ADVERSARY DISTILLATION FOR ONE-SHOT ATTACKS ON 3D TARGET TRACKING

Zhengyi Wang[a], Xupeng Wang[a], Ferdous Sohel[b], Mohammed Bennamoun[c], Yong Liao[a], Jiali Yu[a]*

[a]University of Electronic Science and Technology of China
[b]Murdoch University
[c]The University of Western Australia

ABSTRACT

Considering the vulnerability of existing deep models in the adversarial scenario, the robustness of 3D target tracking is not guaranteed. In this paper, we present an efficient generation based adversarial attack, termed Adversary Distillation Network (AD-Net), which is able to distract a victim tracker in a single shot. In contrast to existing adversarial attacks derived from point perturbations, the proposed method designs a generative network to distill an adversarial example from a tracking template through point-wise filtration. A binary distribution encoding layer is specialized to filter points, which is modeled as a Bernoulli distribution and approximated in a differentiable formulation. To boost the performance of adversarial example generation, a feature extraction module is deployed, which leverages the PointNet++ architecture to learn hierarchical features for the template points as well as similarities with the search areas. Experimental results on the KITTI vision benchmark show that the proposed adversarial attack can effectively mislead popular deep 3D trackers.

Index Terms— 3D target tracking, adversarial attack, adversary distillation

1. INTRODUCTION

Despite a range of works in autonomous driving and intelligent surveillance systems, the task of 3D target tracking has received increasing attentions. While many breakthroughs have been made [1, 2], the dependabilities of 3D deep trackers is ignored, compared to its 2D counterpart [3]. Existing works demonstrate that deep models are vulnerable to adversarial examples, which are maliciously crafted with imperceptible modifications [4, 5, 6]. Considering the vital role that 3D target tracking plays in a variety of security-critical applications, efforts made on the robustness evaluation of deep 3D trackers are in great demand.

Early studies of adversarial attacks on deep 3D models were mainly devoted to classifiers. According to the transformations applied to the victim point cloud, attacks can be classified into point perturbation based method and point drop based method. Adversarial attacks derived from point perturbations are typically constrained with L2 normalization, leading to a failure of the classifier prediction with points moved locally [4, 5, 7]. These methods generate instance-level adversarial attacks, which are based on time-consuming optimizations online and are not effective in the scenario with a real-time requirement. A flexible targeted attack was proposed in [8], which designed an adversarial network guided by labels to mislead a classifier in a single forward pass. In [9], a generative network was developed, focusing on generating adversarial attacks with a strong transferability across victim deep models. On the other hand, point drop is developed as a method to generate adversarial examples, simulating occlusion or inherent defects of point cloud data captured by 3D sensors. In [10], salient points of a point cloud were calculated as their contributions to the classification, which were abandoned to generate an adversarial example. Point-wise robustness of deep 3D classifiers was analyzed in [11], where an iterative salience occlusion algorithm was presented to verify the robustness of neural networks.

Adversarial attacks on 3D object detection have also been investigated. The first attack on LiDAR-oriented deep detectors was proposed in [12], which combined optimization and global sampling to launch attacks. In [13], a method to generate universal adversarial attacks against 3D detectors was developed from point cloud addition, which has the advantage of physically realizability in the autonomous driving scenario. In addition, spoofing attacks were develop in [14] based on the observation that the presence of occlusion in LiDAR point clouds leads to the vulnerability of 3D detectors.

In this paper, we present a generation based method to launch one-shot adversarial attacks on 3D target tracking, called Adversary Distillation Network (AD-Net). Specifically, the generative network constitutes two sub-networks, namely the feature extraction module and the point filtration module. The feature extraction module exploits PointNet++ [15, 16] to learn hierarchical features of the tracking template, fused with its similarity to the search areas. The encoded similarity diverges the extracted features of the target template and the search areas, leading a distraction to the victim

This work was supported by National Natural Science Foundation of China [grant number 61572112], the Applied Basic Research Programs of Science and Technology Department in Sichuan Province [grant number 2019YJ0185]

