine-grained ablation analysis in the third column of Fig. C and Fig. D on the *SemanticKITTI* [1], *nuScenes* [9], and *Waymo Open* [28] datasets. The results verify the effectiveness of each of the



(a) M3Net (*w/* MinkUNet [5] backbone)

(b) M3Net (*w/* PTv2+ [29] backbone)

Figure B. Performance comparisons among **M3Net** [●], *Single-Dataset Training* [●], and *Naïve Joint Training* [●] across **twelve** LiDAR segmentation datasets. Subfigure (a): M3Net *w/* a MinkUNet [5] backbone. Subfigure (b): M3Net *w/* a PTv2+ [29] backbone. For better comparisons, the radius is normalized based on M3Net's scores. The larger the area coverage, the higher the overall performance.

three alignments proposed in M3Net.

D.3. LiDAR Panoptic Segmentation

In this section, we supplement the PQ, RQ, and SQ scores, as well as their fine-grained scores regarding the *things* and *stuff* classes for our panoptic LiDAR segmentation experiments on the *SemanticKITTI* [1] and *Panoptic-nuScenes* [9] datasets.

D.3.1 Panoptic-SemanticKITTI

For the detailed PQ, RQ, and SQ scores of our comparative study on the *SemanticKITTI* [1] dataset, we supplement Tab. H to facilitate detailed comparisons with state-of-the-art LiDAR segmentation approaches on the validation set. We observe that the proposed M3Net is capable of achieving new arts on the validation set, especially for the more fine-grained metrics like RQ and SQ. The results verify the effectiveness of the proposed M3Net compared to the singe-dataset training and naïve joint training baselines.

D.3.2 Panoptic-nuScenes

For the detailed PQ, RQ, and SQ scores of our comparative study on the *Panoptic-nuScenes* [9] dataset, we supplement Tab. H to facilitate detailed comparisons with state-of-the-art LiDAR segmentation approaches on the validation set. We observe that the proposed M3Net is capable of achieving new arts on the validation set, across almost all the fine-grained metrics. The results verify the effectiveness of the proposed M3Net compared to the singe-dataset training and naïve joint training baselines.

D.4. Out-of-Distribution Generalization

In this section, we supplement the class-wise CE and RR scores of the out-of-training-distribution generalization experiments on the *SemanticKITTI-C* and *nuScenes-C* datasets in the Robo3D [17] benchmark. Specifically, Tab. I and Tab. J show the per-corruption IoU scores of prior works, our baselines, and the proposed M3Net on the *SemanticKITTI-C* and *nuScenes-C* datasets, respectively. We observe that M3Net sets up clear superiority over prior arts across almost all eight corruption types. Such robust feature learning is crucial to the safe operation of autonomous vehicles under out-of-training-distribution scenarios, especially in safety-critical areas [17, 33, 34].

E. Qualitative Assessment

In this section, we provide a comprehensive qualitative assessment to validate further the effectiveness and superiority of the proposed M3Net framework.

E.1. Visual Comparisons

We supplement several qualitative comparisons of our proposed M3Net over the single-dataset training baseline. Specifically, the visual comparisons across the *SemanticKITTI* [1], *nuScenes* [9], and *Waymo Open* [28]

