A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI

Jens Behley Andres Milioto Cyrill Stachniss

***Abstract*— Panoptic segmentation is the recently introduced task that tackles semantic segmentation and instance segmentation jointly [18]. In this paper, we present an extension of SemanticKITTI [1], a large-scale dataset providing dense point-wise semantic labels for all sequences of the KITTI Odometry Benchmark [10]. This extension enables training and evaluation of LiDAR-based panoptic segmentation. We provide the data and discuss the processing steps needed to enrich a given semantic annotation with temporally consistent instance information, i.e., instance information that supplements the semantic labels and identifies the same instance over sequences of LiDAR point clouds. Additionally, we present two strong baselines that combine state-of-the-art LiDAR-based semantic segmentation approaches with a state-of-the-art detector enriching the segmentation with instance information and that allow other researchers to compare their approaches against. We believe that our extension of SemanticKITTI with strong baselines enables the creation of novel algorithms for LiDAR-based panoptic segmentation as much as it has for the original semantic segmentation and semantic scene completion tasks. Data, code, and an online evaluation service using a hidden test set are publicly available at http://semantic-kitti.org.**

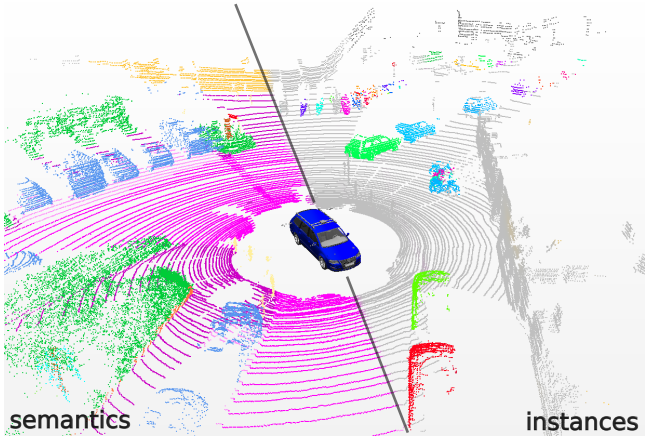

Fig. 1: Using the semantic segmentation (left part) and the point-accurate instance annotations for traffic participants (right part), we provide a benchmark for panoptic segmentation [18] using three-dimensional LiDAR point clouds. Our work extends the SemanticKITTI [1] dataset, which is based on the KITTI Vision Benchmark [10].

I. Introduction

Fine-grained scene understanding is a pre-requisite for truly autonomous systems, such as self-driving cars. This encompasses the type of surfaces, but also identifying individual objects. The former is often designated as *stuff* and the latter as *things* [18]. Only both sources of information together enable autonomous systems to reason about the drivability of surfaces, the type of objects and obstacles, and possibly the intent of other agents in the vicinity.

Assigning to each individual pixel or point a semantic label is called semantic segmentation, while the identification and separation of individual objects is called instance segmentation. These tasks are usually solved in isolation, but an increasing number of methods have been recently developed that solve both jointly using either images [18], [32], [41], [39], [24], [7] or RGB-D data [30], [14], [16], [9], [43], [40], [26], [13]. These developments were mainly driven by the availability of a metric [18] and the swift adaption of the task in different popular semantic segmentation datasets, such as Cityscapes [8], Microsoft's Common Objects in Context (COCO) [20], and Mapillary Vistas [27]. While semantic segmentation will still be relevant in the future, we expect that instance segmentation will be soon replaced and subsumed by the panoptic segmentation task, as it is a part of a panoptic segmentation framework.

In this paper, we present an extension of the SemanticKITTI dataset [1] providing the necessary annotations to evaluate panoptic segmentation on automotive LiDAR scans. Fig. 1 shows an example of the provided instance annotation for all traffic participants, i.e., vehicles, pedestrians, and cyclists. To ease the generation of instance information with provided semantic segmentation of the LiDAR point clouds, we first generate for static and non-static objects instance information using grid-based clustering [4] and distance-based clustering approach. Unfortunately, such a clustering often leads to over- or under-segmentation, which we had to manually correct using our point labeling tool. Furthermore, we provide two baseline approaches that combine state-of-the-art semantic segmentation with state-of-the-art object detection methods.

In summary, our contributions are as follows:

- We provide temporally-consistent instance annotations for all traffic participants including vehicles, pedestrians, bicyclists, and motorcyclists for the KITTI Odometry Benchmark.
- We provide two strong baseline approaches combining current state-of-the-art semantic segmentation and a state-of-the-art 3D object detector.
- We provide a benchmark with a publicly available online evaluation service for approaches solving LiDAR-based panoptic segmentation using a hidden test set.