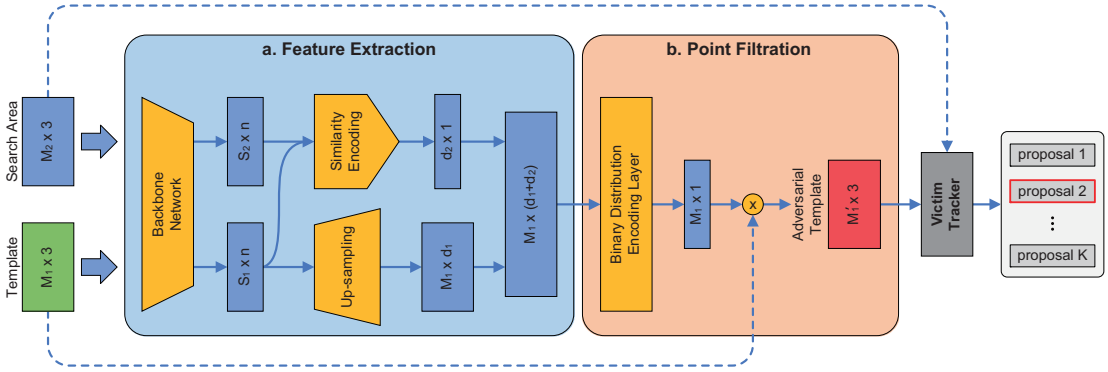


Fig. 1. Illustration of the proposed Adversary Distillation Network (AD-Net) which is made up of a feature extraction module and a point filtration module. The former extracts geometric features from a tracking template, according to which the latter filters points to generate the adversarial counterpart of the tracking template. (a) Given a tracking template, the PointNet++ is adopted as the backbone network to learn hierarchical features for each point, which are further enhanced by encoding the similarity of the target between the template and the search area. (b) The point filtration module leverages a binary distribution encoding layer to filter points of the template and to generate an adversarial template. The victim tracker, fed with the adversarial template, is distracted to produce a prediction with a region proposal deviated from the actual location. (Best viewed in color.)

tracker. The point filtration module is constructed on the basis of a binary distribution encoding layer, and performs adversarial template distillation with respect to the extracted features.

To sum up, the contributions of this work are as follows:

- A generative adversarial attack is proposed to distract 3D target tracking in a single shot.
- A generative network is designed to distill an adversarial example from the tracking template guided both by the feature loss and location loss.
- A binary distribution encoding layer is developed to perform point filtration by approximating the Bernoulli distribution in a differentiable formulation.
- Experimental results show that the proposed method can fool popular deep 3D trackers on the KITTI vision benchmark in an efficient way.

2. METHOD

Given a video sequence $S = \{s_i\}_{i=1}^{N}$ of N frames represented as point clouds, 3D target tracking aims to locate a target in consecutive frames with respect to a tracking template P_{tmp}. Typically, the template P_{tmp} is manually derived from the first frame s_1, satisfying $P_{tmp} \in s_1$. A 3D tracker generates a search area P_{area} for each frame by enlarging the scale of its previous prediction, and produces a number of proposals associated with probability scores as candidate target locations. In an adversarial setting, the goal of adversarial attacks on 3D target tracking is to distract a victim tracker and provide a proposal deviated from the actual location as the final prediction.

Our proposed method of adversarial attacks designed for the 3D target tracking scenario is illustrated in Fig. 1. Given a tracking template P_{tmp} with M_1 points, the feature learning module adopts the PointNet++ architecture as the backbone network to learn geometric features for its S_1 seed points, which are upsampled for each template point afterwards. Similarities between the target template P_{tmp} and the search area P_{area} are further encoded, resulting in a feature description of the template with dimension $M_1 \times (d_1 + d_2)$. The point filtration module leverages the binary distribution encoding layer to learn point-wise Bernoulli distribution, from which an adversarial template $\hat{P}_{tmp}$ is distilled in a single shot.

2.1. Feature Extraction

In feature extraction module, the local geometric features of the template P_{tmp} are aggregated by PointNet++. Specifically, the template P_{tmp} and the search area P_{area} are downsampled into S_1 and S_2 seed points with multi-scale features $f \in \mathbb{R}^n$ by the farthest point sampling algorithm, which denoting as S_{tmp} and S_{area} respectively. Whereafter, the template seed points S_{tmp} is upsampled into $P'_{tmp} \in \mathbb{R}^{M_1 \times d_1}$ to capture latent features for each point in the original template.