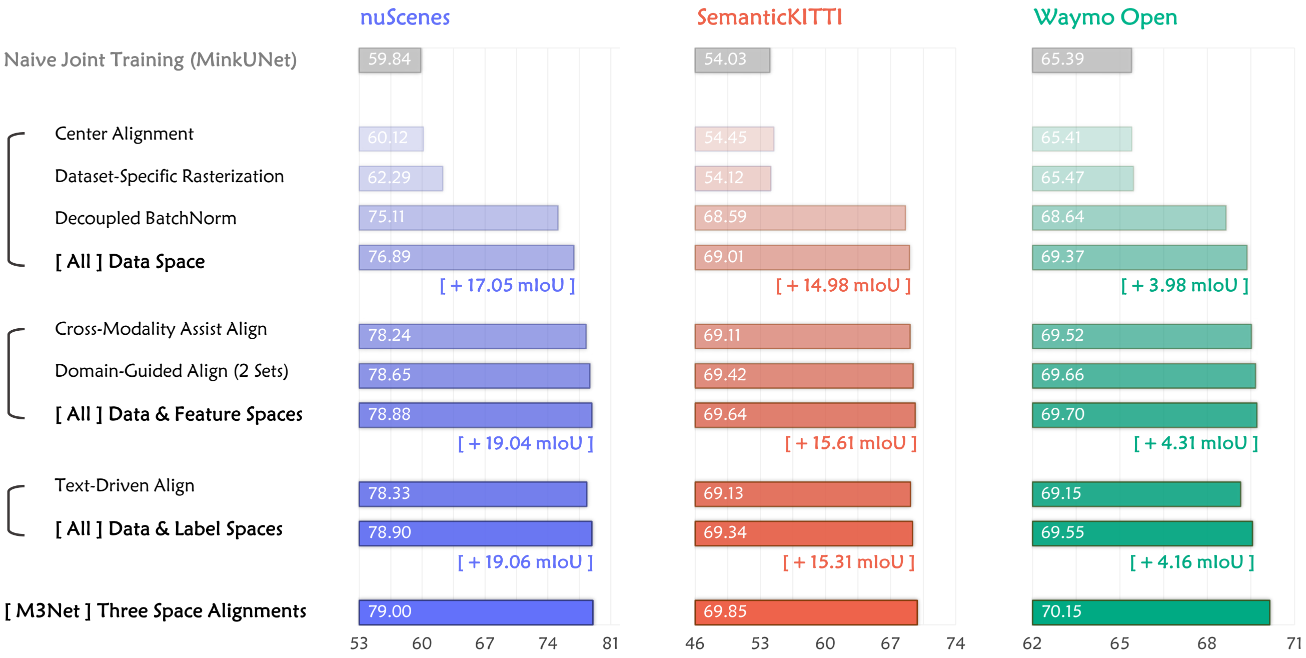


Figure C. **Ablation study** of the data, feature, and label space alignments in the proposed M3Net (*w/* MinkUNet [5] backbone).

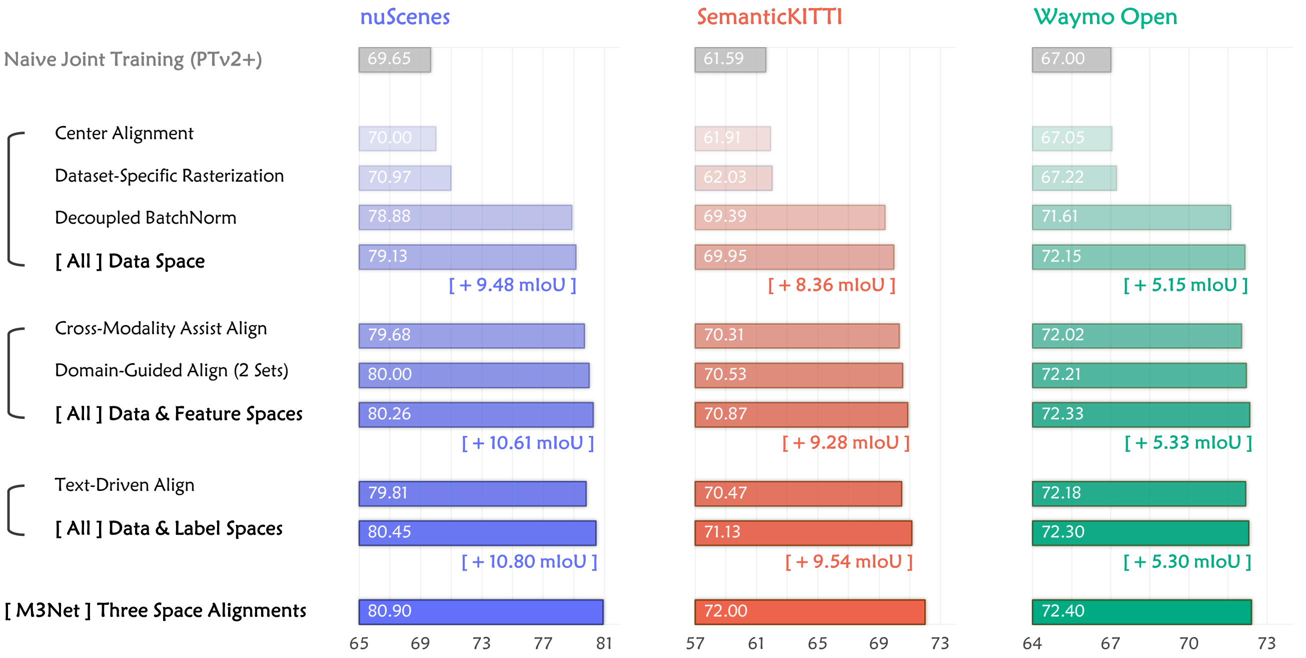


Figure D. **Ablation study** of the data, feature, and label space alignments in the proposed M3Net (*w/* PTv2+ [29] backbone).

datasets are shown in Fig. E, Fig. F, and Fig. G, respectively. As we can see, the proposed M3Net shows superior performance than the baseline under different driving scenarios. Such results highlight the effectiveness of the proposed M3Net in enhancing performance in the multi-task, multi-dataset, multi-modality training setting. Additionally, we present qualitative results in Fig. H to showcase the capability of M3Net in tackling both the LiDAR semantic segmentation and panoptic segmentation tasks. As we can see, the proposed M3Net demonstrates effectiveness in making accurate predictions among the complex object and background classes in the driving scenes, underscoring its effectiveness in handling multi-task LiDAR segmentation.

