

Received 6 July 2023, accepted 20 July 2023, date of publication 25 July 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298706

RESEARCH ARTICLE

DAPS3D: Domain Adaptive Projective Segmentation of 3D LiDAR Point Clouds

ALEXEY A. KLOKOV^{1,2}, DI UN PAK³, ALEKSANDR KHORIN³, DMITRY A. YUDIN^{3,4},
LEON KOCHIEV², VLADIMIR D. LUCHINSKIY², AND VITALY D. BEZUGLYJ³

¹Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 119333 Moscow, Russia

²Autonomics.tech, 3032 Limassol, Cyprus

³Moscow Institute of Physics and Technology, Dolgoprudny, 141701 Moscow, Russia

⁴Artificial Intelligence Research Institute (AIRI), 121170 Moscow, Russia

Corresponding author: Dmitry A. Yudin (yudin.da@mipt.ru)

This work was supported by the Russian Science Foundation under Project 21-71-00131.

ABSTRACT LiDARs are one of the key sources of reliable environmental ranging information for autonomous vehicles. However, segmentation of 3D scene elements (roads, buildings, people, cars, etc.) based on LiDAR point clouds has limitations. On the one hand, point- and voxel-based segmentation neural networks do not offer sufficiently high speed. On the other hand, modern labeled datasets primarily consist of street scenes recorded for driverless cars and contain little data for mobile delivery robots or cleaners that must work in parks and yards with heavy pedestrian traffic. This article aims to overcome these limitations. We have proposed a novel approach called DAPS3D to train deep neural networks for 3D semantic segmentation. This approach is based on a spherical projection of a point cloud and LiDAR-specific masks, enabling the model to adapt to different types of LiDAR. First of all, we have introduced various high-speed multi-scale spherical projection segmentation models, including convolutional, recurrent, and transformer architectures. Among them, the SalsaNextRecLSTM architecture with recurrent blocks showed the best results. In particular, this model achieved the 83.5% mIoU metric for the SemanticKitti dataset with joint categories. Secondly, we have proposed several original augmentations for spherical projections of LiDAR data, including FoV, flip, and rotation augmentation, as well as a special T-Zone cutout. These augmentations increase the model's invariance when dealing with changes in the data domain. Finally, we introduce a new method to generate synthetic datasets for domain adaptation problems. We have developed two new datasets for validating 3D scene outdoor segmentation algorithms: the DAPS-1 dataset, which is based on the augmentation of the reconstructed 3D semantic map, and the DAPS-2 LiDAR dataset, collected by the on-board sensors of a cleaning robot in a park area. Particular attention is given to the performance of the developed models, demonstrating their ability to function in real-time. The code and datasets used in this study are publicly available at: github.com/subake/DAPS3D.

INDEX TERMS Autonomous vehicles, computer vision, artificial neural networks, projection algorithm, image segmentation.

I. INTRODUCTION

The self-driving cars are ever-evolving during the last decade due to the active development of deep learning. This brings the question of the safety and reliability of the

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu .

navigational system. A key challenge of the navigational pipeline is semantic segmentation.

Semantic segmentation of 3D point clouds is one of the essential building blocks for autonomous driving systems [1], [2], [3], [4]. It leverages the precise and dense range data from LiDARs to better understand the driving scene and helps to distinguish vulnerable road agents.

The most popular are several main groups of methods for solving the problem of point cloud segmentation. All of them are learnable and based on different architectures of deep neural networks [5], [6], [7], [8], [9]. One of their key differences is the 3D scene representation. It can be based on points [10], [11], voxels [6], [12], point cloud projection onto a sphere [3], [7], as well as various combinations of these approaches [5], [8]. A separate research area is the fusion of various sensor sources (including onboard cameras) to improve the quality of segmentation of LiDAR point clouds [13], [14].

Currently, state-of-the-art models can solve the segmentation problem on public datasets with high quality [3], [6], [9], [15], [16]. However, the metrics of the models can vary drastically in different environments.

The group of projective neural network methods [3], [7], [17] has the highest performance and real-time potential. But at the same time, they are very sensitive to data domain change like all image-based approaches.

In this article, we explore segmentation of LiDAR point clouds from a data domain when a mobile ground robot with a special sensor setup has to move through park areas with heavy pedestrian traffic. Existing datasets and neural network models are not quite suitable for high-precision and fast 3D segmentation in this case.

To overcome this problem, we propose a novel approach called DAPS3D for domain adaptive projective segmentation of 3D LiDAR point clouds. It includes new augmentation techniques for 3D LiDAR point clouds to make the final model robust to the domain shift:

- The augmentation sets we are investigating, including FoV, flip, noise, dropout, and rotation augmentation, as well as a special T-Zone cutout, have not previously been considered in public sources.
- The labeling enhancing process proposed by us made it possible to augment and improve the labeling quality of the RELLIS-3D open dataset [18] from a domain close to ours.
- The proposed original approach to building a semi-synthetic dataset by adding 3D human models to the existing popular SemanticKITTI dataset [15] has been demonstrated in the creation of the DAPS-1 dataset.

Another important part of our DAPS3D approach is the development and study of novel neural network models for projective point cloud semantic segmentation:

- We propose SalsaNetRec, SalsaNetRecLSTM, and SalsaNextRecLSTM architectures of deep neural networks. Their difference from existing SalsaNet [2] and SalsaNext [7] projective baselines is the use of recurrent blocks in the model. This allows us to extract more useful information from point cloud sequences. For the case with the SalsaNextRecLSTM model, it leads to a significant improvement in the segmentation quality.
- We first tried to adapt fast 2D segmentation methods based on convolutional DDRNet [19] and SegFormer

transformer [20] for projective 3D point cloud segmentation.

An additional contribution is that we have developed and presented two new datasets to validate the quality of the proposed DAPS3D approach: the DAPS-1 dataset based on the original augmentation of the reconstructed 3D semantic map, and the DAPS-2 LiDAR dataset, collected by the onboard sensors of the cleaning robot in the unique park area.

We have demonstrated that the proposed segmentation models from DAPS3D have real-time performance and competitive quality for application in computer vision systems of autonomous robots and vehicles.

II. RELATED WORK

Over the last five years, significant progress in semantic segmentation has been achieved. The main differences between state-of-the-art models lay in the neural network architecture, sensor setup, and data representation. In this section, we review the current state of the semantic segmentation task, paying close attention to the data representation and domain stability.

A. POINT- AND VOXEL-BASED POINT CLOUD SEGMENTATION

The first group of models uses raw point clouds to operate with the data [11], [16], [21]. Models that work with point-wise representation use 3D convolutional blocks that aggregate information about the geometry and relative positions of the points. The downsides of such approaches are their time complexity, sensor sensitivity, and nonmonotonicity of the data. All of that makes them impossible to be used in real-time computation in the outdoor environment. However, in offline problem statements, such models often outperform others. There are different noteworthy papers on point-wise models. Thus said, the authors of S3Net [10] have shown, that point-wise models with sparse convolutions can achieve state-of-the-art performance. Authors of [22] showed that such models can also achieve high inference speed using random sampling.

A logical continuation of point-wise methods is voxel-wise methods. They split the space into a grid and aggregate information about every cell. The size of the voxels brings a tradeoff of speed and accuracy. This phenomenon was shown and researched in the voxel model's papers [5], [12], and [8]. Such representation has inherent qualities of point-wise representation, but changing voxel size provides flexibility in speed and domain robustness to the models. Its advantages have been explored in the SPVNAS [5], achieving both real-time inference speed and competitive accuracy. Another way to look at the voxels was proposed in the paper [6]. They adapted the idea of voxels to the task of LiDAR semantic segmentation by changing the shape of the grid from cubes to cylinder sectors. One of the recent papers [23] presents a multi-task model that bridges the performance gap between the multi-task and multiple single-task networks. The global context pooling module helps

extract global context features from LiDAR scans, there is a separate head for each task, so segmentation results are refined based on detection. To deal with sparse distant points, SphereFormer [9] introduced radial window self-attention that partitions the space into multiple non-overlapping narrow and long windows. It helps to overcome the disconnection issue and enlarges the receptive field of the model.

B. PROJECTIVE POINT CLOUD SEGMENTATION

Another way to use 3D points is to project them on the sphere. This method is organic due to the LiDAR mechanism, which launches lasers with fixed yaw and pitch angle steps. This type of projection was used in FIDNet [30], SalsaNext [7] and showed an outstanding performance in the sense of speed and quality tradeoff. On the one hand, this representation allows using 2D convolutional layers and image augmentations, maintaining high speed and knowledge from computer vision. On the other hand, the overall geometry can be lost since the 2D convolutions can find only two-dimensional patterns. Moreover, models, which use the projection as the source of the data, suffer from domain instability because projection operation is susceptible not only to the LiDAR placement and configuration but also to the robot geometry. Another approach that pushes this to the limits was presented in [17]. This model uses lightweight harmonic dense convolutions and an improved global contextual module to achieve high accuracy without drawbacks in inference time.

C. FUSION MODELS

Some models additionally use data from other sensors for 3D segmentation [1], [31], [32], [33], [34]. There are many reliable and fast ways to semantic segmentation of onboard camera images [4], [35]. Complementing the reliable, accurate, but sparse spatial information obtained from LiDAR with camera images that carry rich information about color and texture will expand the view of the surrounding space.

To benefit from LiDAR and camera data, early work has fused information from two sensors to improve the robustness and accuracy of the 3D segmentation algorithm [1], [31], [32], [33]. RGBAL [1] transforms RGB images to a polar-grid representation and outlines early and middle model fusion. PointPainting [33] obtains the segmentation results of images and projects them to the LiDAR space by spherical projection [36], which are subsequently concatenated with the original point cloud features. Zhuang et al. [13] proposed a model with two parallel streams for information from cameras and LiDAR. After the LiDAR point clouds are projected onto the image plane, the data is passed through the network, where the features obtained from the images are mixed with the sparse LiDAR features. Proposed perception-aware losses estimate the vast perceptual difference between LiDAR and camera modalities and boost the fusion of different perceptual information [13].