For improved feature extraction, a branch is designed to obtain the latent similarity by calculating the cosine distance between the seed points S_{tmp} and S_{area}, which termed as $Sim \in \mathbb{R}^{S_2 \times S_1}$. The encoded latent similarity can fuse the template P_{tmp} and the search area P_{area} in an efficient way. Because of the permutation invariance of the point cloud, we further apply a symmetric function to guarantee a same latent similarity represented as $Sim' \in \mathbb{R}^{d_2 \times 1}$. At last, the enhanced feature of the template P_{tmp} can be represented as $P''_{tmp} \in \mathbb{R}^{M_1 \times (d_1 + d_2)}$ by concatenating with the encoded

similarity of repeated M_1 times.

The latent similarity is decreased as the feature loss to differ the template P_{tmp} and the search area P_{area} in an underlying space, as follows:

$$\mathcal{L}_{feat} = \frac{1}{\|Sim'\|_0} \sum_{i=1}^{d_2} Sim'_i \tag{1}$$

where Sim' is the encoded similarity between the template P_{tmp} and the search area P_{area}.

2.2. Point Filtration

The point filtration module learns the probability of each point to be filtered out, simulating occlusions on the surface of adversarial template to fool 3D deep tracker in one-shot. A binary distribution encoding layer is designed to achieve the point filtering procedure by point-wise filtration. The binary distribution encoding layer learns a Bernoulli distribution to describe the filtering state for each point. Specifically, point-wise filtering states $z_i \in \{0, 1\}$ are attached with each point of the template, representing as $\bar{P}_{tmp} = \{z_i \times x_i\}_{i=1}^{M_1}$. The points whose filtering state $z_i = 0$ are filtered away to generate the adversarial template $\hat{P}_{tmp} \in \mathbb{R}^{M'_1 \times 3}$.

However, the Bernoulli distribution is computationally intractable for its non-differentiability. The binary concrete distribution [17] is a smooth simulation of the Bernoulli distribution, of which is a differentiable formulation of approximation. Given a random variable s satisfies a binary concrete distribution φ lying in the $(0, 1)$ interval, denoting $q_s(s|\phi)$ as the probability density and $Q_s(s|\phi)$ as the cumulative probability. The binary concrete distribution φ can be parameterized by $\phi = (\log \alpha, \beta)$, noting that $\log \alpha$ represents the location while β is the temperature. With a uniform distribution $u \sim U(0, 1)$, the binary concrete distribution can be reparameterized as:

$$s = \text{Sigmoid}((\log u - \log(1 - u) + \log \alpha)/\beta). \tag{2}$$

To ensure the point filtration module can filter points efficient, taking value exactly zero or one is imperative. The binary concrete distribution is stretched to the (γ, ζ) interval, where $\gamma < 0$ and $\zeta > 1$, and then make it a hard concrete distribution [18] by implementing a hard-sigmoid function:

$$\bar{s} = s \times (\zeta - \gamma) + \gamma \tag{3}$$

$$z = \min\{1, \max\{\bar{s}, 0\}\}. \tag{4}$$

For the L_0 regularization induces no shrinkage on the actual values of the filtering states, it is utilized to penalize the binary distillation encoding layer. The binary distribution encoding layer is penalized by the L_0 regularization to minimize the number of points filtered away. The cumulative probability of the hard concrete distribution at zero is defined as the L_0 normalization, which is formulated as:

$$\mathcal{L}_0 = p(\bar{s} \neq 0) = \text{Sigmoid}(\log \alpha - \beta \times \log(-\gamma/\zeta)). \tag{5}$$

2.3. Loss Function

For the invisibility of the attack to human eyes, L_2 distance is utilized to constrain the modification, which is defined as:

$$\mathcal{L}_{perc} = \mathcal{D}_{L_2}(\hat{P}_{tmp}, P_{tmp}) = (\sum_i (\hat{x}_i - x_i)^2)^{\frac{1}{2}}. \tag{6}$$

A deep tracker predicts a number of proposals combined with confidence scores as the candidate target locations, and the proposal with the highest score is chosen as the final prediction. However, other high-scoring proposals are also spotted near the actual location, allowing the predictions to remain accurate. Location loss decreases confidences simultaneously by gathering proposals as groups, which is drafted as:

$$\mathcal{L}_{loc} = \max\{\sum_{i=0}^{p} \mathcal{R}_i - \sum_{i=q}^{r} \mathcal{R}_i, 0\} \tag{7}$$

where $\mathcal{R}$ represents the ranked confidences of proposals and p, q, r are the index ranges of proposals to be grouped.

