Towards Improving the Anti-attack Capability of the RangeNet++

Qingguo Zhou[1], Ming Lei[1], Peng Zhi[1], Rui Zhao[1], Jun Shen[2], and Binbin Yong[1](✉)

[1] School of Information Science and Engineering, Lanzhou University, Tianshui South Road
222, Lanzhou 730000, Gansu, China
yongbb@lzu.edu.cn

[2] School of Computing and Information Technology, University of Wollongong, Wollongong,
NSW, Australia

Abstract. With the possibility of deceiving deep learning models by appropriately modifying images verified, lots of researches on adversarial attacks and adversarial defenses have been carried out in academia. However, there is few research on adversarial attacks and adversarial defenses of point cloud semantic segmentation models, especially in the field of autonomous driving. The stability and robustness of point cloud semantic segmentation models are our primary concerns in this paper. Aiming at the point cloud segmentation model RangeNet++ in the field of autonomous driving, we propose novel approaches to improve the security and anti-attack capability of the RangeNet++ model. One is to calculate the local geometry that can reflect the surface shape of the point cloud based on the range image. The other is to obtain a general adversarial sample related only to the image itself and closer to the real world based on the range image, then add it into the training set for training. The experimental results show that the proposed approaches can effectively improve the RangeNet + +'s defense ability against adversarial attacks, and meanwhile enhance the RangeNet++ model's robustness.

Keywords: Adversarial attacks · Adversarial defenses · Semantic segmentation · RangeNet++ · Local geometry · Adversarial samples

1 Introduction

As part of the 3D point cloud scene understanding, semantic segmentation is a very important task in the field of autonomous driving. Semantic segmentation assigns a class label to each data point in the input modality, where a data point can be a pixel from a camera or a 3D point acquired from a radar. A stable semantic segmentation model can greatly improve the safety of autonomous driving systems. However, most semantic segmentation models perform poorly when facing the adversarial attacks and they have fairly weak adversarial defenses.

In recent years, there are many researches on applying adversarial attacks and adversarial defenses to various practical scenarios. Weng et al. [3] used adversarial attacks to find flaws in image QR codes. Wu et al. [4] summarized methods for image generation and

Y. Zheng et al. (Eds.): ACCV 2022, LNCS 13848, pp. 60–70, 2023.
https://doi.org/10.1007/978-3-031-27066-6_5

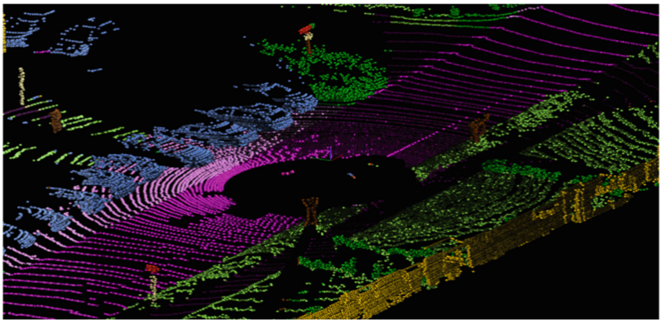

Fig. 1. Velodyne HDL-64E laser scan from KITTI dataset [1] with semantic information from RangeNet++. Best viewed in color, each color represents a different semantic class [2].

editing based on adversarial networks. Deng et al. [5] proposed a hippocampal segmentation method that combines residual attention mechanism with generative adversarial networks. However, there have been few studies on adversarial attacks and adversarial defenses of point cloud semantic segmentation models.

The essence of adversarial attack is to generate adversarial disturbances that can mislead model judgments and add them to samples. Therefore, how to generate aggressive and versatile adversarial samples is the focus of this research, while adversarial defense mainly focuses on improving the robustness of the model so that the model can still work normally in the face of disturbance. Among them, adding adversarial samples to the training set is a commonly used method for adversarial defense.