It is also worth mentioning 2DPASS [14], it additionally uses camera images only at the training stage. Using the

approach proposed in the article allows you to increase the learning of representations without any restrictions on the input data during validation.

D. DOMAIN ADAPTATION IN 3D SEGMENTATION

The domain adaptation problem among different LiDARs is an essential task that helps to use collected and synthesized data to operate in real-world situations without additional expenses. Currently, there are several different approaches to this issue. Models in [37] and [38] rely on adversarial loss that helps them to learn domain-agnostic embeddings. This method can be used for cross-modal tasks, but it limits the variety of available architectures. Another way to handle this problem is to use discrepancy loss which directly measures the distance between representations across domains. This approach was used in [39]. The model's embeddings are forced to be similar in learning this way. However, if the domain gap is big enough, such models can't keep up. The last approach is based on synthetic and semi-synthetic data [36]. The drawback of these methods is the need to solve another domain adaptation task or to make generated pipeline to be from the same domain. However, if generated properly, such data can be used to train any semantic segmentation model with little to no effort and without a drastic drop in quality.

The other way to address the gap between domains is adjusting ground truth data. Several recent papers explore those options. The authors of [40] propose choosing rectangular areas around selected points from domains and swapping them. Another approach to that issue from [41] introduces an encoder-decoder network for adaptation patterns of objects as a whole.

E. DATASETS FOR POINT CLOUD SEGMENTATION

There is a limited number of datasets for solving the problem of semantic segmentation of 360-degree LiDAR data in an urban environment. The key ones are given in Table 1. In it, we also provided statistics on the proportion of points belonging to the five joined categories among the total number of points in LiDAR scans. Their mapping is shown in Table 2. These are the most characteristic categories that can be further used when navigating and planning the movement of a robot or an unmanned vehicle.

One of the most popular datasets for segmentation methods evaluation is the large-scale SemanticKITTI dataset [15]. Similar KITTI-360 dataset [27] provides point-wise instance and semantic segmentation for 3D point clouds. The dataset consists of aligned 360 degrees 2D images and 3D point clouds. It can be used for benchmarking models on a wide variety of tasks including semantic scene understanding, novel view appearance synthesis, and many others. There are some other similar outdoor datasets collected from autonomous cars: SemanticPOSS [42] and Pandaset [43]. An important disadvantage of all these datasets is the small number of people in the near zone of the ego-vehicle at a distance of up to 10-15 meters.

TABLE 1. Modern datasets for projective semantic segmentation of 3D LiDAR point clouds.

Dataset	Resolution	Scans / Maps	LiDAR	Vehicle	Human	Surface	Static Obstacle	Unlabeled
Paris-Lille-3D [24], 2018	1024x32	17	Velodyne HDL-32E	5.91%	0.65%	56.32%	34.45%	3.30%
SemanticKITTI [15], 2019	2048x64	19,130	Velodyne HDL-64E	5.14%	0.019%	43.92 %	48.89%	2.02%
Toronto-3D [25], 2020	1024x32	4	32-lines Teledyne Optech Maverick	5.02%	N/A	53.44%	24.38%	3.94%
SemanticUSL [26], 2020	1024x64	1200	64-lines Ouster OS1	0.82%	0.12%	17.82%	20.81%	60.41%
RELLIS-3D [18], 2021	1024x32	13,556	64-lines Ouster OS1	N/A	0.81%	20.18%	54.54%	24.45%
KITTI-360 [27], 2021	2048x64	80,000	Velodyne HDL-64E	19.76%	10.04%	17.43%	51.87%	0.89%
NuScenes [28], 2022	1024x32	40,000	Velodyne HDL-32E	33.96%	0.20%	38.58%	27.1%	0.173%
Waymo Open Dataset [29], 2022	2650x64	29,667	Waymo Driver	8.92%	12.50%	2.04%	65.44%	11.08%
DAPS-1, 2023	1024x64	19,061	64-lines Ouster OS0	3.89%	0.46%	70.70%	24.89%	0.03%
DAPS-2, 2023	1024x64	109	64-lines Ouster OS0	0.04%	7.97%	47.96%	43.98%	0.05%

TABLE 2. Category mapping for different datasets.

Joint Categories	Dataset Categories
SemanticKITTI	
unlabeled	unlabeled, outlier, other-structure, other-object
vehicle	car, bicycle, bus, motorcycle, on-rails, truck, other-vehicle
human	person, bicyclist, motorcyclist
surface	road, parking, sidewalk, lane-marking, terrain, other-ground
obstacle (static)	building, fence, vegetation, trunk, pole, traffic-sign
RELLIS-3D	
unlabeled	unlabeled, other-object
vehicle	car
human	person
surface	road, terrain, other-ground
obstacle (static)	building, fence, vegetation, pole
DAPS-1, DAPS-2	
unlabeled	unlabeled
vehicle	vehicle
human	human
surface	surface
obstacle (static)	obstacle (static)

The problem of well-labeled urban Toronto-3D [25] and Paris-Lille-3D [24] datasets is that they do not contain separate LiDAR scans, but already combined semantic maps, which limits the possibility of their use for the real-time segmentation task.

Recently two large-scale open nuScenes [28], and Waymo [29] datasets released 3D semantic segmentation extensions. Both of them share the same label schema with detection datasets. Their features are the presence of only urban street scenes and the absence of park areas.

The relevant off-road datasets are SemanticUSL [26] and its updated version REllIS-3D [18]. They consist of fully-annotated LiDAR scans from off-road environments with a lot of scenes containing lawns, dirt roads, and various vegetation. At the same time, these datasets contain a relatively small number of labeled people and vehicles.

The limitations of existing datasets on the presence of people in the near zone and those recorded in park areas have modified us to collect and make publicly available our two datasets DAPS-1 and DAPS-2. Their more detailed description is given in Subsection III-D.

III. METHODOLOGY

A. TASK STATEMENT

In the article, we are addressing the domain adaptation problem for 3D semantic segmentation tasks using only one LiDAR sensor.

More formally, the proposed approach called DAPS3D takes raw $(N \times 3)$ point cloud P from LiDAR and returns (N) label vector C_P with semantic class for every point. A feature of the proposed approach is the study of neural network models M with parameters θ , which use the projection π of LiDAR point cloud onto a sphere. This projection makes it possible to use fast state-of-the-art neural networks designed for processing 2D images to recognize 3D data. Details of the various proposed M model architectures are given in Subsection III-B.

To ensure the domain adaptation of the model, we proposed original sets of augmentations A for the spherical projection, which are discussed in detail in Subsection III-C.

Thus, we can give a formula that simply describes the process of segmentation of a LiDAR point cloud:

$$C_P = M(A(\pi(P)), \theta). \quad (1)$$

Learning the optimal parameters θ of the model M is defined as the task of minimizing the loss function for the 2D image segmentation with known ground truth masks C_P^{gt} from the dataset:

$$L = \text{crossentropy}(M(A(\pi(P)), \theta), C_P^{gt}). \quad (2)$$

The proposed approach also includes the method to generate synthetic datasets for a domain adaptation problem, which is presented in Subsection III-D. For the comparison among different datasets, we decided to aggregate all classes into $N = 4$ ground truth labels: vehicles, humans, surfaces, and static obstacles.

An important aspect of this work is the study of the quality metric of different approaches on different data domains (see Subsection IV-C). Such a metric is Intersection over Union (IoU), generally recognized in the semantic segmentation tasks [7], [8], [20].

The developed model should also provide real-time performance. By real-time, we mean such a latency of the algorithm, which should be less than the period of receipt of

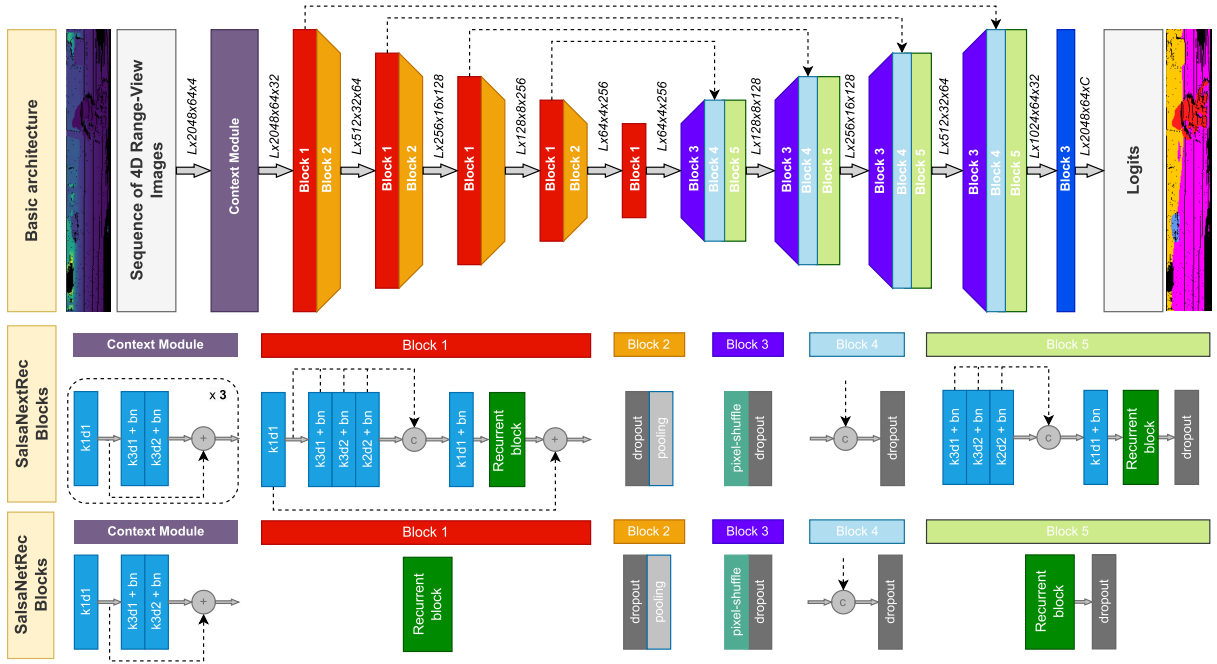


FIGURE 1. Proposed projective segmentation models SalsaNetRec and SalsaNextRec. The figure shows the basic architecture and blocks specific to different models.

sensory data. In our case, this value is equal to 100 ms (this is sufficient for the timely recognition of a three-dimensional scene for a slowly moving mobile ground robot).