F. Broader Impact

In this section, we elaborate on the positive societal influence and potential limitations of our multi-task, multi-dataset, multi-modality LiDAR segmentation framework.

F.1. Positive Societal Influence

In this work, we present a versatile LiDAR segmentation framework dubbed M3Net for conducting multi-task, multi-dataset, multi-modality LiDAR segmentation in a unifying pipeline. LiDAR segmentation is crucial for the development of safe and reliable autonomous vehicles. By accurately interpreting the vehicle surroundings, LiDAR helps in obstacle detection, pedestrian safety, and navigation, thereby reducing the likelihood of accidents and enhancing road safety. LiDAR segmentation contributes significantly to societal welfare through its applications in various fields. Its ability to provide accurate, detailed 3D representations of physical environments enables more informed decision-making, enhances safety, and promotes sustainability.

F.2. Potential Limitation

Although our proposed M3Net is capable of leveraging multiple heterogeneous driving datasets to train a versatile LiDAR segmentation network and achieve promising universal LiDAR segmentation results, there still exists room for improvement. Firstly, our framework leverages calibrated and synchronized camera data to assist the alignments. Such a requirement might not be met in some older LiDAR segmentation datasets. Secondly, we do not handle the minority classes during multi-dataset learning, especially for some dynamic classes that are uniquely defined by a certain dataset. Thirdly, we do not consider the combination of simulation data with real-world LiDAR point clouds. We believe these aspects are promising for future work to further improve our multi-task, multi-dataset, multi-modality LiDAR segmentation framework.

G. Public Resources Used

In this section, we acknowledge the use of datasets, models, and codebases, during the course of this work.

G.1. Public Datasets Used

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

• nuScenes[3]	CC BY-NC-SA 4.0
• nuScenes-devkit[4]	Apache License 2.0
• SemanticKITTI[5]	CC BY-NC-SA 4.0
• SemanticKITTI-API[6]	MIT License
• Waymo Open Dataset[7]	Waymo Dataset License
• RELLIS-3D[8]	CC BY-NC-SA 3.0
• SemanticPOSS[9]	Unknown
• SemanticSTF[10]	CC BY-NC-SA 4.0
• SynLiDAR[11]	MIT License
• DAPS-3D[12]	MIT License
• Robo3D[13]	CC BY-NC-SA 4.0

G.2. Public Models Used

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

• MinkowskiEngine[14]	MIT License
• PointTransformerV2[15]	Unknown
• spvnas[16]	MIT License
• Cylinder3D[17]	Apache License 2.0
• SLidR[18]	Apache License 2.0
• OpenSeeD[19]	Apache License 2.0
• segment-anything[20]	Apache License 2.0
• Segment-Any-Point-Cloud[21]	CC BY-NC-SA 4.0
• Mix3D[22]	Unknown
• LaserMix[23]	CC BY-NC-SA 4.0

G.3. Public Codebases Used

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

• mmdetection3d[24]	Apache License 2.0
• Pointcept[25]	MIT License
• OpenPCSeg[26]	Apache License 2.0

[3] https://www.nuscenes.org/nuscenes.

[4] https://github.com/nutonomy/nuscenes-devkit.

[5] http://semantic-kitti.org.

[6] https://github.com/PRBonn/semantic-kitti-api.

[7] https://waymo.com/open.

[8] http://www.unmannedlab.org/research/RELLIS-3D.

[9] http://www.poss.pku.edu.cn/semanticposs.html.

[10] https://github.com/xiaoaoran/SemanticSTF.

[11] https://github.com/xiaoaoran/SynLiDAR.

[12] https://github.com/subake/DAPS3D.

[13] https://github.com/ldkong1205/Robo3D.

[14] https://github.com/NVIDIA/MinkowskiEngine.

[15] https://github.com/Gofinge/PointTransformerV2.

[16] https://github.com/mit-han-lab/spvnas.

[17] https://github.com/xinge008/Cylinder3D.

[18] https://github.com/valeoai/SLidR.

[19] https://github.com/IDEA-Research/OpenSeeD.

[20] https://github.com/facebookresearch/segment-anything.

[21] https://github.com/youquanl/Segment-Any-Point-Cloud

[22] https://github.com/kumuji/mix3d.

[23] https://github.com/ldkong1205/LaserMix.

[24] https://github.com/open-mmlab/mmdetection3d.

[25] https://github.com/Pointcept/Pointcept.

[26] https://github.com/PJLab-ADG/OpenPCSeg.

Table H. **The class-wise panoptic segmentation scores** on the *val* sets of the *Panoptic-SemanticKITTI* [1] and *Panoptic-nuScenes* [9] datasets. All scores are given in percentage (%). For each evaluated metric: **bold** - best in column; underline - second best in column.