As one of the best point cloud semantic segmentation methods in the field of autonomous driving, the basic concept of RangeNet++ is to convert the point cloud into a range image, and then put it into a 2D convolutional network for learning, so as to obtain the label of the point. An example of RangeNet++ is shown in Fig. 1. However, the semantic information about the point cloud contained in the range image is too simple, which is not enough to support the model's good performance under adversarial attacks. Therefore, we can enrich the semantic information of the point cloud by obtaining the local geometric features of the points, so that the model has better robustness.

In this paper, we improve the security and anti-attack capability of the RangeNet++ model in two ways.

One is to add adversarial samples to the training set. We propose a method to obtain perturbation based on the surrounding local information of each point in the range image. Considering the fact that the semantic segmentation model classifies each point based on the feature differences between different objects, our method of generating adversarial samples calculates the average feature difference of each point and all points around it based on the range image. By adding the feature difference between each point and its surrounding points, the feature difference will become blurred, which makes it difficult for the semantic segmentation model to obtain the unique feature values of different objects and ensures the effectiveness of adversarial samples.

The other method is designed in such a way that instead of directly extracting local features for each point in the point cloud, we will calculate the difference between the normal vectors of different triangular patches formed by each point in the image and the surrounding points according to the range image to obtain the local geometry features. Such features reflect the topographic features of the 3D point cloud surface, thereby they will be increasing the semantic information of the point cloud.

In summary, contribution of our work are as follows:

i) A method for generating local features of point cloud based on range image and reflecting the geometric shape of point cloud.
ii) A method for generating general adversarial samples based on range image that are closer to the real world and adding them to the training set.
iii) Adversarial attack against RangeNet++ added to our method and original RangeNet++.

The rest of this paper is organized as follows. Section 2 presents a brief introduction to related works of local features of point clouds and adversarial examples. In Sect. 3, we describe the methodology of local feature extraction, and the generation of adversarial samples are also detailed. In Sect. 4, experimental results and the analysis are presented. Section 5 concludes this paper.

2 Related Work

2.1 Local Features of Point Clouds

Point cloud features are mainly divided into single point features, local features and global features. Single point features only focus on their own details, ignoring global information, and global features cannot describe the details sufficiently. In contrast, local features strike a balance between these two aspects.

Rusu et al. [6, 7] proposed to use point feature histograms (PFH) to obtain the shape information of the local surface. Using this method to describe point cloud features can extract rich information, but the computational complexity is high. It is also computationally expensive, especially for feature extraction tasks on large point cloud data.

In order to solve this problem, the Fast Point Feature Histograms (FPFH) method [8] reduces the computational complexity of the algorithm at the expense of reducing feature information by simplifying the interconnection characteristics of the PFH center point neighborhood, but it is simply affected by the point cloud sampling density. The feature distribution will be quite different when facing different sampling scales.

The Viewpoint Feature Histogram (VFH) [9] operator uses the angle between the viewpoint direction and the normal to the sampling point as a feature. However, VFH loses the property of rigidly maintaining the relative positional relationship between point clouds, and has poor performance in point cloud segmentation and registration.

Splash descriptor [10] describes local features by calculating the angle component between the normal vector of the key point and the normal vector of all points in the neighborhood.

Inspired by these works, we take the points in the range image as the key points, calculate the normal vectors of several triangular patches formed between itself and other key points in the neighborhood, and take the cosine value of any included angle between two normal vectors as the local feature of the point cloud. This feature can satisfactorily describe the surface shape of the point cloud in the neighborhood of the key points, better utilizing the morphological features of the point cloud.

2.2 Generation of Adversarial Examples

The key to adversarial sample generation is to find suitable and effective adversarial maneuvers. The fast gradient sign method (FGSM) proposed by Ian J et al. [11] maximizes the loss function to obtain the corresponding perturbation by backpropagating the gradient. Based on this, the DeepFool method proposed by Moosavi et al. [12] aimed at the selection of artificially specified perturbation coefficients in FGSM. This method firstly obtains good-performance adversarial samples by calculating the minimum necessary perturbation. Algorithms such as FGSM and DeepFool are all focusing on adversarial samples with different perturbation degrees for each adversarial sample generated by a single sample. In a work based on DeepFool, Moosavi et al. [13] extended it to propose a noise independent of the input image, known as Universal Adversarial Perturbations (UAP), which could destroy most natural images and would become a universal adversarial noise that could be trained offline and generate interference to the corresponding output of a given model online. These methods all rely heavily on the model to be attacked, and the adversarial disturbances generated for a certain model often perform poorly for other models and have poor scalability.