B. PROJECTIVE NEURAL NETWORK MODEL

1) DATA REPRESENTATION

In our research, we are using spherical projection as our data representation for our models. The main benefits of such representation are real-time inference speed, coverage of augmentation methods and model architectures from computer vision, and state-of-the-art quality. The projection process creates an organic dense and compact representation without any information losses. To obtain such Range View (RV) image a special formula was used. Each 3D point (x, y, z) from the raw LiDAR point cloud is associated with another point (u, v) such as projection π :

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix} &= \pi(x, y, z, w, h) \\ &= \begin{pmatrix} \frac{1}{2}[1 - \arctan(y, x)\pi^{-1}]w \\ [1 - (\arcsin(z, r^{-1}) + f_{down})f^{-1}]h \end{pmatrix}, \end{aligned} \quad (3)$$

where h and w represents the height and width of the spherical projection, r denote the range of each point as $r = \sqrt{x^2 + y^2 + z^2}$, and f defines the sensor vertical field of view in the range $[f_{down}, f_{up}]$.

This formula with correct LiDAR parameters guarantees that every point from the 3D point cloud will have a corresponding pixel with the (x, y, z) vector stored in it. However, if we widen our vertical field of view, we will have

overlapping points, which will lead to a dense range image, but with some informational losses. The same artifact occurs when we choose h and w less than the grid parameters from the LiDAR. The important part is that even though we lose some information, the range image remains dense. We use this fact during the training process when using wider FoV and smaller projection sizes to match our target domain.

2) BASELINE MODELS

As a baseline, SalsaNet [2] and SalsaNext [7] were picked due to their performance and speed trade-off. These models are fast and convertible to TensorRT, which is crucial for our primal task. In our work, we use modifications of those models, which take only (x, y, z, r) vector instead of (x, y, z, r, i) , where r is the range index, and i is the LiDAR-dependent intensity value [7]. The reason behind that is that in the domain adaptation scenario, we do not want to use additional data, that can vary from dataset to dataset.

3) RECURRENT MODELS

Originally segmentation neural network takes range-view image obtained after spherical projection. This type of data representation allows the use of 2D convolutional recurrent blocks. In our DAPS3D approach, we propose SalsaNetRec and SalsaNextRec models (Figure 1) with two different types of recurrent approaches for different purposes (Figure 2).

Models, which we have named SalsaNetRecLSTM and SalsaNextRecLSTM, contain block 1 shown in Figure 2. This is an LSTM 2D block that was described in [46]. Its main goal is to convert the idea of memory from the original

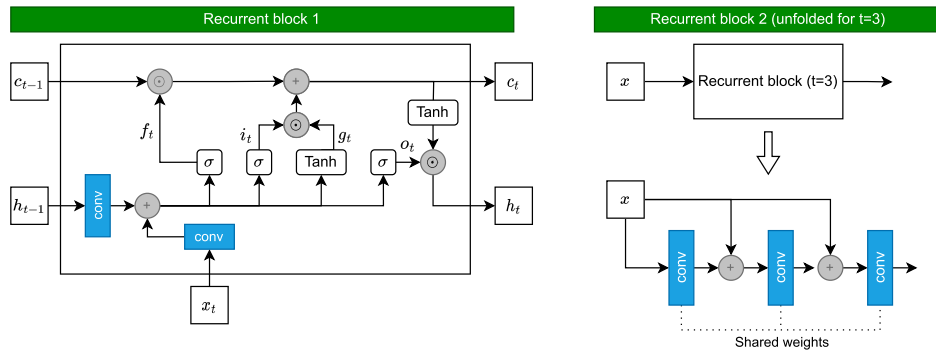


FIGURE 2. Recurrent blocks under study. SalsaNetReclSTM and SalsaNextReclSTM use recurrent block 1. SalsaNetRec contains recurrent block 2.

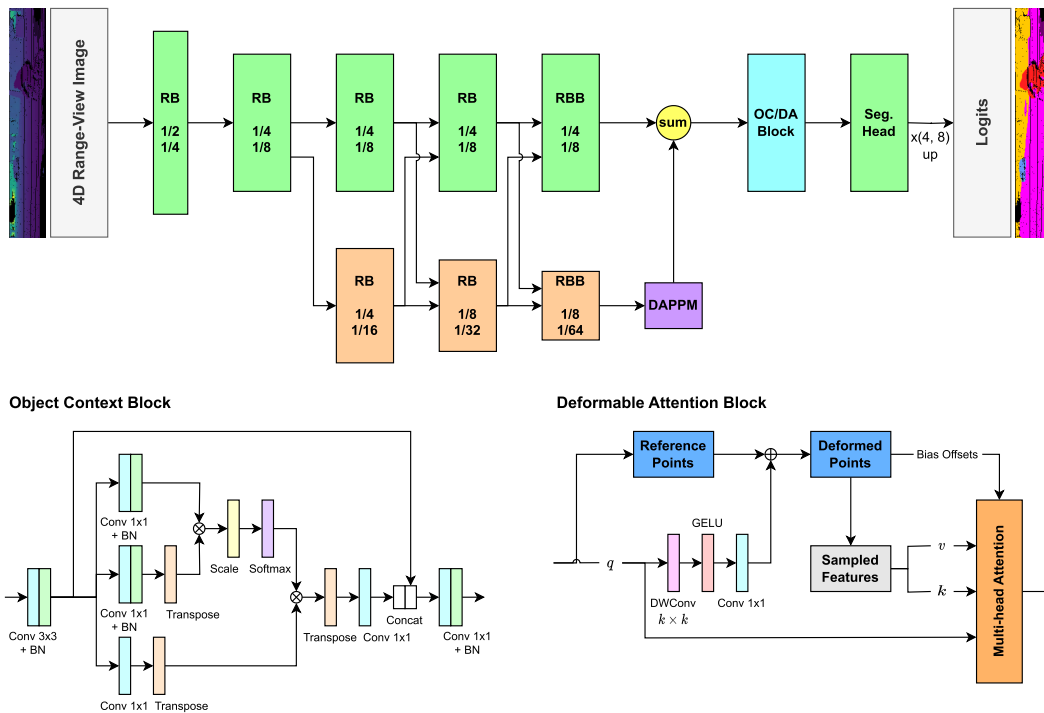


FIGURE 3. Modified DDRNet architecture. “RB” denotes sequential residual basic blocks. “RBB” denotes the single residual bottleneck block. “DAPP” denotes the Deep Aggregation Pyramid Pooling Module [19]. “OC” and “DA” denote Object Context Block [44] and Deformable Attention Block [45] respectively.

LSTM block to sequences of image data. The recurrent block 2 (Figure 2) is used directly in the architecture called SalsaNetRec. It was proposed in [47] and aims to leverage the recurrent architecture to enhance feature representation for segmentation tasks without changing the number of parameters.

4) ATTENTION- AND TRANSFORMER-BASED MODELS

We have also considered the adaptation of 2D segmentation models to perform a 3D segmentation task. DDRNet [19] and SegFormer [20] models were chosen according to quality and speed trade-off. Due to the specifics of spherical projections of LiDAR clouds, it was decided to adjust the network architecture to narrow elongated images by changing the size of the

convolution kernels and their stride. The difference lies in the increase in the resolution of feature maps along the vertical axis, as shown in Figure 3.

For SegFormer we have changed kernel size and stride for the first transformer block, which led to an increase in the size of subsequent feature maps. SegFormer architecture with feature map shapes is shown in Figure 4.

We have also tested the effects of different attention mechanisms. We picked the Object Context module [44] and more recent Deformable Attention module [45] and inserted them into the architecture of the DDRNet model, as shown in Figure 3. The choice of the Object Context module is due to its ability to improve the quality of fast image semantic segmentation models, for example, as shown in [35].

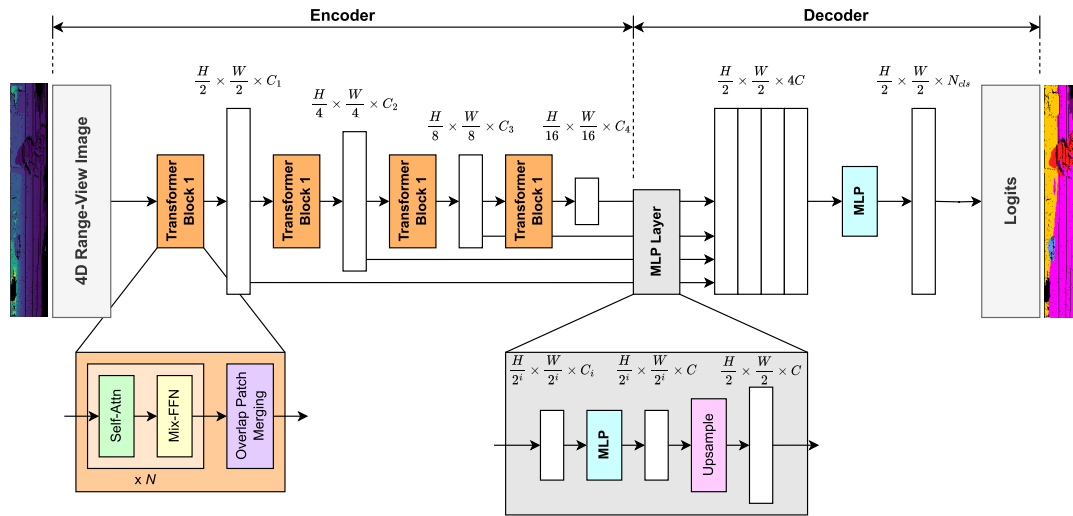


FIGURE 4. SegFormer architecture with feature maps shapes.