All authors are with the University of Bonn, Germany.

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number BE 5996/1-1 and under Germany's Excellence Strategy, EXC-2070 - 390732324 - PhenoRob.

TABLE I: Overview of other LiDAR datasets with annotations for instances (top) and semantic segmentation (bottom).

	Name	#Scans[1]	#Boxes/#Points	#Classes[2]	Data[3]	FoV[4]	Sequential	Reference
Instance Annotation	KITTI (Detection)	7k/7k	1k	3(3)	B	F	✗	[10]
	Argoverse	22k	993k	17	B	C	✓	[6]
	Lyft	46k	1.3M	9	B	C	✓	[17]
	CADC	7k	305k	10	B	C	✓	[31]
	Waymo	200k	12M	4	B	C	✓	[37]
	A2D2	12k	12k	14	B	F	✗	[11]
	H3D	27k	1.1M	8	B	F	✗	[29]
	PandaSet	16k	1.4M	12	B	C	✓	[36]
	nuScenes	44k	1.4M	10 (23)	B	C	✓	[5]
	SemanticKITTI	23k/20k	682k	8	P	C	✓	
Semantic Segmentation	Oakland3d	17	1.6M	5 (44)	P	C	✗	[25]
	Freiburg	77	1.1M	4 (11)	P	C	✗	[3]
	Wachtberg	5	400k	5 (5)	P	C	✗	[3]
	Semantic3d	15/15	4009M	8 (8)	P	C	✗	[12]
	Paris-Lille-3D	3	143M	9 (50)	P	C	✗	[35]
	Zhang et al.	140/112	32M	10 (10)	P	F	✗	[42]
	SemanticPOSS	2k	216M	14	P	C	✗	[28]
	A2D2	31k	930k	38	P†	F	✗	[11]
	PandaSet	16k	1388M	42	P	C	✓	[36]
	nuScenes	40k	1400M	32	P	C	(✓)	[5]
	SemanticKITTI	23k/20k	4549M	25 (28)	P	C	✓	[1]

[1] Number of scans for train and test set, [2] Number of classes used for evaluation and number of classes annotated in brackets, [3] type of annotations, where B and P correspond to bounding boxes (B) and point-wise (P), [4] field-of-view (FoV) of LiDAR sensor with annotations, where F denotes frontal and C denotes complete 360°. † point-wise annotations via projection to annotated image and using corresponding image label.

II. Related Work

Shortly after Kirillov *et al*. [18] proposed panoptic segmentation and a metric to measure the performance of approaches providing such labels, the established datasets for semantic segmentation of *image data*, i.e., Cityscapes [8], Microsoft's Common Objects in Context (COCO) [20], and Mapillary's Vistas [27] adopted the metric and added an evaluation for this task.

Due to the availability of the data, we witnessed a wide adoption and interest for panoptic segmentation in the computer vision community [7], [18], [21], [30], [32], [41], [39], [24]. While there have also been approaches for RGB-D data [14], [30], [9], [16], [43], [40], [13], there were basically no approaches available that operate on LiDAR data, when we released the data in April 2020. There was simply no annotated data available that provided both point-wise semantic labels *and* instance information. Recently, first approaches adopted the provided annotations to develop approaches for LiDAR-based panoptic segmentation [22], [15].

Recently, almost all major self-driving car companies release datasets providing besides camera also LiDAR data [37], [17], [11], [6], [5]. While most datasets provide also annotations for object instances by bounding boxes, only a few datasets provide point-wise semantic annotation [1], [11], [36], [5] needed to evaluate panoptic segmentation for LiDAR. Tab. I summarizes the amount of data provided by the different datasets.

SemanticKITTI [1] is a dataset based on the KITTI Vision Benchmark [10], which might not show the diversity of different inner cities traffic and weather conditions, but still provides unparalleled long sequences showing a variety of different environments and driving situations. Our annotations with point-wise labels for the full 360° field-of-view provide labels for 28 classes including labels distinguishing moving and non-moving objects. By providing now instance annotations together with an online evaluation on a hidden test set, we close the gap to the aforementioned established image-based dataset and provide a benchmark for panoptic segmentation using an automotive LiDAR. We hope that the availability of labeled LiDAR scans for panoptic segmentation opens the door for more research in the direction of LiDAR-based panoptic segmentation.

III. Dataset

In this section, we introduce the provided dataset and discuss the annotation process to extract instance information from a given semantic segmentation in a semi-automatic fashion with acceptable manual labeling effort to adjust for wrong over- and under-segmentations.