Xiang et al. [14] firstly proposed an adversarial sample generation method for 3D point clouds, and several new algorithms to generate adversarial point clouds for PointNet. However, on the one hand, the direct manipulation based on 3D point cloud will result in a large amount of computation. On the other hand, it is mainly used for adversarial sample generation of a single object and does not work well in point cloud segmentation where multiple objects exist.

On the basis of the above work, we herein propose a model-independent adversarial perturbation generation method for the semantic segmentation model. By calculating the difference between each point and its neighboring points in the image, the perturbation is generated to make the points in the image similar, so that the boundaries of each object become blurred in the process of semantic segmentation then the effective adversarial sample can be obtained.

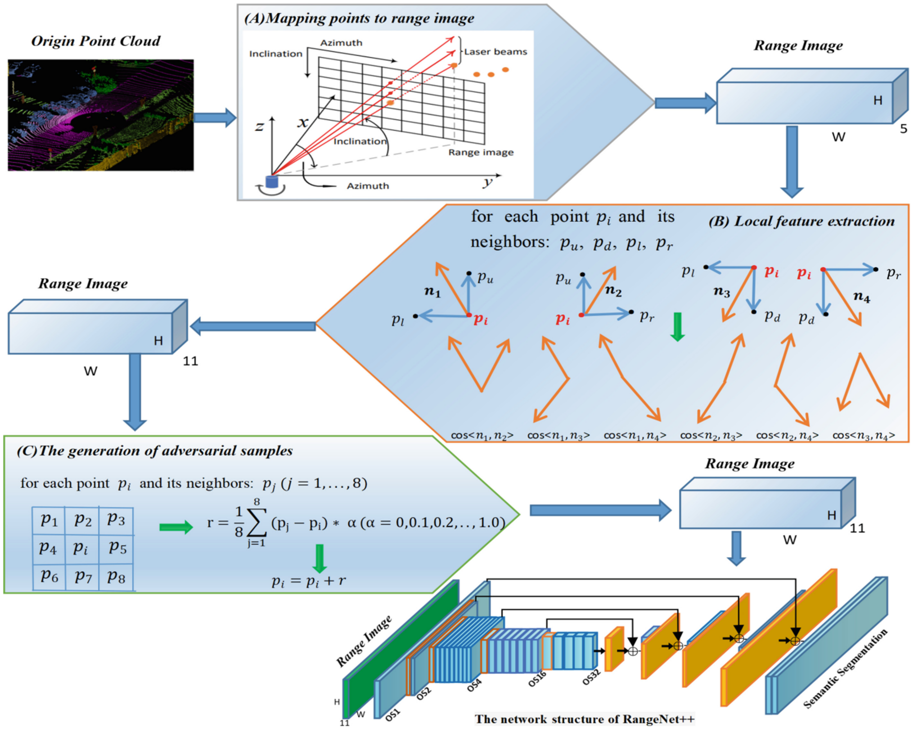

Fig. 2. The flow of our model. (A) convert the point cloud into a range image according to the method in RangeNet++, (B) extract local features based on the range image, (C) generate adversarial samples based on the range image, and put the new range image into the network of RangeNet++.

3 Methods

Our work is an improvement of the RangeNet++ model, which is carried out on the basis of range image, so we will first review the range image processing component in RangeNet++ in this section, and then introduce the local feature extraction method and adversarial samples generation method.

Our model is as follows: (A) convert the point cloud into a range image according to the method in RangeNet++; (B) extract local features based on the range image; (C) generate adversarial samples based on the range image, and then put the new range image into the network of RangeNet++. See Fig. 2 for details.