The Deformable Attention module is well suited for effectively expanding the receptive field of neural networks [45]. The DAPPM (Deep Aggregation Pyramid Pooling Module) block shown in Figure 3 has an architecture similar to the original model [19] and allows taking into account information from multi-scale feature maps when segmenting the projection of a point cloud.

5) EVALUATION METRICS

The common Intersection over Union (IoU) is used as a basic quality metric to evaluate the results of the proposed approaches. Due to practical usefulness and simplicity when testing on different datasets, the semantic categories in them were previously grouped into four main classes: “vehicle” (including cars, trucks, buses, bicycles, motorcycles, and other vehicles), “human” (including pedestrians, bicyclists, and motorcyclists), “surface” (including road, parking, road markings, sidewalks, terrain, i.e. surfaces where the ground mobile robot can pass), and “obstacle (static)” (including buildings, fences, poles, vegetation, etc.).

C. AUGMENTATIONS

In this work, we have developed geometry-preserving augmentations for 3D point clouds and their projections to improve the quality of the projective semantic segmentation model (see Figure 5). The main goal was to introduce a variety of data to the model during the training process to achieve better domain stability. Every augmentation aims to cover a specific problem that causes artifacts during the inference.

1) FoV AUGMENTATION

One of the major problems in utilizing pre-trained models is the difference in the field of view. Projective models are sensitive to this parameter of LiDARs. To overcome this problem, We propose a Field of View augmentation that

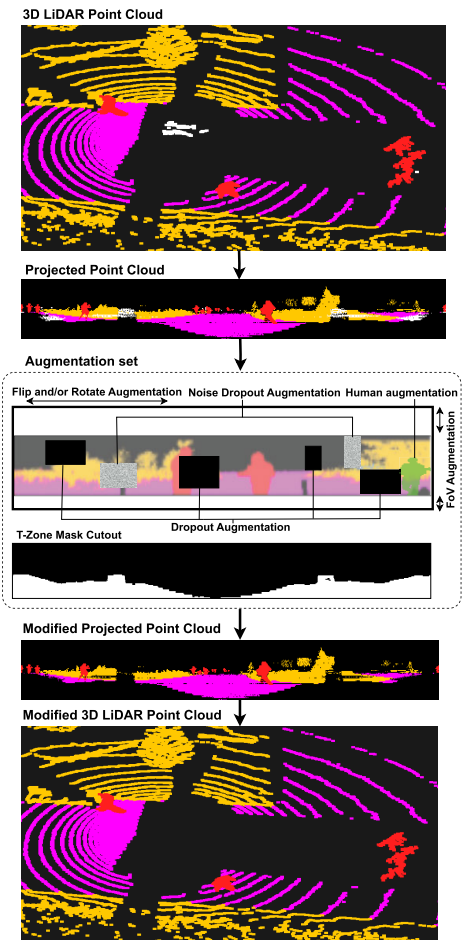


FIGURE 5. Visualization of the proposed augmentation process.

randomly restricts or widens the projection segment in the data preparation pipeline by modifying f_{down} and f_{up} , causing zero strips on the final projection.

2) HEIGHT AUGMENTATION

Another significant challenge is the variation in LiDAR height. This problem is caused by the difference in the LiDAR setups. It drastically changes the point of view on different objects, making it impossible for the model to recognize. To lower this gap height augmentation was developed. In this augmentation, the whole point cloud shifts vertically on the same value. To achieve a realistic scene, the algorithm first finds the ground height and then shifts all points on the random value.

3) FLIP AND ROTATION AUGMENTATION

We decided to apply flip and rotation augmentation to the projection to overcome the bias of LiDAR's rotation angle. This deformation can be interpreted as rotation and flipping of the LiDAR sensor and can help in the cases of non-road environments. It is also natural to use because both flip and rotate preserve the geometry.

4) MASK NOISING AND DROPOUT AUGMENTATION

Mask noising and dropout are other regularization techniques that are borrowed from computer vision. They are applied after the projection process and can be interpreted as an occlusion and sensor noising of the areas. This method helps prevent overfitting on commonly-shaped objects and allows the model to encounter "zero" patches that may be created during field-of-view augmentation.

5) HUMAN AUGMENTATION

To address differences in class balance between the two datasets, We developed a human augmentation. During the PointCloud augmentation phase, random samples from the library were added to the scene. The library of the samples was collected from the NuScenes dataset via their API. Specifically, We extracted the bounding boxes of people with corresponding instance ids based on NuScenes labels. To ensure high-quality samples, we filtered out noise by only retaining samples with high density. Similar to the height augmentation approach, we calculated the ground height and placed the humans at the corresponding height to maintain realism in the scene.

6) T-ZONE AUGMENTATION

In real-life scenarios, the configurations of robots used various experiments can differ significantly from the configuration of the robot used for dataset recording. Some points of the point cloud can fall on the body of the robot or be covered by various sensors. To simulate the presence of our robot's body we employ T-zone augmentation. It cuts out from the projection of the LiDAR point cloud those points that in real life would fall on its body, as it is shown in Figure 5.

D. DEVELOPMENT OF DATASETS

1) MOTIVATION

In our research, we explored the possibility of adopting a SalsaNext-like [7] solution for small autonomous robots that

utilize a different LiDAR sensor configuration. However, we encountered a challenge due to a data domain shift. The original SalsaNext paper [2] and available open-source research datasets mainly focused on self-driving cars, which did not match our specific LiDAR setup. Our LiDAR sensors have a distinct beam angle, and different number of beams compared to SalsaNext's objective dataset. Consequently, the existing pre-trained models designed for projective segmentation methods were unsuitable for our desired data domain. Addressing this issue was neither straightforward nor cost-effective. Hence, we designed a highly efficient alternative approach.

It leverages the SemanticKitti dataset [15] as a foundation. To simulate the LiDAR data captured by a small robot with a different LiDAR configuration, we created a synthetic dataset called DAPS1. This synthetic dataset mirrored SemanticKitti but emulated the data as if it were obtained from a small robot with our desired LiDAR configuration. The creation of DAPS1 involved two distinct steps:

- **Mesh Map Generation:** Firstly, we constructed a detailed mesh map of the scene. This process involved representing the 3D structure of the environment using interconnected triangles. By creating an accurate mesh map, we were able to effectively model the scene and its objects.
- **Virtual Lidar Sensor Simulation:** Once the mesh map was in place, we simulated a virtual LiDAR sensor that precisely matched the lidar configuration of our small robot. This virtual LiDAR sensor was employed to generate data that closely resembled the LiDAR readings captured by our desired setup.

By creating DAPS1, we successfully prepared a synthetic dataset that emulated the LiDAR data collected by our small robot with a different LiDAR configuration. This synthetic dataset enabled us to pre-train the SalsaNext architecture and reduce the size of the subsequent DAPS2 dataset required for fine-tuning.

2) SemanticKITTI

First of all, we used the popular SemanticKITTI dataset [15]. We use the same training, validation, and test split as described in the original paper for SalsaNext [7]. Specifically, we used sequence as the validation set, and sequences from 00 to 10 were chosen as a training set. The dataset has 22 classes which we mapped into our five cross-domain classes for unified evaluation as it is shown in Table 2. This dataset was selected due to its size and variety of road scenes.

3) EXTENDED RELIS-3D

The main reason for selecting RELIS-3D dataset [18] was its size and unique off-road environment. We split the dataset into training and validation sets, taking the first three sequences as our training data.

During the training process on the RELIS-3D, we encountered that some of the points belonging to the

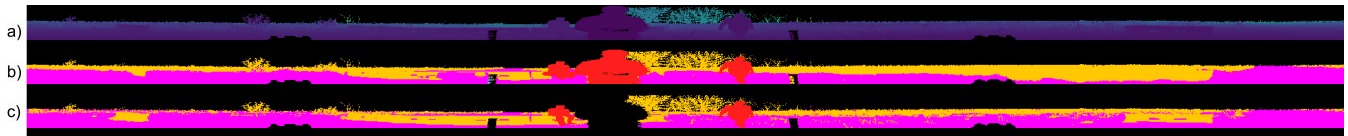


FIGURE 6. Example of inaccurate labels in the RELLIS-3D dataset. a) Range-view image; b) Model inference after 80 epochs. c) Ground truth.

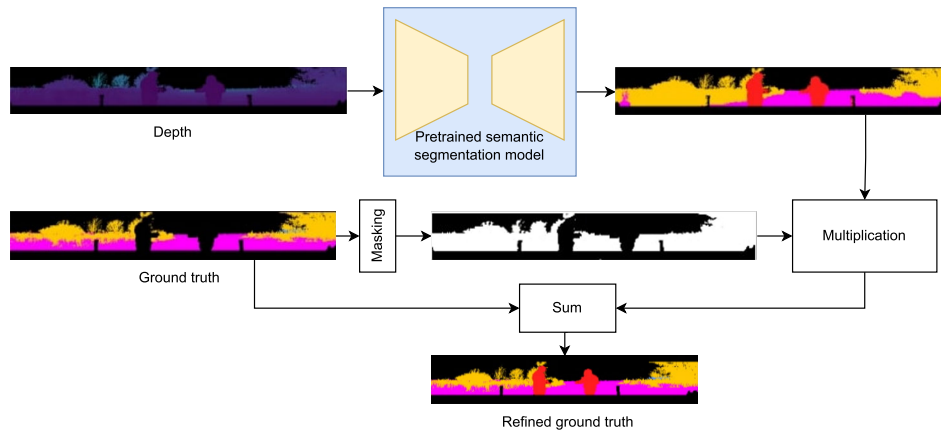


FIGURE 7. Labeling enhancing process.

