The LiDAR Self-Supervised Learning Challenge: Learning From a Limited Amount of High-Resolution LiDAR Data

Frederik Hasecke[1,2] and Anton Kummert[1]

[1]Bergische Universität Wuppertal [2]Aptiv

1 Introduction & Background

This report summarizes our approach to train a 3D object detection network from a dataset with about 1,200 LiDAR frames of which only 100 are annotated. We developed a combination of various augmentation methods for self-supervised training. We trained the same network with a given set of hyperparameters on three sets of data: once on the raw training data, once with our proposed augmentation methods of SECTION 2, and lastly with additional pseudo labels. As we highlight in our main results in SECTION 3: our augmentations and pseudo labels improved the network especially for rare classes. Our proposed solution outlined in the next section draws inspiration from [2], in which the authors propose two new LiDAR point cloud augmentation methods for semantic segmentation. Besides that, we make use of the *OpenPCDet* [5] implementation of *PV-RCNN* [1] for the provided *InnovizTwo* dataset [4].

2 Method

In a first statistical analysis, we counted the number of instances per target class. In order to have both a training and a validation set, we split the 103 annotated LiDAR frames so that both sets had a similar class distribution. Thus, we end up with a training set of 53 frames and 50 validation frames. We divide our method into three main aspects: instance injections, scene fusion, and our novel use of pseudo labels. Our focus for the injection and fusion augmentations is to manipulate the data only in certain ways, such that the resulting scene could in fact be recorded by the sensor. In plain language, we avoid any manipulation to the point cloud which the sensor is not able to capture due to the underlying physical limitations such as resolution, range and occlusion. An example of a non valid injection would be a pedestrian standing behind a high wall.

2.1 Instance Injection

To ensure a structurally intact instance injection, we do not simply consider the objects as the contiguous contents of a box, but check each individual LiDAR point belonging to the object. The range image projection is an efficient method for this task [2]. We move the point clouds of the dataset back to the position of the LiDAR sensor, in order to be able to use a coordinate transformation of the LiDAR points as an image-like representation. Due to the discrete sampling of the individual LiDAR channels, and the absence of intersections between the individual channels, we can assign each LiDAR point to a fixed pixel within the range image. This feature allows us to inject an object fully or partially into the scene, depending on whether the corresponding LiDAR point of the new object or an existing object within the original scene is closer to the LiDAR sensor. The projection onto a range image representation reduces the dimensionality to a two instead of a three-dimensional problem, which is computationally much more efficient.

We sample the chosen training data split for all bounding box labels and the points inside of those boxes. Thus we create a database of 439 cars, 19 pedestrians, 14 cyclists and 36 trucks. These objects can be injected only at two relative positions to the LiDAR sensor due to the underlying physical properties of the InnovizTwo LiDAR sensor: the original sampling position and the x-axis (left-right) mirror of the same. Any other shift of the instances would destroy the structural integrity of the LiDAR point cloud and thus create a scene that the LiDAR sensor could not capture in reality.

The objects mentioned above are too few examples of important classes such as pedestrians and cyclists. By re-injecting the present instances in more frames we can alleviate this problem slightly but we risk overfitting on these few examples and diminishing the performance for unseen examples. In most object detection datasets, pedestrians and cyclists are present as classes, but the insertion of certain objects captured by another type of LiDAR sensors would not be injectable with the presented structure-preserving method without causing flaws in the structure of the resulting point cloud. Therefore we decided to generate more data in the structure of the *InnovizTwo* for these classes ourselves. Here we have considered the basic functionality and the recording mode of the sensor[3] and recreated it as a virtual twin. For this we re-engineered the origin of the sensor, the relative position of every single lidar measurement in the spherical coordinate system projection and the reliable detection range. We combine these information and re-formulate them as an orthogonal camera in the sphere coordinate system. With this virtual twin sensor, we recorded open-source 3D mesh models of pedestrians and cyclists in the spherical coordinate system. We re-transformed them onto the corresponding Cartesian coordinates and saved them in a second injection database. Since the resulting instance point clouds have the same properties as the original ones, we can insert the objects with their bounding boxes into the scenes while keeping the structure of the LiDAR sensor intact.

2.2 Scene Fusion

The small number of training scenes poses another problem: the neural networks can only use a limited number of scenes for negative examples, i.e. locations in the LiDAR frames that are not the target classes. A clear separation between foreground and background

objects is not easy with so little data. The more scenes we have available - whether full of objects or almost completely empty - the better our networks can define the difference between the desired classes and background objects that look very similar to them. We therefore implement the second augmentation method of [2], where two complete scenes of a given LiDAR sensor are compared point by point, and each possible position within a range image can be populated only by the respective closer points. Since we establish a depth-dependent relationship between two random point clouds, we can thus generate new frames that keep only partial aspects of the two "parent" point clouds. Of the 53 originally available frames we can therefore create $53^2 = 2{,}809$ by combining each point cloud with every other. With an additional x-axis mirroring of one of the two "parent" point clouds, we can even generate a total of $53^3 = 148{,}877$ training frames.