3.1 Mapping of Points to Range Image

For the points in the point cloud, the contained information of each of them includes coordinates *(x, y, z)* and remission. To obtain accurate semantic segmentation of point clouds, RangeNet++ converts each point cloud into a depth representation. To achieve this, it converts each point $p_i = (x, y, z)$ to spherical coordinates by mapping *Π: R^3*

→ *R^2,* and finally to image coordinates, we denote the height and width of the range image by *H, W,* while using *u, v* to represent the mapping of each point to the range image, then this mapping can be defined as:

$$\binom{u}{v} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\pi^{-1}\right]W \\ \left[1 - \left(\arcsin\left(zr^{-1}\right) + f_{down}\right)f^{-1}\right]H \end{pmatrix} \tag{1}$$

where, $r = \sqrt{x^2 + y^2 + z^2}$ represents the depth of each point, and $f = f_{up} + f_{down}$ is the vertical field-of-view of the sensor. With this approach, we get a list of *(u, v)* tuples containing a pair of image coordinates for each p_i. For each p_i, we extract its distance *r*, its *x*, *y*, and *z* coordinates, and its remission, and store them in the image, ultimately creating a *[5 × h × w]* tensor.

3.2 Local Feature Extraction

It should be noted that, after we obtain the mapping of each point on the range image, for the multiple points contained in each pixel, RangeNet++ selects the minimum depth point to represent this pixel, since there are multiple points mapped to the same pixel.

In order to ensure the proposed time complexity advantage of the RangeNet++ model, we did not choose to directly extract local features for each point in the point cloud, but relied on the corresponding points that their index could be used to easily find in the range image so as to quickly obtain its neighbor points, thus completing the extraction of local features.

Specifically, we first extract the neighborhood points of each point p_i in the range image in four directions: up, down, left, and right, namely p_u, p_d, p_l, p_r. With p_i and the four points around it, we can get four vectors:

$$\begin{cases} \boldsymbol{n_{ui}} = p_u - p_i \\ \boldsymbol{n_{di}} = p_d - p_i \\ \boldsymbol{n_{li}} = p_l - p_i \\ \boldsymbol{n_{ri}} = p_r - p_i \end{cases} \tag{2}$$

Then, we use "×" to denote the Vector product. Meanwhile, we can get four planes, and each of which includes p_i and two non-collinear neighbor points: (p_i, p_l, p_u), (p_i, p_r, p_u), (p_i, p_l, p_d), (p_i, p_r, p_d). We use their normal vectors to represent them:

$$\begin{cases} \boldsymbol{n}_1 = \boldsymbol{n_{li}} \times \boldsymbol{n_{ui}} \\ \boldsymbol{n}_2 = \boldsymbol{n_{ri}} \times \boldsymbol{n_{ui}} \\ \boldsymbol{n}_3 = \boldsymbol{n_{li}} \times \boldsymbol{n_{di}} \\ \boldsymbol{n}_4 = \boldsymbol{n_{ri}} \times \boldsymbol{n_{di}} \end{cases} \tag{3}$$

$\boldsymbol{n}_1$, $\boldsymbol{n}_2$, $\boldsymbol{n}_3$, $\boldsymbol{n}_4$ represent the directions of the four planes. Then, we calculate the angle between these normal vectors, here we use the cosine of the angle to represent the angle:

$$\begin{cases} \theta_1 = cos < \boldsymbol{n}_1, \boldsymbol{n}_2 > \\ \theta_2 = cos < \boldsymbol{n}_1, \boldsymbol{n}_3 > \\ \theta_3 = cos < \boldsymbol{n}_1, \boldsymbol{n}_4 > \\ \theta_4 = cos < \boldsymbol{n}_2, \boldsymbol{n}_3 > \\ \theta_5 = cos < \boldsymbol{n}_2, \boldsymbol{n}_3 > \\ \theta_6 = cos < \boldsymbol{n}_3, \boldsymbol{n}_4 > \end{cases} \tag{4}$$

Finally, we add these 6 values to the *[5, H, W]* tensor created in the previous section to get a *[11, H, W]* tensor and put it into the RangeNet++ semantic segmentation convolutional network for learning.