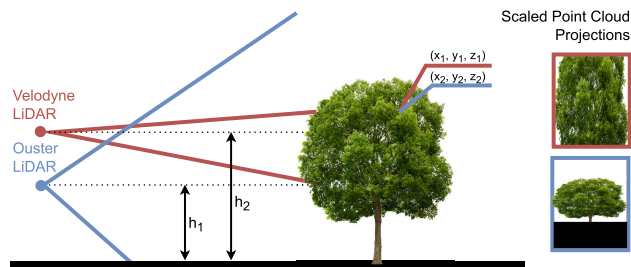


FIGURE 8. Domain gap between scaled point cloud projections from different LiDARs. Orange vertical field of view (FOV) visualization is shown for Velodyne LiDAR used in SemanticKITTI. Blue FOV is shown for Ouster LiDAR used in our DAPS datasets.

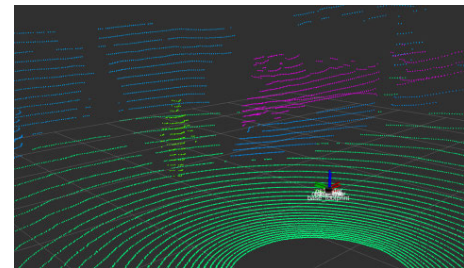


FIGURE 9. Example of semi-synthetic DAPS-1 dataset.

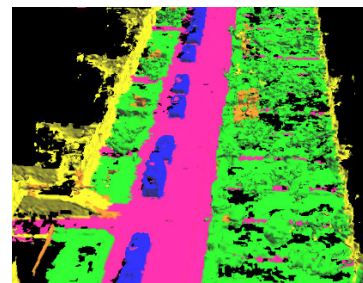


FIGURE 10. Example 3D semantic mesh reconstructed by Kimera-Semantic approach [48].

“human” class in the close zone were not labeled (Figure 6). This problem had a significant impact on the model evaluation because such points map to the empty class that did not affect training, even though the information about the points comes correct. The model, however, has begun to accurately label such points after 80 epochs of training on the SemanticKITTI and RELLIS-3D datasets.

We then used this trained model to refine the labels. We segmented the original point cloud and assigned the “human” class to the points that the model attributed to “human”, and which in the original markup were labeled as an empty class. Our proposed labeling enhancement process is shown in Figure 7.

4) DAPS-1

The natural approach to overcome the domain gap (see Figure 8) is to capture or synthesize realistic data from a similar domain. This idea was implemented in our

semi-synthetic DAPS-1 dataset (see Figure 9), which was generated based on the SemanticKITTI [15]. Its main advantage is that the process can be reproduced for any setup with only a 3D model of the robot and LiDAR parameters. The pipeline consists of 5 steps. First, the data was converted to the 3D semantic mesh using the Kimera-Semantic approach [48] as it is shown in Figure 10. Then the layers of the map are separated in the open-source program Mesh-Lab [49]. It is important, that Kimera-Semantic works only with static

TABLE 3. Augmentation set configurations. The letter “T” in the name of the augmentation sets means the use of T-zone augmentation.

Configuration	FoV aug	human aug	height aug	xy flip aug	xy rotate	dropout aug	noise aug	T-Zone
No augmentation	False	False	False	False	False	False	False	False
Augmentation set 1	True	True	$h \in [0.3, 1.8]$	True	True	False	False	False
Augmentation set 2	True	True	$h \in [0.3, 1.8]$	True	True	True	False	False
Augmentation set 3	True	True	$h \in [0.3, 1.8]$	True	True	False	True	False
Augmentation set 4	True	False	$h \in [0.3, 1.8]$	True	True	True	False	False
Augmentation set 5	True	False	False	True	True	False	False	False
Augmentation set 5T	True	False	False	True	True	False	False	True
Augmentation set 6	True	True	False	True	True	False	True	False
Augmentation set 7	True	True	False	True	True	True	False	False
Augmentation set 7T	True	True	False	True	True	True	False	True
Augmentation set 8	True	False	$h \in [0.3, 1.8]$	True	True	False	True	False
Augmentation set 9	True	False	$h \in [0.3, 1.8]$	True	True	True	False	False
Augmentation set 10	True	True	False	True	True	True	True	False

**FIGURE 11.** Cleaning robot used in DAPS-2 dataset recording process.

objects, that is why We manually added some human models, that were generated in the MakeHuman [50] program, on the track. The next step is to simulate in Gazebo [51] with the semantic layers and desired LiDAR configuration with the open-source plugin. Finally, we reproduce the original locations of the robot and capture the data to the rosbag archives.

5) DAPS-2

This dataset was recorded during a real field trip of the cleaning robot (see Figure 11) to the territory of the VDNH Park in Moscow in the summer of 2021. The robot model contained a configuration of 3 LiDARs (central and 2 side ones). DAPS-2 contains several robot scenes in different parts of the park with different pedestrian fillings, with all points from the main central LiDAR.

6) CATEGORY MAPPING

In our work, we combined the main semantic categories for 3D scene elements: “vehicle”, “human”, “surface”, and “obstacle (static)”. We did not include the “unlabeled” category in our models. Table 2 shows category mapping between our five cross-domain classes and labels from different datasets.

IV. EXPERIMENTAL RESULTS

A. MODEL TRAINING

We implemented our models using PyTorch and performed training and evaluation on an NVIDIA GeForce RTX 2080 Ti GPU. For the SalsaNet- and SalsaNext-based

models, we trained them from scratch with randomly initialized weights. During training, we applied various data augmentations configurations, which are specified in Table 3. We do not crop projected point clouds and set batch size to 8 on all datasets. We trained the models using an SGD optimizer for 40 epochs, the first of which was a warm-up one. The momentum for SGD was set to 0.9, the learning rate to an initial value of 0.05, and the learning rate decay with a factor of 0.99.

For SegFormer we utilized the MiT-B1 encoder pre-trained on the Imagenet-1K dataset and randomly initialize the decoder. We set the batch size to 6 and the learning rate to an initial value of 0.001. All other settings remained the same as the configuration for the SalsaNet-based models.

The DDRNet-based models share the same configuration as the SalsaNet-based models, except for the number of epochs and a learning rate scheduler. We trained the models using the SGD optimizer for 140 epochs, the first of which was a warm-up one. The momentum for SGD was set to 0.9, the learning rate to an initial value of 0.05, and the learning rate decay with a factor of 0.99 for the first 40 epochs and a factor of 0.96 for the rest.

B. INFLUENCE OF DATA AUGMENTATION

The results of the models on the SemanticKITTI+RELLIS-3D dataset are reported in Table 4. The parameters of the augmentations are shown in Table 3. Our proposed Augmentation set 5 enhances the quality of the baseline model on 4.3% in mIoU and 8.5% in the human class, which is one of the most important during inference and underrepresented classes. This indicates that we can effectively regularize our model with augmentations during the training phase to fix the class balance problem without any drops in the quality. From the table, it can be observed that the second augmentation set is the best in the dynamic classes, while the fourth set is the best in the static classes. The reason behind that is that human augmentation brings a lot of new information about dynamic objects, even though the data was noisy, because the dataset, that was used to create a library, had drastically different LiDAR parameters. The results also demonstrate that We can modify and train sequential models within our framework.

TABLE 4. Results of the SalsaNet baseline model on SemanticKITTI and RELLIS-3D with different input data augmentation.

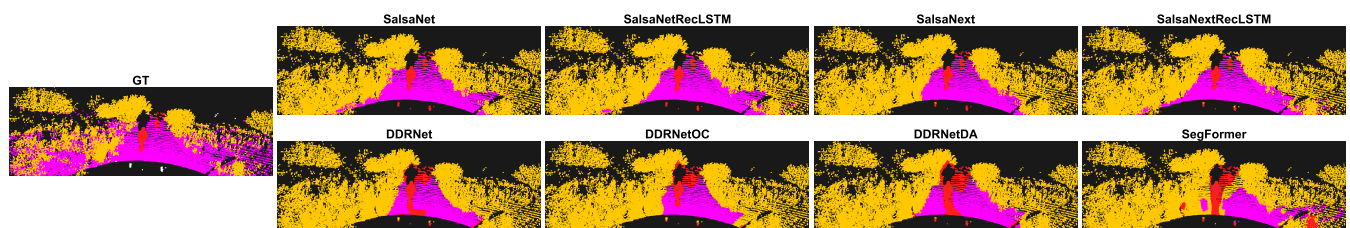
Model	IoU avg	IoU [vehicle]	IoU [human]	IoU [surface]	IoU [obstacle]
No augmentation	0.799	0.907	0.794	0.754	0.742
Augmentation set 1	0.798	0.866	0.848	0.751	0.728
Augmentation set 2	0.791	0.881	0.827	0.737	0.717
Augmentation set 3	0.782	0.841	0.852	0.724	0.711
Augmentation set 4	0.809	0.874	0.852	0.768	0.744
Augmentation set 5	0.842	0.913	<u>0.881</u>	0.720	0.713
Augmentation set 5T	<u>0.832</u>	0.869	0.886	0.789	0.782
Augmentation set 6	0.815	<u>0.907</u>	0.849	0.747	0.759
Augmentation set 7	0.820	0.905	0.874	0.746	0.757
Augmentation set 7T	0.811	0.894	0.850	0.744	0.755
Augmentation set 8	0.809	0.862	0.873	0.749	0.753
Augmentation set 9	0.816	0.829	0.886	<u>0.777</u>	<u>0.771</u>
Augmentation set 10	0.810	0.886	0.837	0.755	0.761

TABLE 5. Comparison of the segmentation models trained and evaluated on the SemanticKITTI dataset.