Fig. 2 shows a qualitative example of the annotation provided by our dataset. The left part of the figure shows the semantic segmentation of our SematicKITTI dataset, which we use to determine the instance annotation. The right side depicts the temporally consistent annotations, where different colors correspond to individual instances. Note, that same colored instances in the top and the bottom row of this figure correspond to the same instance ID.

A. Annotation Process

For annotation of the instances, we employ a semi-automatic process using different strategies to generate a temporally consistent instance annotation. Our goal is to

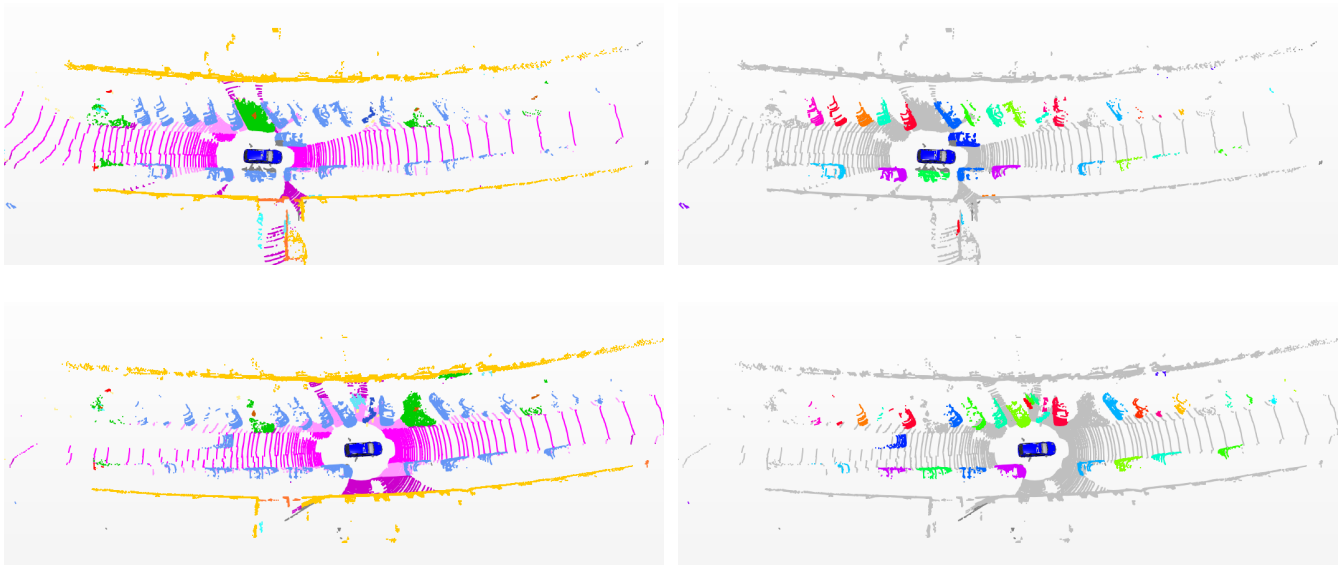

Fig. 2: Qualitative example of the instance annotation over a sequence of scans: on the left is the semantic annotation and on the right is the instance annotation shown. Top and bottom rows show consecutive timestamps from sequence 13. Note, same colors at different timestamps correspond to the same instance id. Best viewed in color.

label the same instance through the whole sequence with the same instance ID – even for instances that move. For static objects, the data association can be simply performed by considering the location of the segment after performing a pose correction using a Simultaneous Localization and Mapping (SLAM) system [2]. For moving objects, we have to account for the motion of the object as well as the motion of the sensor at the same time.

Overall, the SemanticKITTI dataset [1] provides 28 classes (including 6 classes to distinguish moving from non-moving classes) from which we select the traffic participants as *thing* classes for the panoptic segmentation, i.e., car, truck, other-vehicle, motorcycle, bicycle, person, bicyclist, and motorcyclist. The remaining classes are *stuff* classes for the panoptic segmentation, i.e., road, sidewalk, parking, other-ground, building, vegetation, trunk, terrain, fence, pole, and traffic-sign.

For *static thing classes*, we first cluster all points for each individual class using a fast grid-based segmentation approach [4] to handle the large number of points efficiently. We then split the aggregated point cloud into tiles of size $100\,\mathrm{m}$ by $100\,\mathrm{m}$ using the pose information by our SLAM system [2]. For each tile, we use a two-dimensional grid with cell size $0.1\,\mathrm{m}$ by $0.1\,\mathrm{m}$, which allows us often to separate even close parking cars. Next, all points are inserted into the corresponding grid cells using their x and y-coordinates. Finally, only grid cells with points exceeding a height threshold $\Delta > 0.5\,\mathrm{m}$ are combined using a flood fill algorithm to combine neighboring grid cells into segments.