Method	**Panoptic-SemanticKITTI**					**Panoptic-nuScenes**				
	PQ	PQ†	RQ	SQ	mIoU	PQ	PQ†	RQ	SQ	mIoU
Panoptic-TrackNet [13]	40.0	-	48.3	73.0	53.8	51.4	56.2	63.3	80.2	58.0
Panoptic-PolarNet [37]	59.1	64.1	70.2	78.3	64.5	63.4	67.2	75.3	83.9	66.9
EfficientLPS [27]	59.2	65.1	69.8	75.0	64.9	59.2	62.8	82.9	70.7	69.4
DSNet [11]	61.4	65.2	72.7	79.0	69.6	64.7	67.6	76.1	83.5	76.3
Panoptic-PHNet [20]	61.7	-	-	-	65.7	**74.7**	**77.7**	**84.2**	**88.2**	79.7
Naïve Joint (MinkUNet)	47.8	54.1	56.9	71.6	54.0	45.0	50.3	55.3	79.4	59.8
Single-Dataset (MinkUNet)	60.7	65.6	70.6	**83.2**	64.4	58.4	62.7	82.9	69.3	73.3
M3Net (MinkUNet)	**63.9**	68.5	**73.2**	82.3	69.9	67.9	71.1	78.1	85.9	79.0
Naïve Joint (PTv2+)	56.0	59.6	65.8	73.7	61.6	56.7	60.6	66.8	83.5	69.7
Single-Dataset (PTv2+)	59.5	63.6	69.5	75.3	65.1	67.0	69.8	77.8	85.0	78.1
M3Net (PTv2+)	**63.9**	**68.7**	73.1	82.4	**72.0**	71.7	74.0	82.2	86.5	**80.9**

Table I. **The class-wise robustness evaluation scores** on the *SemanticKITTI-C* dataset from the Robo3D benchmark [17]. All scores are given in percentage (%). For each evaluated metric: **bold** - best in column; underline - second best in column.

Method	**mCE ↓**	**mRR ↑**	fog	wet-ground	snow	motion-blur	beam-missing	crosstalk	incomplete-echo	cross-sensor
Naïve Joint (MinkUNet)	113.7	84.7	48.5	54.0	39.8	41.1	49.1	39.8	47.7	46.1
Single-Dataset (MinkUNet)	95.1	85.0	50.6	52.3	51.4	54.5	59.3	**56.9**	56.2	56.6
M3Net (MinkUNet)	86.4	85.8	56.7	63.8	**55.1**	63.3	64.5	50.6	60.7	**58.1**
Naïve Joint (PTv2+)	109.8	77.5	49.7	53.1	43.6	45.3	51.6	39.7	50.2	48.8
Single-Dataset (PTv2+)	92.7	**85.9**	52.3	53.7	51.8	55.8	60.2	56.4	59.3	57.6
M3Net (PTv2+)	**83.3**	84.0	**60.4**	**66.1**	52.7	**63.9**	**65.1**	55.1	**62.6**	57.9

Table J. **The class-wise robustness evaluation scores** on the *nuScenes-C* dataset from the Robo3D benchmark [17]. All scores are given in percentage (%). For each evaluated metric: **bold** - best in column; underline - second best in column.

Method	**mCE ↓**	**mRR ↑**	fog	wet-ground	snow	motion-blur	beam-missing	crosstalk	incomplete-echo	cross-sensor
PPKT [22]	105.6	76.1	64.0	72.2	59.1	57.2	63.9	36.3	60.6	39.6
SLidR [26]	106.1	76.0	65.4	72.3	56.0	56.1	62.9	41.9	61.2	38.9
Seal [21]	92.6	**83.1**	**72.7**	74.3	**66.2**	66.1	66.0	**57.4**	59.9	39.9
Naïve Joint (MinkUNet)	129.0	81.5	54.0	57.3	50.9	57.5	47.3	42.3	49.4	30.9
Single-Dataset (MinkUNet)	99.6	79.1	60.7	74.6	50.8	65.0	67.1	32.4	63.2	50.0
M3Net (MinkUNet)	91.0	79.2	62.5	76.2	49.7	75.4	66.2	43.3	64.7	52.5
Naïve Joint (PTv2+)	122.2	73.4	55.2	60.0	51.4	58.7	52.7	43.3	52.9	34.7
Single-Dataset (PTv2+)	89.6	79.1	63.1	76.4	51.6	75.2	67.9	41.4	65.4	53.5
M3Net (PTv2+)	**85.9**	78.2	54.4	**78.0**	51.2	**76.8**	**68.0**	44.3	**66.7**	**55.9**