Thus, the objective loss function can be formulated as:

$$\mathcal{L} = \mathcal{L}_{feat} + \mathrm{a} * \mathcal{L}_{loc} + \mathrm{b} * \mathcal{L}_0 + \mathrm{c} * \mathcal{L}_{perc}. \tag{8}$$

Here $\mathrm{a}(= 0.5)$, $\mathrm{b}(= 0.3)$ and $\mathrm{c}(= 0.1)$ are the hyper parameters to balance the items in the loss function.

3. EXPERIMENT

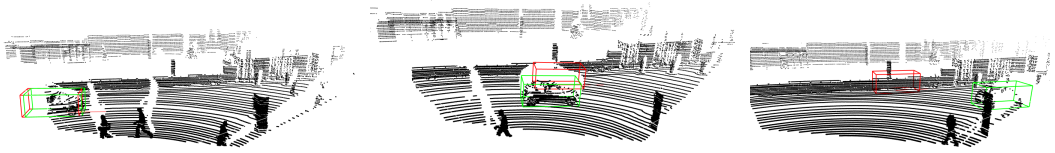
Our attack approach against the tracker was implemented on KITTI [19] vision benchmark. The 3D object tracking model P2B was the victim model in experiments. The parameters of the hard concrete distribution were set as $\beta = 0.6$, $\gamma = -0.1$ and $\zeta = 1.1$. A desktop with a CPU (Intel Core-i5) and a GPU(NVIDIA GTX-1080) supported all the experiments.

3.1. Dataset

The KITTI dataset contains 21 video sequences for training and 29 for testing, labeled with 21 categories. Because the label of the test set is inaccessible, we only used the training set to train and test. We split the dataset as: 0-16 for training, 17-18 for validation and 19-20 for testing. Tracklets for target objects were all generated within all sequences.

3.2. Evaluation Metrics

We use Success and Precision as the metrics to evaluate the effectiveness of our attack. Success is calculated as the average IOU of the bounding box between the prediction and the ground truth. In our context a lower value of Success and Precision will represent better results.



(a) Frame 2 (b) Frame 20 (c) Frame 40

Fig. 2. Qualitative evaluation for the proposed adversary distillation attack on a video sequence of the KITTI dataset, in which a car drives from left to right. The P2B tracker was used for evaluation. The tracking result deviated further from the actual location, where the green box is the ground truth and the red one is the tracking result after our attack.

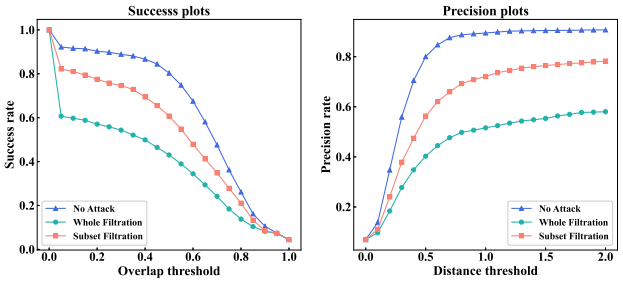


Fig. 3. Evaluation results of the tracker with or without adversarial attack on KITTI dataset.

3.3. Comparison With State-of-the-art

The qualitative evaluation of the proposed attack method is shown in Fig. 2. The ground truth of the target car is colored green, while the prediction of the attacked tracker is red. The prediction deviates further with time, reflecting proposed adversarial attack method is efficient to distract the deep tracker.

The comparison results are shown in Table 1. Notes the proposed method reduces the average success and the average precision by 22.9% and 31.0% respectively, while filtering the seed points resulting only 44.9% and 57.1% to the success and precision. The former attack results in an overall better outcome than the latter, indicating that our attack method can acquire global features effectively to filter the key points.

It can be seen from Fig. 3 that there are 8% proposals whose IOU is less than 0.05 predicted from the original tracker, however our method significantly improves the ratio to 39%. In addition, the percentage of proposals to be less than half a meter away from the actual center of the ground truth drops from 81% to 40%, indicating more predictions locate far from the actual region.

3.4. Ablation Study

We evaluate the contribution of the similarity encoding branch, which is designed to enhance the feature extraction, through comparing the results under situations as: with concatenation (the default setting), without the concatenation and without the whole branch. In the absence of the concatenation, the similarity is only used to calculate the loss to guide the optimization. The ablation comparison results are shown in Table 2. The results show that the similarity encoding branch can play a good role in reducing the success and precision of the tracker by 21.2% and 28.8% respectively in the case of without concatenation. After concatenating, the success and precision are further reduced to our best results while they only decrease by 5.5% and 6.9% without the similarity encoding branch.