Model	IoU avg	IoU [vehicle]	IoU [human]	IoU [surface]	IoU [obstacle]
SalsaNet	0.787	0.882	0.412	0.929	0.924
SalsaNetRec	0.789	0.855	0.488	0.913	0.900
SalsaNetRecLSTM	0.751	0.887	0.271	<u>0.927</u>	<u>0.920</u>
SalsaNext	<u>0.821</u>	<u>0.907</u>	<u>0.564</u>	0.905	0.907
SalsaNextRecLSTM	0.835	0.914	0.600	0.911	0.914
DDRNet	0.692	0.750	0.225	0.901	0.893
DDRNetOC	0.687	0.739	0.222	0.900	0.889
DDRNetDA	0.696	0.754	0.232	0.903	0.895
SegFormer	0.539	0.437	0.048	0.893	0.777

TABLE 6. Comparison of the segmentation models trained and evaluated on SemanticKITTI and RELLIS-3D.

Model	IoU avg	IoU [vehicle]	IoU [human]	IoU [surface]	IoU [obstacle]
SalsaNet	<u>0.832</u>	0.869	0.886	0.789	0.782
SalsaNetRec	0.808	0.858	0.847	0.763	0.766
SalsaNetRecLSTM	0.828	0.889	0.880	<u>0.770</u>	<u>0.774</u>
SalsaNext	<u>0.832</u>	<u>0.904</u>	<u>0.906</u>	0.755	0.763
SalsaNextRecLSTM	0.833	0.908	0.916	0.748	0.760
DDRNet	0.706	0.759	0.654	0.685	0.725
DDRNetOC	0.705	0.749	0.652	0.693	0.728
DDRNetDA	0.658	0.715	0.471	0.716	0.731
SegFormer	0.533	0.448	0.423	0.616	0.643

**FIGURE 12.** Example of 3D semantic segmentation with different models trained and evaluated on SemanticKITTI extended with RELLIS-3D dataset. Here and in the following figures, we show the effect of adding the highest quality LSTM-based recurrent block to SalsaNet and SalsaNext-based architectures. We also show the results of lower quality models based on the DDRNet and Segformer architectures.

The recursive modifications of SalsaNet, trained in the same pipeline, improve the quality of the semantic segmentation. The effectiveness of the recurrent blocks is also evident, as they achieve similar quality to the best augmentation sets when using only the most neutral augmentations.

C. RESULTS ON OPEN DATASETS

1) EXPERIMENTS ON SemanticKITTI

Table 5 shows the segmentation results of four joint classes “vehicle”, “human”, “surface”, and “obstacle” on LiDAR

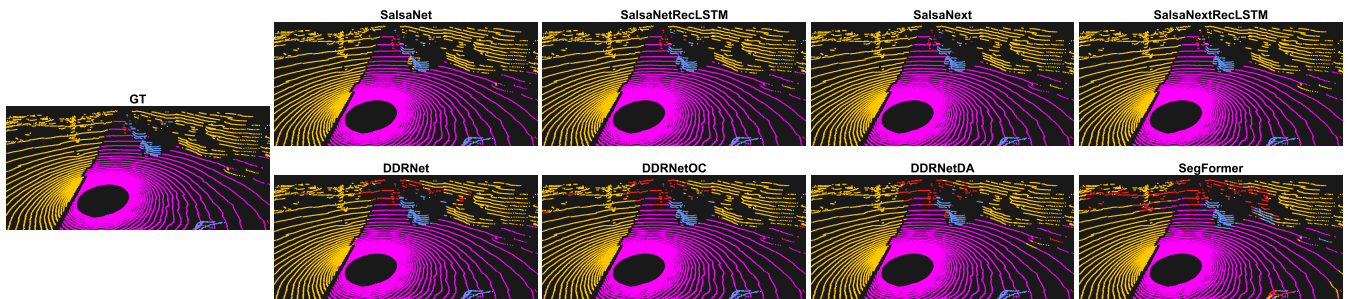
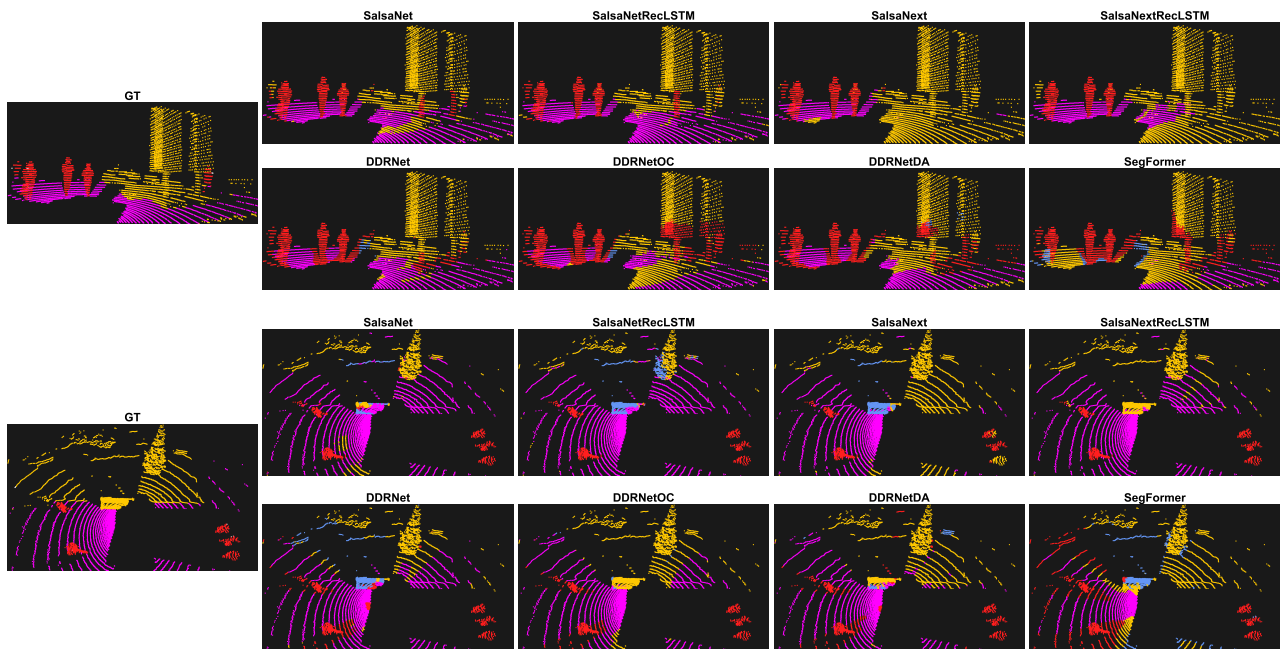
data from SemanticKITTI [15]. All models were trained using Augmentation set 5. SalsaNextRecLSTM shows the best overall metrics and demonstrated significant improvement compared to the basic SalsaNext model.

2) EXPERIMENTS ON SemanticKITTI EXTENDED WITH RELLIS-3D

Table 6 presents the segmentation results on LiDAR data from SemanticKITTI [15] extended with RELLIS-3D [18]. The expansion was performed by grouping the categories

TABLE 7. Comparison of the segmentation models trained and evaluated on the DAPS-1 dataset.

Model	IoU avg	IoU [vehicle]	IoU [human]	IoU [surface]	IoU [obstacle]
SalsaNet	0.867	0.880	0.646	0.989	0.953
SalsaNetRec	0.850	0.836	0.680	0.974	0.908
SalsaNetRecLSTM	0.862	0.878	0.632	0.988	0.949
SalsaNext	0.886	0.878	0.721	0.990	0.954
SalsaNextRecLSTM	0.932	0.929	0.830	0.994	0.974
DDRNet	0.690	0.773	0.126	0.977	0.886
DDRNetOC	0.694	0.769	0.138	0.978	0.889
DDRNetDA	0.691	0.770	0.129	0.977	0.886
Segformer	0.530	0.495	0.052	0.909	0.665

**FIGURE 13.** Example of 3D semantic segmentation with different models trained and evaluated on the DAPS-1 dataset.**FIGURE 14.** Comparison of 3D semantic segmentation results on the DAPS-2 dataset for models trained on the DAPS-1 dataset.

from both datasets into four categories. All models are trained with configuration including Augmentation set 5 and T-Zone cutout. The SalsaNextRecLSTM demonstrates the best result on SemanticKITTI extended with RELLIS-3D. The SalsaNet and SalsaNext approaches demonstrate the second-best results after it. The DDRNetDA is inferior in the “human” segmentation compared to DDRNetOC and the basic DDRNet.

Figure 12 displays the visualized results of SalsaNet-based models on RELLIS-3D. Both recurrent models and SalsaNext

outperform SalsaNet in segmenting “human” legs. However, DDRNet models have artifacts in the “human” category and detect fewer instances of “surface”.