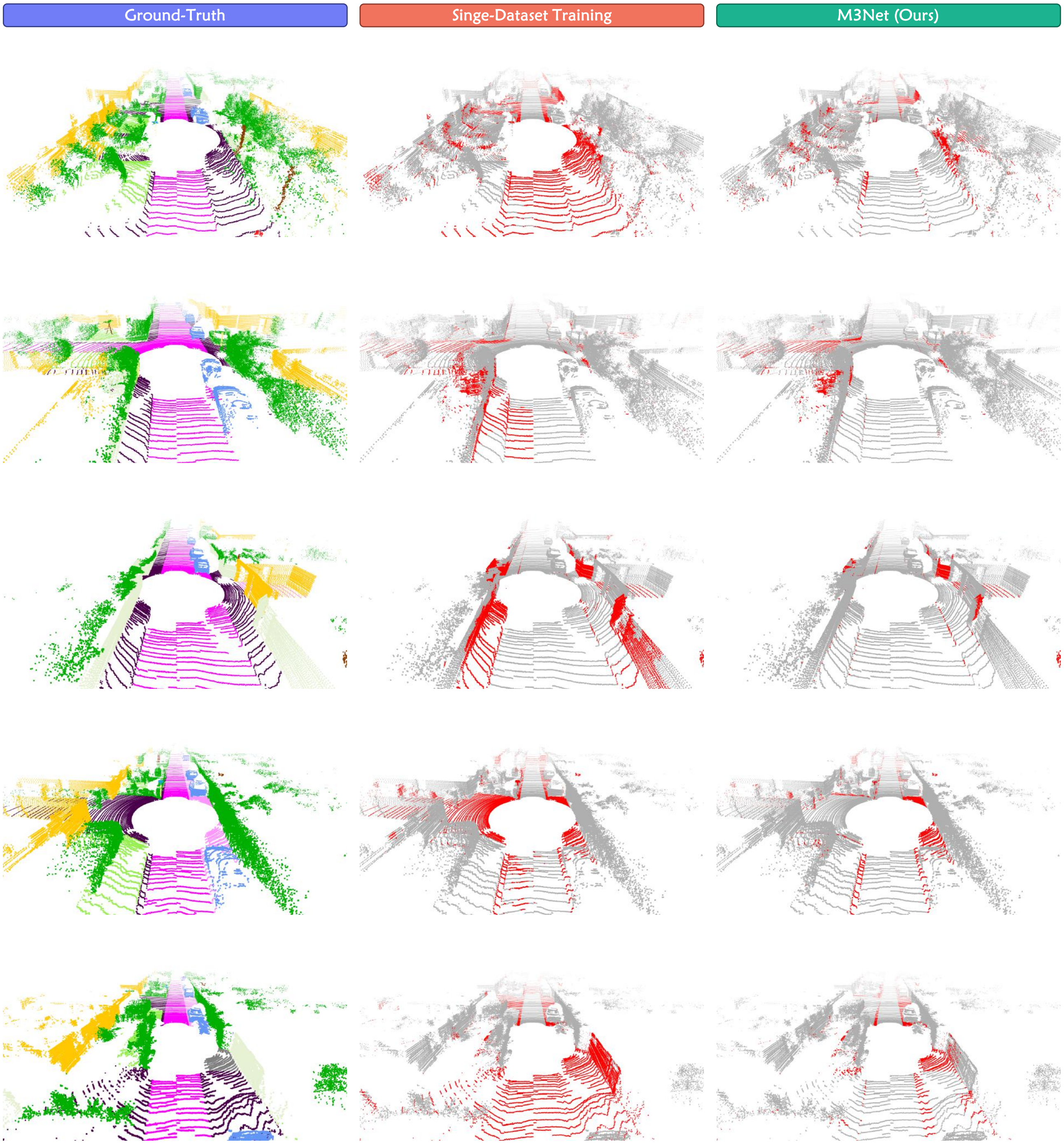


Figure E. **Qualitative comparisons** between the Single-Dataset Training and the proposed M3Net for LiDAR semantic segmentation on the *SemanticKITTI* dataset [1]. To highlight the differences, the correct / **incorrect** predictions are painted in **gray** / **red**, respectively.