For *moving thing classes*, we generate clusters for each scan individually using a distance-based clustering as this provided more reliable results and could be also used to associate instances between consecutive scans. First, we search for each point in a radius of $0.5\,\mathrm{m}$ for the nearest neighbor and cluster points together that share neighbors. To find associations with the previous 4 scans, we use a slightly larger radius of $1.0\,\mathrm{m}$ to find neighbors between two different timestamps. If we find enough neighbors with the previous segments at different timestamps, we associate them together and assign the same instance ID.

The described clustering leads inevitably to over- and under-segmentation (cf. Fig. 3), but also to wrong or missing associations between consecutive timestamps. We correct these issues manually using an own point labeling tool, which provides tools to create, join, and split instances. Overall, the manual correction for all 22 sequences took roughly $70\,\mathrm{h}$ of additional labor.

B. Statistics

Fig. 4 provides an overview of the number of instances and the actual number of bounding boxes per class. We show in the upper part of the figure the sequence-wise counts of instance annotations, i.e., we count each object only once, even if it is seen multiple times by the sensor. The lower part of the figure shows the accumulated scan-wise counts of instances, where we count the instances without considering the temporally consistent instance ID.

The bulk of the instances correspond to cars, which are naturally occurring in city-like environments and also correspond to the normal statistics in autonomous driving scenarios. Usually, an autonomous car will also encounter some classes far fewer than other classes or situations. They are usually denoted as the 'long tail' problem, referring to the underrepresented entities in a given distribution. This adds complexity to the task, since panoptic segmentation approaches, which are designed to tackle this scenario must be able to deal with such skewed class distributions.

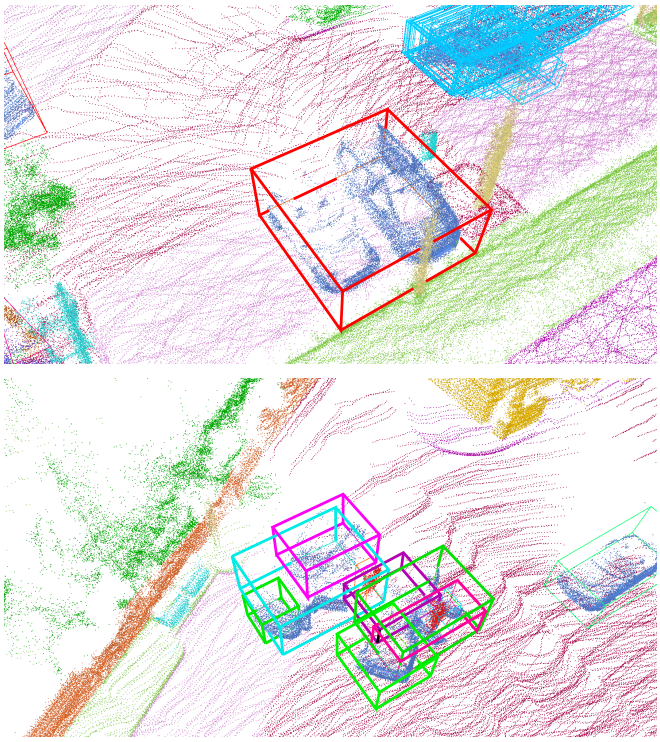

Fig. 3: Example of under- (top) and over-segmentation (bottom) generated by our semi-automated clustering approach, which we manually corrected by splitting or joining segments.

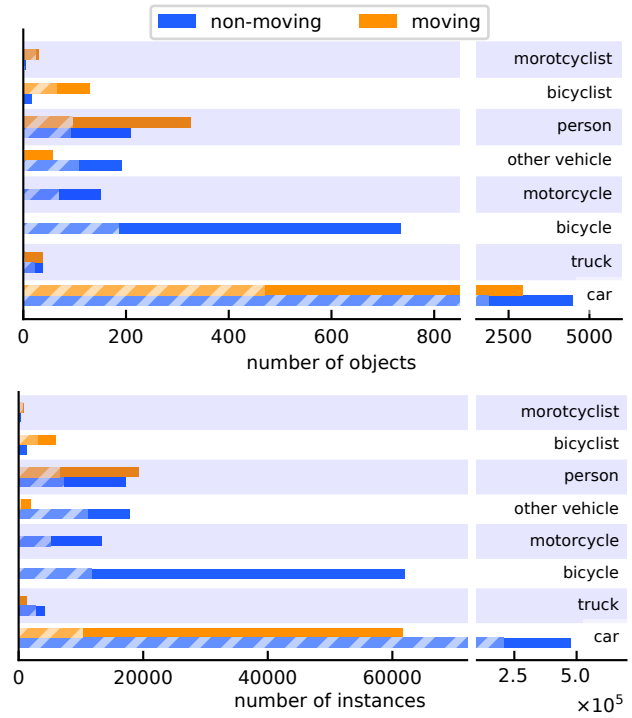

Fig. 4: Top: number of (sequence-wise) objects. Bottom: number of (scan-wise) instances. The hashed bars correspond to the training data. The large number of scan-wise annotations in relation to the number of objects indicates that many objects are seen over an extended period of time.