3) EXPERIMENTS ON DAPS-1

Segmentation results on the DAPS-1 synthetic dataset are shown in Table 7. All models are trained with Augmentation set 5 and T-Zone cutout. Models with recurrent blocks are inferior in terms of segmentation quality to the SegFormer by 0.5–1%. DDRNet and SegFormer models notably

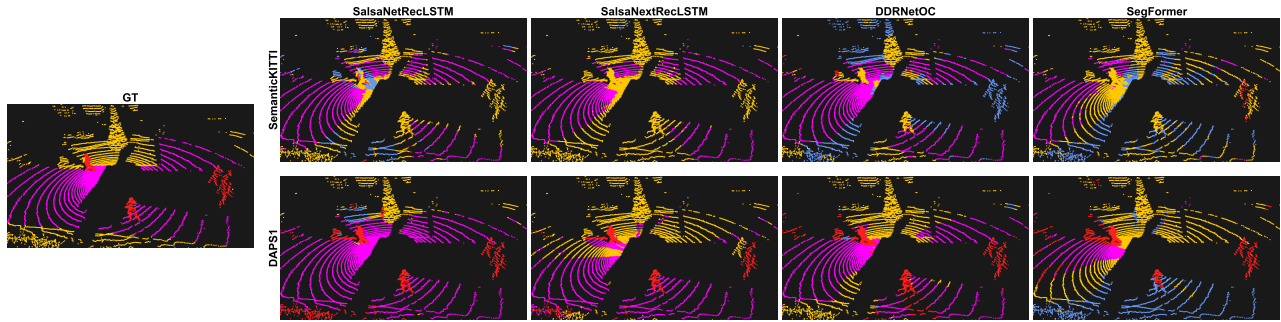


FIGURE 15. Comparison of 3D semantic segmentation results on the DAPS-2 dataset for models trained on SemanticKITTI with augmentation set 5 (first line) and DAPS-1 with augmentation set 5 + T-Zone cutout (second line).

TABLE 8. Comparison of the segmentation models trained and evaluated on the DAPS-2 dataset.

Model	IoU avg	IoU [human]	IoU [surface]	IoU [obstacle]
Trained on SemanticKITTI				
SalsaNet	0.405	0.056	0.662	0.498
SalsaNetRec	0.267	0.034	0.478	0.290
SalsaNetRecLSTM	0.338	0.075	0.565	0.373
SalsaNext	0.258	0.074	0.423	0.277
SalsaNextRecLSTM	0.327	0.039	0.548	<u>0.394</u>
DDRNet	0.345	0.080	0.768	0.187
DDRNetOC	0.323	0.015	<u>0.739</u>	0.216
DDRNetDA	<u>0.369</u>	0.040	0.705	0.361
Segformer	0.230	0.166	0.182	0.343
Trained on SemanticKITTI + RELIS-3D				
SalsaNet	<u>0.712</u>	0.733	0.760	0.643
SalsaNetRec	0.481	0.446	0.590	0.406
SalsaNetRecLSTM	0.689	0.770	<u>0.761</u>	0.537
SalsaNext	0.663	<u>0.799</u>	0.684	0.505
SalsaNextRecLSTM	0.736	0.813	0.784	0.612
DDRNet	0.580	0.697	0.535	0.509
DDRNetOC	0.616	0.724	0.586	0.539
DDRNetDA	0.541	0.614	0.507	0.503
Segformer	0.290	0.412	0.054	0.404
Trained on DAPS-1				
SalsaNet	0.575	0.479	0.741	0.504
SalsaNetRec	0.609	0.596	<u>0.748</u>	0.482
SalsaNetRecLSTM	0.624	0.532	0.783	0.557
SalsaNext	<u>0.643</u>	<u>0.706</u>	0.620	<u>0.602</u>
SalsaNextRecLSTM	0.759	0.709	0.746	0.823
DDRNet	0.516	0.354	0.737	0.458
DDRNetOC	0.562	0.403	0.705	0.577
DDRNetDA	0.531	0.360	0.718	0.515
Segformer	0.393	0.355	0.418	0.407

struggle in the “human” category because, due to the low resolution, they cannot distinguish “human” from its surroundings.

Figure 13 showcases the visualized results of neural network models on the DAPS-1 dataset. All models encounter challenges segmenting the distant “vehicle” and the result is noisy, but to the presence of a nearby “human”. The SalsaNetRec model mistakenly recognized the “surface” to the left and the SalsaNet labeled parts of the first “vehicle” as “obstacle”.

The proposed SalsaNextRecLSTM model turned out to be the most qualitative method, surpassing the others both in terms of metrics and in visualizing the results.

TABLE 9. Parameter number and latency comparison using hardware with RTX2080Ti GPU.

Model	Params, M	Latency, ms
SalsaNet	23.3	14.35
SalsaNetRec	23.3	20.96
SalsaNetRecLSTM	108	24.11
SalsaNext	6.71	19.69
SalsaNetRecLSTM	25.1	24.85
DDRNet	5.73	18.63
DDRNetOC	5.96	18.89
DDRNetDA	6.59	30.32
Segformer	13.7	21.38

4) EXPERIMENTS ON DAPS-2

A comparison of the quality metrics of semantic segmentation on the dataset from the DAPS-2 park area is shown in Table 8. All models are trained with Augmentation set 5. Models trained on SemanticKITTI + RELIS-3D and DAPS-1 additionally use T-Zone augmentation. Models trained on SemanticKITTI show low quality metrics due to different data domains and a small amount of green park zones in the training set. SalsaNextRecLSTM significantly outperformed other models on SemanticKITTI + RELIS-3D, DAPS-1, and DAPS-2. It should be noted that the use of the LSTM-based recurrent block gives an increase in quality for both the SalsaNet and the SalsaNext baselines.

IoU for the “surface” category of SegFormer trained on SemanticKITTI + RELIS-3D is an outlier, which might be caused by a large number of “human” instances. Trained on DAPS-1 models with recurrent blocks outperform basic SalsaNet by 3.4–4.9% mIoU.

Figure 14 shows the visualized results of various models on DAPS-2. All listed models are trained on DAPS-1 with Augmentation set 5 and T-Zone cutout. SalsaNextRecLSTM outperforms other models in the “human” and “obstacle” classes and shows one of the best results for the “surface” category. DDRNetOC result is less noisy compared to other models, and it did not label bench as a “vehicle”. It is worth noting that due to the low resolution, the DDRNet and SegFormer models often mark objects around “human” with an inappropriate class.

The importance of domain adaptation is demonstrated in Figure 15. The creation of a DAPS-1 synthetic dataset with a real 3D model of the robot and LiDAR parameters is crucial

for real-life applications. All listed models demonstrate significantly better results, especially in the “human” category. Domain adaptation boosted SalsaNetRecLSTM quality metric by more than 43% and 23% for DDRNetOC.

All our models have a small number of trainable parameters, except SalsaNetRecLSTM, and provide real-time point cloud segmentation on RTX2080Ti GPU as shown in Table 9.

V. CONCLUSION

In our article, we extensively explored the projective segmentation of three-dimensional LiDAR point clouds and investigated the domain adaptive approach called DAPS3D.

We observed that existing open datasets lack scenes captured from mobile robot LiDARs in parks or on roads with a significant number of people nearby. To address this limitation, we proposed an approach to augment existing datasets by including “human” objects in the near-field of a vehicle. We created the semi-synthetic DAPS-1 dataset using this approach. Additionally, we collected the DAPS-2 dataset in a park area using a real robot. These datasets showcased the potential of domain transfer when training different state-of-the-art deep neural networks on other open datasets.

The paper introduces enhancements to the existing basic neural network methods for projective segmentation, focusing on the fast SalsaNet model with the incorporation of various types of recurrent blocks. The SalsaNextRecLSTM model, which we developed, achieved the best performance indicators when evaluated on the proposed DAPS-1 and DAPS-2 datasets. For it, the integration of recurrent blocks provided a significant increase in the mIoU metric by 1.4% for SemanticKitti dataset, by 4.6% for DAPS-1 and by 6.9-11.6% for DAPS-2 compared to the basic state-of-the-art projective model SalsaNext.

We also explored the application of convolutional models for 3D segmentation using the real-time DDRNet architecture with different attention modules and the SegFormer transformer method, which we adapted for LiDAR spherical projections. However, their use did not allow them to surpass SalsaNet- and SalsaNext-based approaches in terms of quality and performance.

We proposed and demonstrated the effectiveness of various augmentation techniques. Among them, the combination of field of view, flip and rotate augmentations, along with the T-Zone cutout approach, yielded the best results in training neural network models on open datasets. These augmentations ensured the stability of projective neural network segmentation methods when encountering variations in the data domain, including the use of different LiDAR devices.

In our experiments evaluating the proposed and investigated approaches, we found that PyTorch implementations of various methods exhibited latencies ranging from 14.4 to 30.2 ms when tested on the widely used RTX2080Ti GPU. This performance highlights the potential for model acceleration or their direct utilization in onboard perception systems of autonomous vehicles.

In our future work, we aim to significantly increase the volume of the DAPS-2 dataset with other park zones. It is promising to study the approaches of spatiotemporal feature aggregation [52] on data sequences along with recurrent blocks as part of models for point cloud projective segmentation. We also plan to explore the adaptation of the proposed neural network models for use on edge devices.

ACKNOWLEDGMENT

The authors would like to thank LLC Integrant for providing materials for the development of DAPS-1 and DAPS-2 datasets.