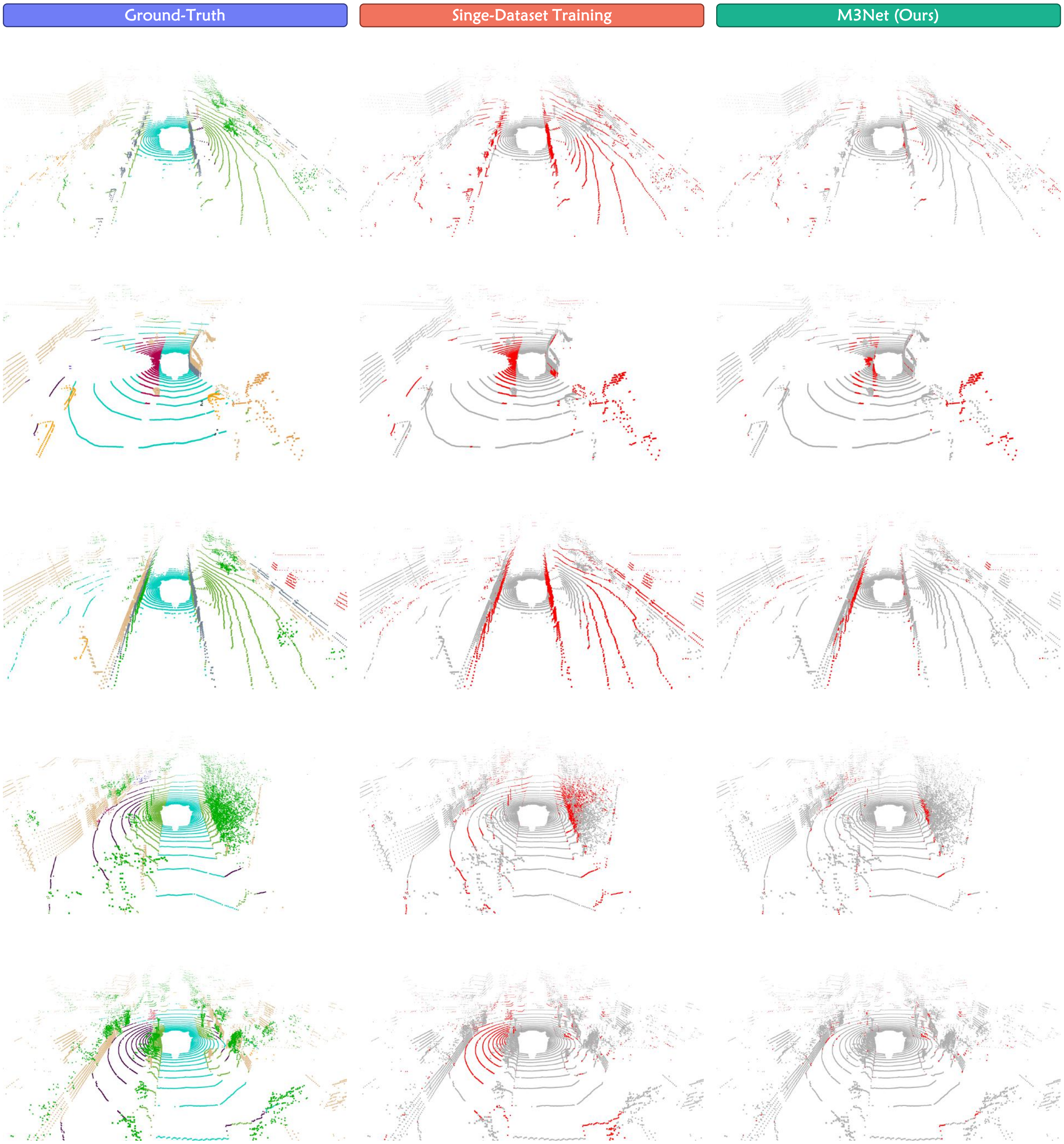


Figure F. **Qualitative comparisons** between the Single-Dataset Training and the proposed M3Net for LiDAR semantic segmentation on the *nuScenes* dataset [9]. To highlight the differences, the **correct** / **incorrect** predictions are painted in **gray** / **red**, respectively.