IV. Baseline Approaches

None of the available RGB-D approaches [14], [30], [13], [9], [16], [43], [40] could be easily adapted to the size of the data and the specific characteristics of the LiDAR point clouds. Available approaches for RGB-D panoptic segmentation either use a truncated signed distance function (TSDF) [14], [26] or voxel grids [13], or employ PointNets [33], [34] for feature extraction [30], [9], [16], [43], [40]. A TDSF representation and PointNets perform purely on single point clouds due to the characteristic sparsity pattern of the rotating LiDAR sensor, which generates dense point clouds at close ranges and sparse point clouds at larger ranges. Furthermore, our evaluation for the SemanticKITTI datatset [1] of variants of the PointNets [33], [34] showed inferior performance with such point clouds.

Thus, we propose two baseline approaches combining current state-of-the-art semantic segmentation approaches on SemanticKITTI, namely KPConv [38] and RangeNet++ [23], paired with a state-of-the-art object detector on the KITTI 3D object detection benchmark, namely PointPillars [19], providing instance-level information.

To this end, we use the oriented bounding boxes of the object detector, i.e., bounding boxes for *cars*, *pedestrians*, and *cyclists* trained on the KITTI detections benchmark [10], to determine the instance ID for points inside the bounding boxes. By combining the predictions of the semantic segmentation and assigning the instance ID of each bounding box to each point inside of it, we obtain a panoptic segmentation. Note that we only assign instance IDs to points from the *thing* classes, i.e., points under a *car* classified as *road* or *parking* are not assigned an instance ID.

For the baseline, we used pre-trained models or publicly available predictions for KPConv [38] and RangeNet++ [23], which were trained on SemanticKITTI. PointPillars had to be trained from scratch using the provided implementation[1], modifying the configuration of the object detector such that it provides region proposals and bounding boxes for the full 360-degree field-of-view of the LiDAR sensor.

These networks were run independently for semantic segmentation and object detection and then merged to generate a panoptic segmentation. None of the approaches can, therefore, run at the frame rate of the LiDAR, i.e., 10 Hz, and thus having computational budgets that are not suitable in an autonomous car. Furthermore, the PointPillars detector [19] requires training separate networks, one for the class *car* and one for *pedestrian* combined with *cyclist*, which accentuates the problem further. We provide an evaluation of the performance of these approaches including runtime information in the experimental section of this paper.

Note that the decision for using an object detector providing oriented bounding boxes was made to minimize the negative effect of axis-aligned bounding boxes, which would lead to large overlaps between cars parking near to each other, see also Fig. 5 for an example. Thus, oriented bounding boxes lead to more accurate instance annotations in the depicted case.

[1]See the GitHub repository at https://git.io/Je251.

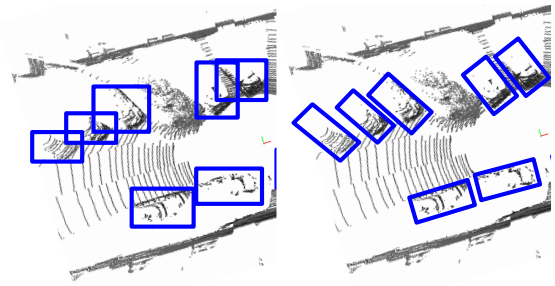

Fig. 5: Overlapping of axis-aligned bounding boxes and therefore wrong or ambiguous assignment of points inside bounding boxes (left). With oriented bounding boxes this ambiguity due to overlapping bounding boxes does not occur (right).

V. Experiments

Before we discuss details of the baseline implementations and the results of our baseline approaches, we shortly provide a summary of the panoptic segmentation metric.

A. Evaluation Metric

In panoptic segmentation, each point $\mathbf{p}_i$ not only carries a class label $y_i \in \mathcal{Y}$, where $|\mathcal{Y}|$ is the number of classes, but also can have an instance ID n_i, where $n_i = 0$ denotes no specific instance.

To measure the quality of this joint assignment, we briefly recapitulate the recently proposed panoptic quality (PQ) metric [18]. Let $\mathcal{S}, \hat{\mathcal{S}}$ denote segments, i.e., sets of points in our specific case, sharing an class and instance ID. Here, we assume that the *stuff* classes simply get instance ID $n_i = 0$ corresponding to no specific instance assigned.