Table 1. Comparison of the results from the original tracker, the tracker attacked by filtering the point subset and the entire template point cloud. (Lower value represents better result)

Point set to be filtered	Success(%)	Precision(%)
No attack	53.3	68.4
Seed points of template	44.9	57.1
The whole template	**30.4**	**37.4**

Table 2. Ablation studies of the similarity branch to explore the contribution. (Lower value represents better result)

Similarity encoder	Success(%)	Precision(%)
Our default setting	**30.4**	**37.4**
Without concatenation	32.1	39.6
Without the similarity encoder	47.8	61.5

4. CONCLUSION

In this paper, we propose a generative method to launch spoofing attacks on deep 3D trackers, which generates adversarial examples with adversary distillation. We focus on one-shot attacks that solely modify the tracking template, which guide subsequent predictions deviated from the actual locations. A binary distribution encoding layer is designed to perform point-wise filtration. A feature extraction module learns hierarchical features from the tracking template and its dissimilarities with the search areas to boost adversarial example generation. Experimental results show the effectiveness of our attack method against popular deep 3D trackers.

5. REFERENCES

[1] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2B: point-to-box network for 3D object tracking in point clouds," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, Jun.*, 2020, pp. 6328–6337.

[2] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem, "Leveraging shape completion for 3D siamese tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, Jun.*, 2019, pp. 1359–1368.

[3] X. Chen, X. Yan, F. Zheng, Y. Jiang, S. Xia, Y. Zhao, and R. Ji, "One-shot adversarial attacks on visual tracking with dual attention," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, Jun.*, 2020, pp. 10173–10182.

[4] Y. Wen, J. Lin, K. Chen, C. L. P. Chen, and K. Jia, "Geometry-aware generation of adversarial point clouds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[5] Chong Xiang, Charles R. Qi, and Bo Li, "Generating 3D adversarial point clouds," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, Jun.*, 2019, pp. 9136–9144.

[6] Xupeng Wang, Mumuxin Cai, Ferdous Sohel, Nan Sang, and Zhengwei Chang, "Adversarial point cloud perturbations against 3D object detection in autonomous driving systems," *Neurocomputing*, 2021.

[7] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin, "Robust adversarial objects against deep learning models," in *Proc. Conference on Artificial Intelligence, New York, USA, Feb.*, 2020, pp. 954–962.

[8] Hang Zhou, Dongdong Chen, Jing Liao, Kejiang Chen, Xiaoyi Dong, Kunlin Liu, Weiming Zhang, Gang Hua, and Nenghai Yu, "LG-GAN: label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, Jun.*, 2020, pp. 10353–10362.

[9] Abdullah Hamdi, Sara Rojas, Ali K. Thabet, and Bernard Ghanem, "Advpc: Transferable adversarial perturbations on 3D point clouds," in *Proc. European Conference on Computer Vision, Glasgow, UK, Aug.*, 2020, pp. 241–257.

[10] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren, "Pointcloud saliency maps," in *Proc. IEEE International Conference on Computer Vision, Seoul, Korea (South), Oct.*, 2019, pp. 1598–1606.

[11] Matthew Wicker and Marta Kwiatkowska, "Robustness of 3D deep learning in an adversarial setting," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, Jun.*, 2019, pp. 11767–11775.

[12] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *Proc. ACM SIGSAC Conference on Computer and Communications Security, London, UK, Nov.*, 2019, pp. 2267–2281.

[13] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun, "Physically realizable adversarial examples for lidar object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, Jun.*, 2020, pp. 13713–13722.

[14] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao, "Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *Proc. USENIX Security Symposium, Aug.*, 2020, pp. 877–894.

[15] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, Jul.*, 2017, pp. 77–85.

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. MIT Press. Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, USA, Dec.*, 2017, pp. 5099–5108.

[17] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *Proc. 5th International Conference on Learning Representations, Toulon, France, Apr.*, 2017.

[18] Christos Louizos, Max Welling, and Diederik P. Kingma, "Learning sparse neural networks through l_0 regularization," in *Proc. OpenReview International Conference on Learning Representations, Vancouver, Canada, Apr.*, 2018.

[19] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, USA, Jun.*, 2012, pp. 3354–3361.