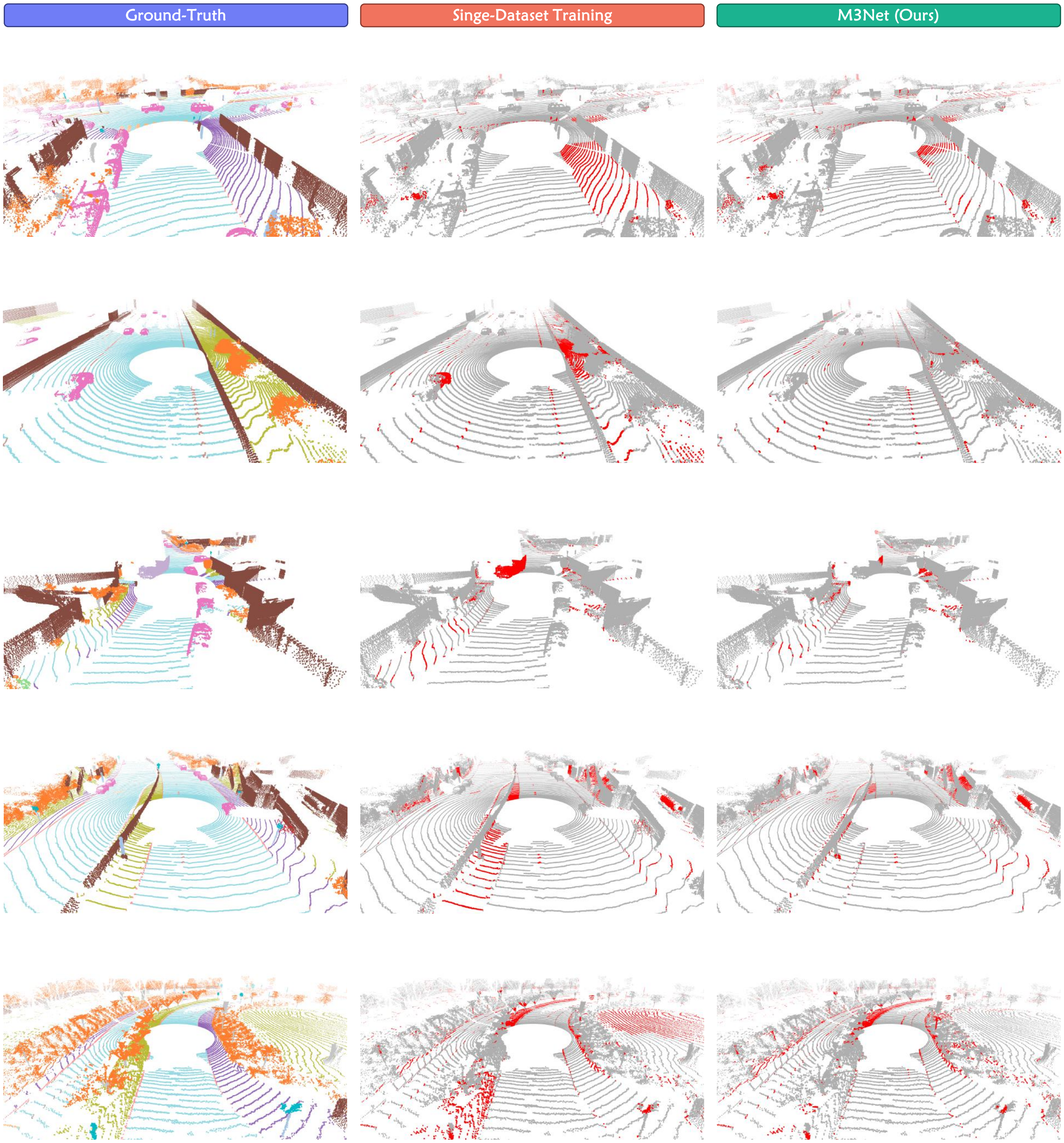


Figure G. **Qualitative comparisons** between the Single-Dataset Training and the proposed M3Net for LiDAR semantic segmentation on the *Waymo Open* dataset [28]. To highlight the differences, the **correct** / **incorrect** predictions are painted in **gray** / **red**, respectively.