Furthermore, let $\text{IoU}(\mathcal{S}, \hat{\mathcal{S}}) = (\mathcal{S} \cap \hat{\mathcal{S}}) \cdot (\mathcal{S} \cup \hat{\mathcal{S}})^{-1}$ denote the intersection-over-union of these two sets. Let the set of true positive matches TP_c be the pairs of predicted segments $\hat{\mathcal{S}}$ that overlap at least with 0.5 IoU with a ground truth segment $\mathcal{S}$, $\text{TP}_c = \{(\mathcal{S}, \hat{\mathcal{S}}) \mid \text{IoU}(\mathcal{S}, \hat{\mathcal{S}}) > 0.5\}$. Likewise, let FP_c the set of unmatched predicted segments $\hat{\mathcal{S}}$ and FN_c the set of unmatched ground truth segments $\mathcal{S}$.

With the above definitions, the class-wise PQ_c is given by

$$\text{PQ}_c = \frac{\sum_{(\mathcal{S},\hat{\mathcal{S}}) \in \text{TP}_c} \text{IoU}(\mathcal{S}, \hat{\mathcal{S}})}{|\text{TP}_c| + \frac{1}{2}|\text{FP}_c| + \frac{1}{2}|\text{FN}_c|}. \tag{1}$$

The panoptic quality metric is computed for each class independently and averaged over all classes, which makes the metric insensitive to class imbalance [18], i.e.,

$$\text{PQ} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \text{PQ}_c. \tag{2}$$

Kirillov *et al.* [18] furthermore define the segmentation quality (SQ) as average IoU over matched segments and the recognition quality (RQ) corresponding to the F_1 score.

Porzi *et al.* [32] proposed to alter the metric to account for *stuff* classes having only a single segment since no pixels (or, in our case, points) have an instance ID. Hence, the IoU-based criterion could often lead to an unmatched prediction. To account for *stuff* classes, Porzi *et al.* use

$$\text{PQ}_c^{\dagger} = \begin{cases} \text{IoU}(\mathcal{S}, \hat{\mathcal{S}}) & \text{, if c is a } \textit{stuff} \text{ class} \\ \text{PQ}_c & \text{, otherwise.} \end{cases} \tag{3}$$

Consequently, we denote by $\text{PQ}^{\dagger}$ the average over the class-wise modified $\text{PQ}_c^{\dagger}$ as defined in (2).

Furthermore, the quality of the semantic segmentation is also measured using the mean intersection-over-union (mIoU), which also enables the comparison with other approaches in the semantic segmentation benchmark. This metric is defined as follows:

$$\text{mIoU} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \frac{|\{i \mid y_i = c\} \cap \{j \mid \hat{y}_j = c\}|}{|\{i \mid y_i = c\} \cup \{j \mid \hat{y}_j = c\}|}, \tag{4}$$

where y_i corresponds to the ground truth label of point $\mathbf{p}_i$ and $\hat{y}_i$ to the prediction.

B. Baseline Parameters, Training, and Inference Details

In this section, we provide more details on the training and inference of the two-stage baselines. We, furthermore, provide details on the modifications needed to use the models on the SemanticKITTI [1] benchmark, which requires to use full point clouds of a single turn for training and inference. We will provide code for merging the predictions to enable the reproduction of our results.

KPConv by Thomas *et al.* [38]. For scene classification, Thomas *et al.* [38] extract 10 overlapping spheres of 10 m radius, subsample the point clouds with a voxel grid (resolution of 0.1 m), and drop random points in case there are more than $15,000$ points left. To aggregate predictions, they perform majority vote on the overlapping parts of the predictions. Overall, this achieves state-of-the-art single scan performance of 58.5 mIoU and performs better than taking a subsampled single point cloud with mIoU 56.6 (subsampling with voxel grid of resolution 0.1 m).

RangeNet++ by Milioto *et al.* [23]. Here, we directly use the predictions available in our repository[2], which are also provided for the test set. RangeNet++ uses a range image of size 2048×64 for training and inferences, which is then upsampled to the complete point cloud by using nearest neighbors. To remove artifacts from the reprojection, it applies a k-nearest neighbor filtering, which accounts for k neighbors in a certain range.

PointPillars by Lang *et al.* [19]. We used for training of the approach the aforementioned implementation supported by the Point Pillar authors. Since SemanticKITTI does not offer oriented bounding boxes, we use the 3D object detection part of KITTI Object Detection [10] for training. The KITTI dataset was recorded with the same sensor and a similar environment, but there is no overlap between the point cloud sequences of the odometry and the detection benchmark.