REFERENCES

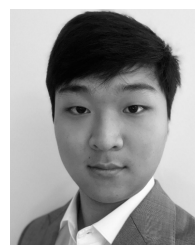
- [1] K. E. Madawi, H. Rashed, A. E. Sallab, O. Nasr, H. Kamel, and S. Yogamani, “RGB and LiDAR fusion based 3D semantic segmentation for autonomous driving,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 7–12.
- [2] E. E. Aksoy, S. Baci, and S. Cavdar, “SalsaNet: Fast road and vehicle segmentation in LiDAR point clouds for autonomous driving,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 926–932.
- [3] J. Jeong, H. Song, J. Park, P. Resende, B. Bradai, and K. Jo, “Fast and lite point cloud semantic segmentation for autonomous driving utilizing LiDAR synthetic training data,” *IEEE Access*, vol. 10, pp. 78899–78909, 2022.
- [4] S. Abdgapporov, S. Miraliev, V. Kakani, and H. Kim, “Joint multiclass object detection and semantic segmentation for autonomous driving,” *IEEE Access*, vol. 11, pp. 37637–37649, 2023.
- [5] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, “Searching efficient 3D architectures with sparse point-voxel convolution,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 685–702.
- [6] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, “Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9934–9943.
- [7] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, “SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds,” in *Proc. Int. Symp. Vis. Comput.* Cham, Switzerland: Springer, 2020, pp. 207–222.
- [8] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, “RPVNet: A deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16004–16013.
- [9] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, “Spherical transformer for LiDAR-based 3D recognition,” 2023, *arXiv:2303.12766*.
- [10] R. Cheng, R. Razani, Y. Ren, and L. Bingbing, “S3Net: 3D LiDAR sparse semantic segmentation network,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14040–14046.
- [11] G. Qian, Y. Li, H. Peng, J. Mai, H. A. A. K. Hammoud, M. Elhoseiny, and B. Ghanem, “PointNext: Revisiting PointNet++ with improved training and scaling strategies,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 23192–23204.
- [12] X. Roynard, J.-E. Deschaud, and F. Goulette, “Classification of point cloud scenes with multiscale voxel deep network,” 2018, *arXiv:1804.03583*.
- [13] Z. Zhuang, R. Li, K. Jia, Q. Wang, Y. Li, and M. Tan, “Perception-aware multi-sensor fusion for 3D LiDAR semantic segmentation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16260–16270.
- [14] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, “2DPASS: 2D priors assisted semantic segmentation on LiDAR point clouds,” in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 1–16.
- [15] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.
- [16] B. Wang, M. Zhu, Y. Lu, J. Wang, W. Gao, and H. Wei, “Real-time 3D object detection from point cloud through foreground segmentation,” *IEEE Access*, vol. 9, pp. 84886–84898, 2021.

- [17] R. Razani, R. Cheng, E. Taghavi, and L. Bingbing, "Lite-HDseg: LiDAR semantic segmentation using lite harmonic dense convolutions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 9550–9556.
- [18] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 1110–1116.
- [19] Y. Hong, H. Pan, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes," 2021, *arXiv:2101.06085*.
- [20] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," 2021, *arXiv:2105.15203*.
- [21] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5432–5439, Oct. 2020.
- [22] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11105–11114.
- [23] D. Ye, Z. Zhou, W. Chen, Y. Xie, Y. Wang, P. Wang, and H. Foroosh, "LiDARMultiNet: Towards a unified multi-task network for LiDAR perception," 2022, *arXiv:2209.09385*.
- [24] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *Int. J. Robot. Res.*, vol. 37, no. 6, pp. 545–557, May 2018.
- [25] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, K. Yang, and J. Li, "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 797–806.
- [26] P. Jiang and S. Saripalli, "LiDARNet: A boundary-aware domain adaptation model for point cloud semantic segmentation," 2020, *arXiv:2003.01174*.
- [27] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," 2021, *arXiv:2109.13410*.
- [28] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic nusences: A large-scale benchmark for LiDAR panoptic segmentation and tracking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3795–3802, Apr. 2022.
- [29] P. Sun, H. Kretschmar, X. Dotiwala, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, and Y. Chai, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.
- [30] Y. Zhao, L. Bai, and X. Huang, "FIDNet: LiDAR point cloud semantic segmentation with fully interpolation decoding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4453–4458.
- [31] G. Krispel, M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "FuseSeg: LiDAR point cloud segmentation fusing multi-modal data," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1863–1872.
- [32] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Vallespi-Gonzalez, "Sensor fusion for joint 3D object detection and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1230–1237.
- [33] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4603–4611.
- [34] F. Zhang, J. Fang, B. W. Wah, and P. H. S. Torr, "Deep fusionnet for point cloud semantic segmentation," in *Proc. ECCV*, 2020, pp. 1–10.
- [35] I. Shepel, V. Adeshkin, I. Belkin, and D. A. Yudin, "Occupancy grid generation with dynamic obstacle segmentation in stereo images," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14779–14789, Sep. 2022.
- [36] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.
- [37] P. Jiang and S. Saripalli, "LiDARNet: A boundary-aware domain adaptation model for point cloud semantic segmentation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2457–2464.
- [38] S. Zhao, Y. Wang, B. Li, and B. Wu, "ePointDA: An end-to-end simulation-to-real domain adaptation framework for LiDAR point cloud segmentation," in *Proc. AAAI Conf. Artif. Intell.*, Dec. 2021, vol. 35, no. 4, pp. 3500–3509.
- [39] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4376–4382.
- [40] D. Tsai, J. Stephany Berrio, M. Shan, E. Nebot, and S. Worrall, "Viewer-centred surface completion for unsupervised domain adaptation in 3D object detection," 2022, *arXiv:2209.06407*.
- [41] Y. Wang, J. Yin, W. Li, P. Frossard, R. Yang, and J. Shen, "SSDA3D: Semi-supervised domain adaptation for 3D object detection from point cloud," 2022, *arXiv:2212.02845*.
- [42] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A point cloud dataset with large quantity of dynamic instances," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 687–693.
- [43] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, Y. Wang, and D. Yang, "PandaSet: Advanced sensor suite dataset for autonomous driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3095–3101.
- [44] Y. Yuan and J. Wang, "OCNET: Object context network for scene parsing," 2018, *arXiv:1809.00916*.
- [45] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4784–4793.
- [46] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 123–145.
- [47] M. Zahangir Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation," 2018, *arXiv:1802.06955*.
- [48] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1689–1696.
- [49] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An open-source mesh processing tool," in *Proc. Eurograph. Italian Chapter Conf.*, Salerno, Italy, 2008, pp. 129–136.
- [50] L. Briceno and G. Paul, "MakeHuman: A review of the modelling framework," in *Proc. 20th Congr. Int. Ergonom. Assoc. (IEA)*. Cham, Switzerland: Springer, 2019, pp. 224–232.
- [51] E. Ackerman, "Latest version of gazebo simulator makes it easier than ever to not build a robot," *IEEE Spectr.*, Feb. 2016. [Online]. Available: <https://spectrum.ieee.org/latest-version-of-gazebo-simulator>
- [52] Y. Murhij and D. Yudin, "FMFNet: Improve the 3D object detection and tracking via feature map flow," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.



ALEXEY A. KLOKOV was born in Moscow, Russia, in 1995. He received the master's degree in applied math and physics from the Moscow Institute of Physics and Technology (MIPT), in 2020. He is currently pursuing the Ph.D. degree with the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (FRC CSC RAS).

Since 2020, he has been a Lead Data Scientist at Autonomics.tech. He headed the Department of Signal AI-Processing from all robot's sensors.



DI UN PAK was born in Novosibirsk, Russia, in 2000. He received the B.S. degree in computer science from the Higher School of Economics, Moscow, Russia, in 2022.

From 2021 to 2022, he was a Research Assistant with the Intelligent Transport Laboratory, Moscow Institute of Physics and Technology, Moscow. His main research interests lay in computer vision and its applications in robotics.



ALEKSANDR KHORIN was born in Tula, Russia, in 1999. He received the B.A.Sc. degree in mathematics and computer science from the Moscow Institute of Physics and Technology, Moscow, Russia, in 2021.

Since 2021, he has been a Research Engineer with the Intelligent Transport Laboratory, Moscow Institute of Physics and Technology. His research interests include computer vision, optical character recognition, and deep learning.

Mr. Khorin was the bronze medalist from the All-Russian Student Olympiad “Yandex” in “Mathematical Modeling” in 2021 and was a recipient of the grant of the President of the Russian Federation for outstanding abilities in educational and scientific activities from 2021 to 2022.



DMITRY A. YUDIN was born in Belgorod, Russia, in 1988. He received the Engineering Diploma degree in automation of technological processes and production, in 2010, and the Ph.D. degree in computer science from the Belgorod State Technological University (BSTU named after V. G. Shukhov), Belgorod, in 2014.

From 2009 to 2019, he was a Researcher and an Assistant Professor with the Technical Cybernetics Department, BSTU. Since 2019, he has been the Head of the Intelligent Transport Laboratory, Moscow Institute of Physics and Technology, Moscow, Russia. Since 2021, he has been a Senior Researcher with the Artificial Intelligence Research Institute (AIRI), Moscow. He is the author more than 100 articles. His research interests include computer vision, deep learning, and robotics.

Dr. Yudin was a recipient of the IEEE Czechoslovakia Section Young Scientist Paper Award, in 2017, the grant of the President of the Russian Federation for state support of young Russian scientists, from 2017 to 2018, and the grant of the Russian Science Foundation, in 2021.



LEON KOCHIEV was born in Vladikavkaz, Russia, in 1998. He received the M.Sc. degree in computer science from Simon Fraser University (SFU), Burnaby, in 2021.

From 2021 to 2022, he was a Machine Learning Engineer at Autonomics.tech. His main research interests lay in 3D computer vision and visual question answering.



VLADIMIR D. LUCHINSKIY was born in Ulyanovsk, Russia, in 1998. He received the bachelor's degree in the software and administration of information systems from Saint-Petersburg State University (SPbU), in 2020.

From 2020 to 2021, he was a Machine Learning Engineer with Autonomics.tech. His research interests include deep learning computer vision problems.



VITALY D. BEZUGLYY was born in Kiev, Ukraine, in 2000. He received the bachelor's degree in applied mathematics and computer science from MEPhI, National Research Nuclear University, Moscow, Russia, in 2022.

Since 2021, he has been a Research Engineer with the Intelligent Transport Laboratory, Moscow Institute of Physics and Technology, Moscow. His research interests include 3D semantic mapping, computer vision, and robotics.

...