3.3 The Generation of Adversarial Samples

Our work on adversarial sample generation is based on range image. The basic concept is as follows: firstly, obtain the difference between each point in the range image and each point in the surrounding neighborhood, and then multiply it by a perturbation parameter, which determines the strength of the adversarial attack, so as to obtain an adversarial perturbation, and generate adversarial samples.

For the semantic segmentation model, its essence is to find the feature differences between different objects. The smaller the differences are, the more blurred the different objects become, and the semantic segmentation model is more difficult to play a role.

Therefore, in the first step, we need to calculate the difference between each point p_i in the range image and its surrounding points $p_j (j = 1, 2, \ldots, 8)$:

$$d_j = p_j - p_i \tag{5}$$

When getting the difference value from each surrounding point, we average these difference values to generate the average difference value $\overline{d_i}$:

$$\overline{d_i} = \frac{1}{8} \sum\nolimits_{j=1}^{8} d_j \tag{6}$$

$\overline{d_i}$ can be used as an adversarial perturbation for the semantic segmentation model.

In order to ensure that the adversarial samples are effective, we must have some control over the adversarial perturbation, so that it cannot be observed. Therefore, we set the perturbation parameter α, which is in the range [0.0, 1.0]. For the difference value obtained in Eq. (6), we multiply it by α as the final perturbation value. Then, we set the perturbation parameter α in the range [0.0, 1.0]. For the difference value obtained in Eq. (6), we multiply it by α as the final perturbation value:

$$r_i = \alpha * \overline{d_i} \tag{7}$$

Finally, the perturbation generated by each point is added to the sample, and the corresponding adversarial sample is obtained.

3.4 Adversarial Attacks Against Models

In the previous subsection, we obtained adversarial examples based on range image. Using these adversarial examples, we design adversarial attacks against RangeNet++ and our model. The adversarial attack mainly consists of two parts. The difference between the two parts lies in the selection of the perturbation parameters α. In the first part, for each range image, we randomly select the perturbation parameters in the range of [0, 1] to achieve a more comprehensive attack. In the second part, we set the same perturbation parameters to conduct a more targeted attack on the model.

Table 1. IoU [%] before and after adversarial attack.

Approach	RangeNet++			RangeNet++ + Local Feature			RangeNet++ + Local Feature + Adversarial Samples		
State	Before	After	Diff	Before	After	Diff	Before	After	Diff
Car	80.2	60.1	**20.1**	79.1	67.1	**12.0**	80.2	79.8	**0.4**
Bicycle	16.4	7.5	**8.9**	15.9	7.2	**8.7**	16.6	12.4	**4.2**
Motorcycle	34.5	10.6	**23.9**	34.2	20.3	**13.9**	30.6	29.8	**0.8**
Truck	3.5	1.7	**1.8**	2.0	0.9	**1.1**	4.8	3.5	**1.3**
Other-vehicle	21.3	3.7	**17.6**	22.0	11.4	**10.6**	21.0	20.6	**0.4**
Person	15.1	10.8	**4.3**	12.3	8.8	**3.5**	13.9	14.1	**−0.2**
Bicyclist	35.4	16.8	**18.6**	33.8	25.5	**8.3**	33.9	33.6	**0.3**
Motorcyclist	0.0	0.0	**0.0**	0.0	0.0	**0.0**	0.0	0.0	**0.0**
Road	89.8	73.0	**16.8**	91.0	77.1	**13.9**	90.1	90.3	**−0.2**
Parking	49.7	27.7	**22.0**	52.0	28.1	**23.9**	48.9	48.7	**0.2**
Sidewalk	76.2	49.6	**26.6**	77.9	54.7	**23.2**	76.6	76.9	**−0.3**
Other-ground	0.0	0.1	**−0.1**	0.1	0.5	**−0.4**	0.0	0.0	**0.0**
Building	76.5	60.9	**15.6**	77.2	73.0	**4.2**	77.1	77.0	**0.1**
Fence	46.1	22.4	**23.7**	44.6	25.8	**18.8**	48.7	48.5	**0.2**
Vegetation	75.0	50.6	**24.4**	76.0	63.4	**12.6**	75.9	76.0	**−0.1**
Trunk	37.8	34.3	**3.5**	39.5	37.3	**2.2**	32.0	32.6	**−0.6**
Terrain	61.5	38.5	**23.0**	64.9	45.5	**19.4**	64.1	64.1	**0.0**
Pole	27.6	24.3	**3.3**	31.6	28.5	**3.1**	29.6	29.2	**0.4**
Traffic-sign	27.2	20.7	**6.5**	29.4	26.3	**3.1**	25.3	25.2	**0.1**
MeanIoU	40.7	27.0	**13.7**	41.2	31.7	**9.5**	40.5	40.1	**0.4**
MeanAcc	83.4	67.7	**16.6**	85.0	74.5	**10.5**	84.9	84.9	**0.0**

