DLAD Homework 3 Problem 2

Yunke Ao, Kaiyue Shen

July 4, 2021

Abstract

In this paper, we study the two-stage point-based 3D object detection problem, where we focus on improving the refinement network given the first stage RPN output. Our proposed refinement network improves the performance from three aspects: 1. model structure: learning global and local spatial feature by introducing MLP and canonical transformation; 2. loss function: including GIoU loss for regression task; 3. training scheme: change the optimizer from SGD to Adam. For evaluation metrics, we use mean average precision (mAP) of three difficulty levels: easy, moderate and hard. The ablation test proves the effectiveness of our three techniques. The combination of them performs the best among all difficulty levels.

1 Introduction

In autonomous driving, 3D object detection is one of the most important tasks. Based on the type of sensors we use to get the data, there are mainly two types of 3D object detection methods: one leverages relatively mature 2D detection algorithms using 2D images [1, 2], and the other takes in the 3D point clouds[4, 8, 5, 6]. The latter can be further divided into one-stage approaches and two-stage ones. Different from one-stage approach that directly produce the 3D bounding box, two-stage approaches first use Region Proposal Network (RPN) to generate coarse detection results and refine them in the second stage. In our work, as we are already given results of the first stage RPN alongside the extracted features and point cloud data, we focus on improving the performance of the refinement network.

The baseline method is done in a naive way with many limitations. For network inputs, it directly concatenates the point coordinates with the first stage decoder features, which, we think not fully makes use of the information in the point coordinates. For loss computations, it uses smooth-L1 loss for the regression task, that might not be the best choice considering that the evaluation metric is IoU-based. For training procedure, it uses SGD optimizer, which has proven to be less stable or converge slower than other modern optimizers.

Our contributions come down to the following three aspects:

- We apply MLP and canonical transformation to point coordinates to extract both global and local spatial features, together with decoder feature from RPN network to form a more informative feature input.
- We include an extra GIoU loss for the regression task, which is directly related to the evaluation metric we use and helps to improve the final result.
- We replace the SGD optimizer with Adam, which shows greater effect combined with two other newly-introduced modules.

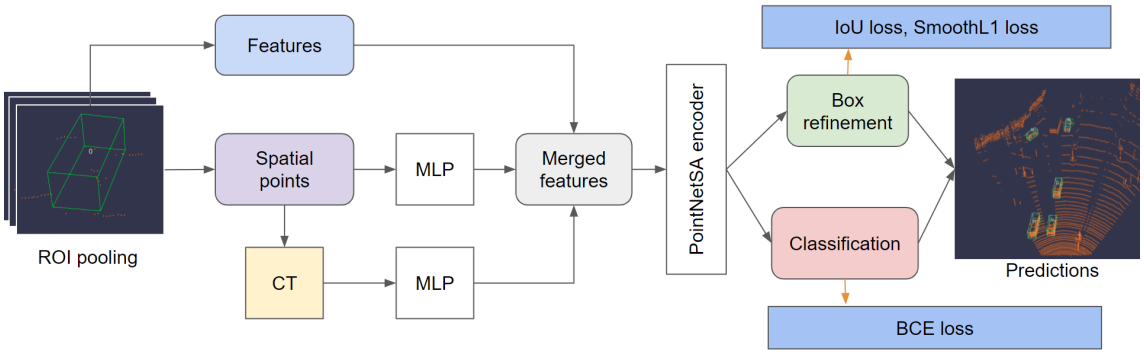

Figure 1: The architecture for the second-stage refinement network. We add two MLP to extract features from ROI for box refinement, and GIoU loss for better final performance.

2 Methods

In this section, we present our proposed second-stage framework for 3D object detection. The overall framework is shown in Figure 1, in which our main contribution includes feature extraction from spatial points for refinement, and adding GIoU loss for better performance.

2.1 Feature Learning for Box Refinement

In the baseline model, the input of the refinement model is obtained by simply concatenating the first stage decoder features with the point coordinates. The dimension of point coordinates is 3, which is much smaller than that of features, 128. These points are within the predicted bounding box in the first stage, so that they can provide the information of a rough location of the target bounding box. That's why we introduce the MLP module to increase the dimension of point coordinates.