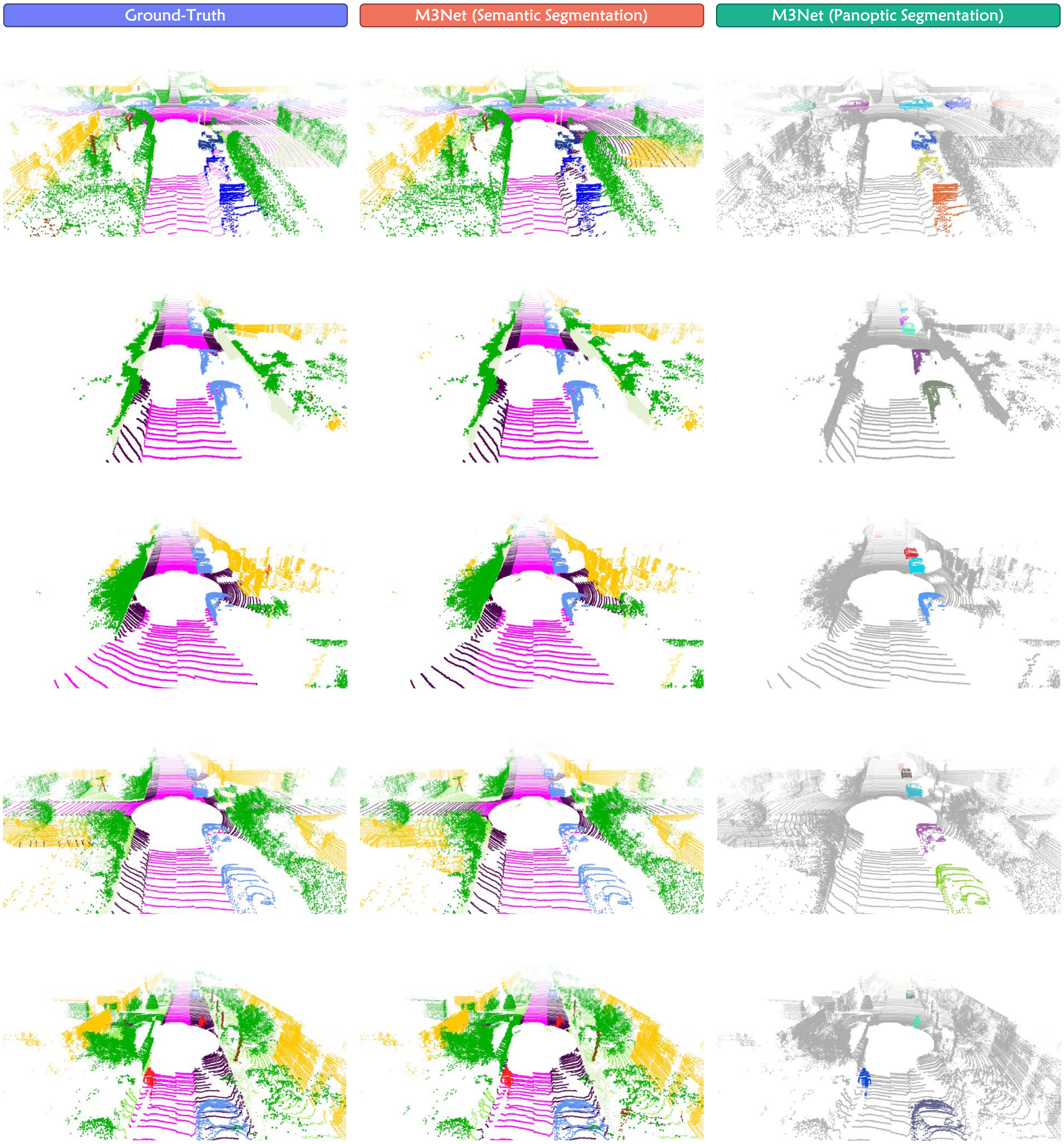


Figure H. **Qualitative comparisons** between the Ground-Truth and the proposed M3Net for LiDAR panoptic segmentation on the *SemanticKITTI* dataset [1]. To highlight the panoptic segmentation effect, the semantic predictions in the third column are painted in **gray**. For panoptic segmentation predictions, each color-coded cluster represents a distinct instance.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 1, 2, 3, 5, 7, 8, 9, 11, 12, 15, 16, 19

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 9

[3] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 3

[4] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 3

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 7, 9, 10, 11, 12, 13

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2005. 7

[7] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020. 8

[8] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. `https://github.com/Pointcept/Pointcept`, 2023. 8

[9] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7:3795–3802, 2022. 1, 2, 3, 4, 7, 8, 11, 12, 15, 17

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[11] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13090–13099, 2021. 7, 10, 15

[12] Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Unified 3d and 4d panoptic segmentation via dynamic shifting networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–16, 2024. 7

[13] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 15

[14] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripallig. Rellis-3d dataset: Data, benchmarks and analysis. In *IEEE International Conference on Robotics and Automation*, pages 1110–1116, 2021. 2, 7

[15] Alexey Klokov, Di Un Pak, Aleksandr Khorin, Dmitry Yudin, Leon Kochiev, Vladimir Luchinskiy, and Vitaly Bezuglyj. Daps3d: Domain adaptive projective segmentation of 3d lidar point clouds. *IEEE Access*, 11:79341–79356, 2023. 2, 7

[16] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023. 9

[17] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 2, 3, 7, 10, 12, 15

[18] Lingdong Kong, Niamul Quader, and Venice Erin Liong. Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. In *IEEE International Conference on Robotics and Automation*, pages 9338–9345, 2023. 9

[19] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023. 9

[20] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11809–11818, 2022. 10, 15

[21] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, 2023. 7, 15

[22] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H. Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.0468*, 2021. 15

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 8

[24] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *International Conference on 3D Vision*, pages 116–125, 2021. 9

[25] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020. 2, 7

[26] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022. 15

[27] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 2021. 10, 15

[28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 18

[29] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems*, 2022. 7, 8, 9, 11, 12, 13

[30] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. In *Advances in Neural Information Processing Systems*, pages 11035–11048, 2022. 9

[31] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2795–2803, 2022. 2, 7

[32] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023. 2, 7

[33] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robobev: Towards robust bird's eye view perception under corruptions. *arXiv preprint arXiv:2304.06719*, 2023. 12

[34] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and analyzing bird's eye view perception robustness to corruptions. *Preprint*, 2023. 12

[35] Xiang Xu, Lingdong Kong, Hui Shuai, and Qingshan Liu. Frnet: Frustum-range networks for scalable lidar segmentation. *arXiv preprint arXiv:2312.04484*, 2023. 9

[36] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, pages 677–695, 2022. 9

[37] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 10, 15