2.3 Pseudo Labels

After training a first *PV-RCNN* [1] network on the augmented data described above, we inferred this network on the unlabeled frames of the dataset. We use the validation set to define desired threshold scores for each class at different distance bins: We want to keep enough detections to bolster our training data without adding to many false positives. Detections above the given threshold are therefore taken as additional new ground truth. All detections below these scores are removed from the new pseudo data for the network. We remove not only the box label but also the corresponding LiDAR points inside the boxes in case we deleted a true positive. This simple data cleaning enables us to add the pseudo ground truth labels into our injection database and also to use the whole frames as additional scene fusion frames. This combination of ground truth and pseudo ground truth boosts the *mean Average Precision* of our final *PV-RCNN* by a considerable amount as can be seen in TABLE 3.

3 Evaluation

We used the standard metric of *mean Average Precision* (mAP) to evaluate the performance of the *PV-RCNN* models on the validation data. We compare the performance of the network trained solely on the original data, on the training data with our augmentations and finally the performance on the data with our augmentations enhanced with the proposed pseudo label addition. Each network is trained on a single *NVIDIA RTX 2080 Ti* for a total of 1000 epochs, a batch size of 1 and a one cycle learning rate scheduler with a maximum rate of 0.01. The resulting class wise *mAP* can be seen in TABLE 3. We take the average of a total of nine bins: the three range bins $0-50$ m, $50-100$ m, and >100 m and for each range bin the bins 0.3, 0.5, and 0.7 *Intersection over Union* (IoU). We also provide the metric of the *average Intersection over Union* (IoU) in the xy-coordinate plane (*AVG_XY_IOU*) on our choosen validation set as well as the two parts of the equation *GT_VS_DET* and *DET_VS_GT*. It is worth mentioning, that the *AVG_XY_IOU* of our final *PV-RCNN* trained with the augmentation methods and fused pseudo labels we have outlined above performs worse than the same network trained on the raw data. This is because the class *"cars"* is heavily over-represented in the validation set, and no class-wise evaluation is used. We focused on the mAP while developing our method, and only used the *AVG_XY_IOU* as a target metric for our final submission.

Method	mAP_{Car}	$\text{mAP}_{Ped.}$	$\text{mAP}_{Cyc.}$	mAP_{Truck}	AVG_XY_IOU	GT_VS_DET	DET_VS_GT
No Augmentations	**93.32**	0.00	33.33	**97.22**	**0.60**	0.49	**0.71**
Augmentations	91.47	8.33	43.06	86.14	0.50	0.69	0.31
Augs. + Pseudo Labels	90.41	**75.00**	**61.11**	92.96	0.51	**0.70**	0.31

Table 1: **Mean Average Precision per Class and total Average Intersection over Union over all classes.** We compare the performance of raw data to our augmentations, and the same with additional pseudo labels.

4 Submission

The evaluation metric of this challenge is *AVG_XY_IOU*, where for each annotation the best fitting prediction is taken into account (*AVG_GT_VS_DET_XY_IOU*) and vice versa (*AVG_DET_VS_GT_XY_IOU*). We decided to use an ensemble of multiple PV-RCNN models instead of a single one, and to train them for more epochs. We combine the detections of different voxel grid resolutions. We keep all proposed detections without any confidence threshold, to keep the hard and uncertain detections of pedestrians and cyclists. These three networks enable us to detect more of the ground truth instances than a single one would, but they also introduce more false positives. Therefore we extended our model ensemble by additional checkpoints for each of the three voxel grids, in which we raise the minimum confidence to very high values, in order to only keep the predictions that are very certain to be correct.

This ensemble enables us to detect very hard ground truth instances such as far away cyclists and pedestrians, while at the same time very confidently detecting easy targets, i.e. close cars. We are therefore able to raise the *AVG_GT_VS_DET_XY_IOU* two-fold: With our three networks, we find very difficult GT objects; with our additional confident networks we improve the IoU of the easy targets by offering multiple available predictions for the evaluation, since each GT cuboid is only compared to the best matching prediction.

By using multiple confident model outputs, we also raise the *AVG_DET_VS_GT_XY_IOU* term of the equation, since we have multiple predictions boxes especially for easy targets, that fit the GT cuboids relatively well and thus pull up the average which minimizes the influence of any false positives from our three non-threshold networks.

Thus we raise both parts of the equation for the final metric: our final results for this challenge has an *AVG_XY_IOU* score of 0.68.

5 Conclusion

From our experiments we conclude that our augmentation methods and novel fused pseudo labels improve the performance on the *InnovizTwo* dataset with a very limited amount of labeled data especially for underrepresented classes.

References

[1] Shi et al., S.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)

[2] Hasecke, F., Alsfasser, M., Kummert, A.: What can be seen is what you get: Structure aware point cloud augmentation. In: 2022 IEEE Intelligent Vehicles Symposium (IV). pp. 594–599. IEEE (2022)

[3] Innoviz: Innoviztwo website (2022), https://innoviz.tech/innoviztwo, accessed: 2022-09-05

[4] Innoviz, NVIDIA: Eccv workshop on 3d perception for autonomous driving: The lidar self-supervised learning challenge: Learning from a limited amount of high-resolution lidar data. https://innoviz.tech/eccv-challenge (2022), accessed: 2022-09-15

[5] Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet (2020)