Moreover, taking the idea from [4], we introduce the canonical transformation (CT) to transform the spatial points within each proposal from the LiDAR coordinate system to the canonical coordinate system of the corresponding 3D proposal. The canonical coordinate system is constructed with the center of the proposal as the origin, heading direction parallel and perpendicular to the X, Z axis. The advantage of such transformation is to provide better local spatial features for each proposal, which might help to predict more accurate size of the bounding box. Again, we use MLP to increase the dimension. One thing need to note here is that CT removes the depth information, so we append an extra distance feature, i.e., $d^{(p)} = \sqrt{(x^{(p)})^2 + (y^{(p)})^2 + (z^{(p)})^2}$ to the first stage decoder feature of point p. Finally, we concatenate decoder features, spatial features w/o CT, spatial features with CT, and apply a merge layer to get the merged features for the refinement network.

The advantage of our improvement is that all three sub-features provide useful information from different aspects. The decoder features encodes high-level information via learning for segmentation and proposal generation in the first stage. The spatial features without and with CT contain the global and local spatial information separately.

2.2 Including GIoU loss for Regression

In our baseline model, the loss function for box pose regression is the smooth-L1 loss for poses of boxes. However, the evaluation metrics for prediction during validation and testing are all based on IoU, rather than the euclidean distance between poses. This mismatch may cause limited improvement of testing performance even though there is significant decrease of training loss and validation loss. This phenomenon is also shown in our experiments in the next section. Therefore we also include GIoU loss [7, 3] in our loss function for box refinement.

For two shapes A and B, we use C to denote the smallest convex shape that enclosing both A and B. Then given the 3D IoU between A and B, GIoU is defined by:

$$GIoU = IoU - \frac{V_C - U_{AB}}{V_C} \tag{1}$$

where $U_{AB} = V_A + V_B - S_{overlap} \times h_{overlap}$ and the second item is a generalized metric of distance between A and B. With all this being defined, the loss we use for training is

$$L = L_{reg} + w \cdot L_{IoU} + L_{cls} \tag{2}$$

where $L_{IoU} = E[1 - GIoU(pred, target)]$ for all the prediction-target pairs with IoU larger than the threshold.

3 Results

3.1 Implementation Details

Network Architecture. Based on the provided baseline, we add two shared MLP for spatial points with and without CT, with size [3, 128, 128] and [4, 64, 64] respectively. Then the extracted features from points with or without CT are merged with the feature output from Region Proposal network (RPN) using a merge layer with size [320, 128]. Therefore the final feature input to the set abstraction modules also has the same dimension (128) as before.

The Training Scheme. We use Adam optimizer instead of SGD to train our network. The learning rate is 0.001, with step-wise decrease at milestones [5, 10, 15, 20, 25, 30] and decrease factor 0.5. The weight for GIoU loss is set as 0.2, and other hyperparameters are kept as default.

3.2 Performance on Provided Dataset

As is shown in Figure 2 and Table 1, the performance of our method surpass all other compared methods. The final validation results for easy, moderate and hard scenes are 84.449, 85.115 and 84.445 respectively, while the testing results are 77.63, 82.38 and 81.27 respectively. These demonstrate the effectiveness of our method.

3.3 Ablation Study

By comparing method 1 and method 2, it is shown in Figure 2 (d)-(e) that even though the method 2 results in much lower validation loss, the validation performance does not improve. This proves our statement of mismatch between loss and validation metrics in section 2. By comparing method 2 and method 3, it can also be seen that the Adam optimizer improves the performance a lot. After adding GIoU loss, method 4 outperforms method 3 for all different scenes, which proves that adding GIoU loss improves the performance.