4 Experiments and Results

Dataset: We use SemanticKITTI as experiment dataset, which is a sub-dataset of KITTI in the direction of semantic segmentation. It annotates all sequences in the KITTI Vision Odometry Benchmark and provides dense point-by-point annotations for the full 360° field of view of the automotive LiDAR used in [2]. It includes a total of 22 sequences, of which sequences 00-10 are the training set and sequences 11-21 are the test set.

Parameters: To get objective and fair results, we train RangeNet++ and our model with the same parameters. We choose the sequences 00-05 as the training set, 07-10 as the test set, and 06 as the validation set. Meanwhile, we train each model for 100 epochs. We employ random perturbation parameters in the first experiment, i.e. $\alpha \in [0.0, 1.0]$, and specific perturbation parameters in the second experiment, i.e. $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

To ensure consistency, we chose the same evaluation metric as in RangeNet++. That is, we use the mean intersection over union (mIoU) over all classes [6], with the following formula:

$$mIoU = \frac{1}{D}\sum\nolimits_{d=1}^{D} \frac{TP_d}{TP_d + FP_d + FN_d} \tag{8}$$

where TP_d, FP_d and FN_d correspond to the number of true positive, false positive, and false negative predictions for class d and D is the number of classes.

4.1 Adversarial Attacks with Randomly Perturbed Parameters

The first experiment aims to demonstrate the effectiveness of the two methods proposed in this paper, that is, to improve the security and anti-attack capability of RangeNet++ in the face of adversarial attacks. Therefore, we compare the defense ability of the original RangeNet++ with the RangeNet++ with only local features added and the model containing local features and adversarial samples in the face of adversarial attacks.

Table 1 shows the changes of the IoU of these three models before and after being attacked. As can be seen from Table 1, compared to the original RangeNet++, the RangeNet++ added to our method performs better in the face of adversarial attacks. More specifically, the RangeNet++ that only contains local features is improved compared to the original RangeNet++ and the RangeNet++ that includes both methods have a higher improvement, which fully shows that our two methods are both effective in improving the defense capability of the RangeNet++.