For training on the KITTI object detection, we follow the original approach of Lang *et al.* [19] and use 0.16 m as voxel grid resolution with a maximum of 12.000 pillars with at most 100 points for each pillar for training on the KITTI Object Detection subset of the KITTI Vision Benchmark [10]. As commonly done and also advocated by Lang *et al.* [19], we trained a network for *cars*, car network,

[2] https://github.com/PRBonn/lidar-bonnetal

Method	mIoU	PQ	PQ†	RQ	SQ	PQ^{Th}	RQ^{Th}	SQ^{Th}	PQ^{St}	RQ^{St}	SQ^{St}
KPConv [38] + PointPillars [19]	58.8	44.5	52.5	54.4	80.0	32.7	38.7	81.5	53.1	65.9	79.0
RangeNet++ [23] + PointPillars [19]	52.4	37.1	45.9	47.0	75.9	20.2	25.2	75.2	49.3	62.8	76.5

TABLE II: Comparison of test set results on SemanticKITTI using *stuff*(St) and *thing*(Th) *classes*. All results in [%].

Method	PQ	road	sidewalk	parking	other ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign
KPConv [38] + PointPillars [19]	44.5	84.6	60.1	34.1	8.8	80.7	72.5	17.2	9.2	30.8	19.6	77.6	53.9	42.2	29.9	59.4	22.8	49.0	46.2	46.8
RangeNet++ [23] + PointPillars [19]	37.1	90.6	63.2	41.3	6.7	79.2	66.9	6.7	3.1	16.2	8.8	71.2	34.6	37.4	14.6	31.8	13.5	38.2	32.8	47.4

TABLE III: Detailed class-wise results of test set results on SemanticKITTI in panoptic quality (PQ) [18]. All results in [%].

Method	PQ†	road	sidewalk	parking	other ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign
KPConv [38] + PointPillars [19]	52.5	88.8	72.7	61.3	31.6	90.5	72.5	17.2	9.2	30.8	19.6	84.8	69.2	69.1	29.9	59.4	22.8	64.2	56.4	47.4
RangeNet++ [23] + PointPillars [19]	45.9	91.8	75.1	65.0	27.7	87.4	66.9	6.7	3.1	16.2	8.8	80.5	55.1	64.8	14.6	31.8	13.5	58.6	47.9	55.9

TABLE IV: Detailed class-wise results of test set results on SemanticKITTI in fixed panoptic quality ($PQ^{\dagger}$) [32]. All results in [%].

and a separate network for *pedestrian* and *cyclist*, called pedcyclist network.

For the car network, we consider the part in front of the sensor inside the ranges $x = (0.0, 69.12)$, $y = (-39.68, 39.68)$, and $z = (-3.0, 1.0)$, where we assume that the sensor is located at $(0, 0, 0)$. For the pedcyclist network, we use $x = (0.0, 48.0)$, $y = (-20.0, 20.0)$, and $z = (-2.5, 0.5)$ as volume of the point pillar grid.

For prediction on the SemanticKITTI dataset, we are interested in predicting bounding boxes for the full field-of-view of the sensor. Thus, we adapted the parameters for inference. For the car network, we use a grid volume of size $x = (-69.12, 69.12)$, $y = (-69.12, 69.12)$, $z = (-3.0, 1.0)$. For the pedcyclist network, we use a similar grid volume of $x = (-69.12, 69.12)$, $y = (-69.12, 69.12)$, $z = (-2.5, 0.5)$. Furthermore, we increase the number of maximal pillars to 30000 and adopt the anchor generation strides to accommodate the large input volume.

Considering the runtime of the proposed two-stage approach, we observe a large discrepancy between the reported runtime and our obtained runtime, which cannot be only explained by using a different system (Nvidia Geforce RTX 2080 Ti vs. a Nvidia Geforce 1080 Ti). First, we have to note that the implementations might be different from the originally used implementation. We believe that the main reason seems to be the 3.4 times increase input volume and the increased number of pillars. The fact that the KITTI object detection benchmark only uses a part of the point cloud is also acknowledged in Sec. 6 of Lang *et al.* [19]. We furthermore do not use TensorRT for inference, which could additionally improve the runtime.

C. Baseline Results

Tab. II summarizes the results in a breakdown according to mean Intersection-over-Union (mIoU) and the different panoptic quality metrics. Due to the overall stronger performance on semantic segmentation of KPConv (58.8 mIoU vs. 52.4 mIoU in Tab. II), the panoptic baseline using KPConv is stronger in all metrics. We believe that this discrepancy can be directly attributed to the stronger performance on small classes. Tab. III and Tab. IV show the detailed results for all classes using the panoptic quality and the fixed panoptic quality, respectively.