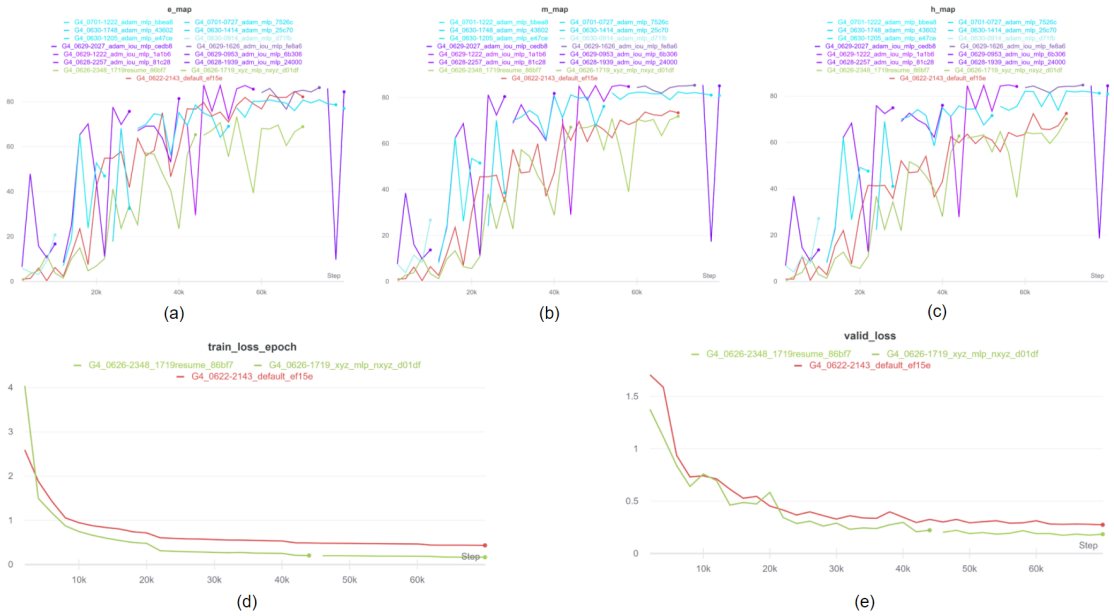

Figure 2: Learning curves for different combinations of methods. (a)-(c): Average precision performance for easy, moderate and hard scenes. (d)-(e): Training and validation loss curve for the methods 1 and 2. Red, green, blue and purple curves correspond to method 1-4 in Table 1.

Index	MLP	GIoU loss	Adam	AP_E	AP_M	AP_H
1	×	×	×	82.189	73.397	72.527
2	✓	×	×	66.595	71.866	70.098
3	✓	×	✓	77.048	80.885	80.907
4	✓	✓	✓	**84.449**	**85.115**	**84.445**

Table 1: Performance of different methods for ablation study. Our proposal surpass all the remaining combinations.

4 Discussion

We present an architecture that extract features from the local spatial points of ROI to encode more information for box refinement, and include GIoU loss to reduce the gap between loss and validation metrics. The whole model is trained using a more efficient optimizer and finally the performance is largely improved. However, due to the limited time and resources, we do not try higher weight for GIoU loss to see whether the performance could be further improved. Besides, the scheme to including both spatial features with and without CT could be further simplified, for example, by only include features of points with CT and also predict the box pose after CT, as is adopted by [4]. With these further improvement, we believe better performance could be achieved.

References

[1] Arsalan Mousavian et al. "3d bounding box estimation using deep learning and geometry". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 7074–7082.

[2] Charles R Qi et al. "Frustum pointnets for 3d object detection from rgb-d data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 918–927.

[3] Hamid Rezatofighi et al. "Generalized intersection over union: A metric and a loss for bounding box regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 658–666.

[4] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "Pointrcnn: 3d object proposal generation and detection from point cloud". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 770–779.

[5] Zetong Yang et al. "Ipod: Intensive point-based object detector for point cloud". In: *arXiv preprint arXiv:1812.05276* (2018).

[6] Zetong Yang et al. "Std: Sparse-to-dense 3d object detector for point cloud". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1951–1960.

[7] Dingfu Zhou et al. "Iou loss for 2d/3d object detection". In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 85–94.

[8] Yin Zhou and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4490–4499.