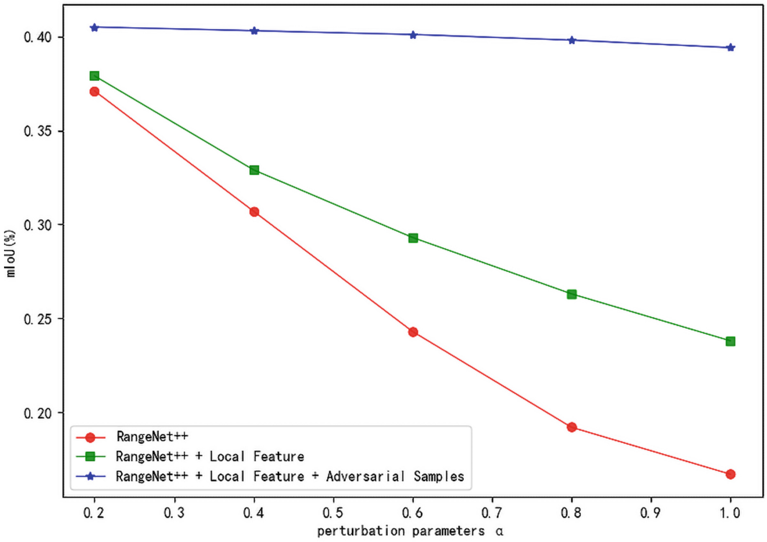

Fig. 3. Experimental results: mIoU with different perturbation parameters.

4.2 Adversarial Attack with Specified Perturbation Parameters

The second experiment demonstrates the effectiveness of adversarial attacks based on the adversarial samples proposed in this paper. Figure 3 shows the variation of mIoU for the three models under different parameters, which represent adversarial attacks of different strengths.

As can be seen from Fig. 3, as the perturbation parameter increases, the mIoU of the three models gradually decreases, which proves the effectiveness of adversarial samples. At the same time, we can find that our method has better performance in the face of different levels of adversarial attacks, which further illustrates that our method can improve the defense ability of the RangeNet++.

5 Conclusions

In this work, we proposed two methods to improve the security and anti-attack capability of the semantic segmentation model RangeNet++. One was to extract the local geometric features of each point reflecting the topographic features of the point cloud based on the range image. The other was to generate adversarial examples against the semantic segmentation model based on the range image. And then we put them into the training set. Through experiments, we demonstrated the effectiveness of these two methods. Moreover, the adversarial attack based on our designed adversarial samples also had good performance for the RangeNet++ model.

Acknowledgments. This work was partially supported by the Gansu Provincial Science and Technology Major Special Innovation Consortium Project (No. 21ZD3GA002), the name of the innovation consortium is Gansu Province Green and Smart Highway Transportation Innovation

Consortium, and the project name is Gansu Province Green and Smart Highway Key Technology Research and Demonstration.

References

1. Behley, J., et al.: SemanticKITTI: a dataset for semantic scene understanding of LiDAR sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
2. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: RangeNet++: fast and accurate LiDAR semantic segmentation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2019)
3. Weng, H., et al.: Towards understanding the security of modern image captchas and underground captcha-solving services. Big Data Mining Anal. **2**(2), 118–144 (2019)
4. Wu, X., Xu, K., Hall, P.: A survey of image synthesis and editing with generative adversarial networks. Tsinghua Sci. Technol. **22**(6), 660–674 (2017)
5. Deng, H., Zhang, Y., Li, R., Hu, C., Feng, Z., Li, H.: Combining residual attention mechanisms and generative adversarial networks for hippocampus segmentation. Tsinghua Sci. Technol. **27**(1), 68–78 (2022)
6. Rusu, R.B., Marton, Z.C., Blodow, N., et al.: Learning informative point classes for the acquisition of object model maps. In: 2008 10th International Conference on Control, Automation, Robotics and Vision, pp. 643–650. IEEE (2008)
7. Rusu, R.B., Blodow, N., Marton, Z.C., et al.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3384–3391. IEEE (2008)
8. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217. IEEE (2009)
9. Rusu, R.B., Bradski, G., Thibaux, R., et al.: Fast 3D recognition and pose using the viewpoint feature histogram. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2155–2162. IEEE (2010)
10. Stein, F., Medioni, G.: Structural indexing: efficient 3-D object recognition. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 125–145 (1992)
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations, California (2015)
12. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Nevada, pp. 2574–2582. IEEE (2016)
13. Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, pp. 86–94. IEEE (2017)
14. Xiang, C., Qi, C.R., Li, B.: Generating 3D adversarial point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9128–9136 (2019). https://doi.org/10.1109/CVPR.2019.00935