For the runtime, we assume that the separate object detectors can be run in parallel (314 ms for *pedestrian/cyclist* and 105 ms for *car*) after the semantic segmentation (200 ms for KPConv and 95 ms for RangeNet++) resulting in 514 ms and 409 ms respectively.

VI. Conclusion

We present an extension of the SemanticKITTI dataset that enables the community to evaluate and benchmark panoptic segmentation approaches using data generated by an automotive LiDAR. We provide the data, code, as well as, an online platform for evaluation using a hidden test set. Additionally, we provide two panoptic segmentation baselines that are built from a combination of state-of-the-art semantic segmentation approaches and a 3D object detector. The goal of this dataset paper is to propel the research on LiDAR-based panoptic segmentation, since it is an important task that will become more relevant, and provide a platform for easy benchmarking of such approaches.

Acknowledgment

We thank Hugues Thomas for allowing us to use the predictions from his original KPConv approach for our work.

REFERENCES

[1] 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. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[2] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[3] J. Behley, V. Steinhage, and A. Cremers. Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[4] J. Behley, V. Steinhage, and A. Cremers. Laser-based Segment Classification Using a Mixture of Bag-of-Words. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[5] H. Caesar, V. Bankiti, A. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] B. Cheng, M.D. Collins, Y. Zhu, T. Liu, T.S. Huang, H. Adam, and L.C. Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] M. Cordts, S.M. Omran, Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] L. Du, J. Tan, X. Xue, L. Chen, H. Wen, J. Feng, J. Li, and X. Zhang. 3DCFS: Fast and Robust Joint 3D Semantic-Instance Segmentation via Coupled Feature Selection. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[11] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V.H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth. A2D2: Audi Autonomous Driving Dataset. *arXiv preprint*, 2020.

[12] T. Hackel, N. Savinov, L. Ladicky, J.D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017.

[13] L. Han, T. Zheng, L. Xu, and L. Fang. OccuSeg: Occupancy-aware 3D Instance Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] J. Hou, A. Dai, and M. Niessner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] J. Hurtado, R. Mohan, W. Burgard, and A. Valada. MOPT: Multi-Object Panoptic Tracking. *arXiv preprint:2004.08189*, 2020.

[16] H. Jiang, F. Yan, J. Cai, J. Zheng, and J. Xiao. End-to-end 3D Point Cloud Instance Segmentation without Detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft Level 5 AV Dataset 2019. https://level5.lyft.com/dataset/, 2019.

[18] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] A. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–755, 2014.

[21] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang. An End-To-End Network for Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] A. Milioto, J. Behley, C. McCool, and C. Stachniss. LiDAR Panoptic Segmentation for Autonomous Driving. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[23] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[24] R. Mohan and A. Valada. EfficientPS: Efficient Panoptic Segmentation. *arXiv preprint*, 2004.02307v2, 2020.

[25] D. Munoz, J.A. Bagnell, N. Vandapel, and M. Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[26] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[27] G. Neuhold, T. Ollmann, S.R. Bulo, and P. Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[28] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao. SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances. *arXiv preprint:2002.09147*, 2020.

[29] A. Patil, S. Malla, H. Gang, and Y.T. Chen. The H3D dataset forfull-surround 3D multi-object detection and tracking in crowded urban scenes. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[30] Q. Pham, D. Nguyen, B. Hua, G. Roig, and S. Yeung. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds With Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[31] M. Pitropov, G. Danson, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander. Canadian Adverse Driving Conditions Dataset. *arXiv preprint:2001.10117*, 2020.

[32] L. Porzi, S.R. Bulo, A. Colovic, and P. Kontschieder. Seamless Scene Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[34] C. Qi, K. Yi, H. Su, and L.J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2017.

[35] X. Roynard, J. Deschaud, and F. Goulette. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Intl. Journal of Robotics Research (IJRR)*, 37(6):545–557, 2018.

[36] scale and Hesai. PandaSet Dataset (Available at https://scale.com/open-datasets/pandaset), 2020.

[37] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[39] H. Wang, R. Luo, M. Maire, and G. Shakhnarovich. Pixel Consensus Voting for Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia. Associatively Segmenting Instances and Semantics in Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[41] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. UPSNet: A Unified Panoptic Segmentation Network. In *Proc. of*

the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2019.

[42] J. Zhang and S. Singh. Visual-Lidar Odometry and Mapping: Low-Drift, Robust, and Fast. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[43] L. Zhao and W. Tao. JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds. In *Proc. of the Conference on Advancements of Artificial Intelligence (AAAI)*, 2020.