Transferable Adversarial Attack on 3D Object Tracking in Point Cloud

Xiaoqiong Liu[1(✉)], Yuewei Lin[2], Qing Yang[1], and Heng Fan[1]

[1] Department of Computer Science and Engineering, University of North Texas, Denton, USA
xiaoqiongliu@my.unt.edu, {qing.yang,heng.fan}@unt.edu
[2] Computational Science Initiative, Brookhaven National Laboratory, New York, USA
ywlin@bnl.gov

Abstract. 3D point cloud tracking has recently witnessed considerable progress with deep learning. Such progress, however, mainly focuses on improving tracking accuracy. The risk, especially considering that deep neural network is vulnerable to adversarial perturbations, of a tracker being attacked is often neglected and rarely explored. In order to attract attentions to this potential risk and facilitate the study of robustness in point cloud tracking, we introduce a novel *transferable attack network* (TAN) to deceive 3D point cloud tracking. Specifically, TAN consists of a 3D adversarial generator, which is trained with a carefully designed multi-fold drift (MFD) loss. The MFD loss considers three common grounds, including classification, intermediate feature and angle drifts, across different 3D point cloud tracking frameworks for perturbation generation, leading to high transferability of TAN for attack. In our extensive experiments, we demonstrate the proposed TAN is able to not only drastically degrade the victim 3D point cloud tracker, *i.e.*, P2B [21], but also effectively deceive other *unseen* state-of-the-art approaches such as BAT [33] and M^2Track [34], posing a new threat to 3D point cloud tracking. Code will be available at https://github.com/Xiaoqiong-Liu/TAN.

Keywords: 3D Point Cloud Tracking · Transferable adversarial attack

1 Introduction

3D point cloud tracking, aiming at localizing the target of interest in a sequence of point clouds given its initial state (*e.g.*, a 3D bounding box), is one of most fundamental components of 3D computer vision and has a wide range of crucial applications such as autonomous driving, robotics, and scene understanding. In recent years, rapid progress has been made in 3D point cloud tracking owing to the development of deep learning on point sets (*e.g.* [19,20]), and many excellent deep trackers (*e.g.*, [6,11,21,24,27,33,34]) have been proposed. Despite this, current progress mainly focuses on improving the accuracy of point cloud tracking. The potential risk that, a deep tracker may be attacked by the, even small,

D.-T. Dang-Nguyen et al. (Eds.): MMM 2023, LNCS 13834, pp. 446–458, 2023.
https://doi.org/10.1007/978-3-031-27818-1_37

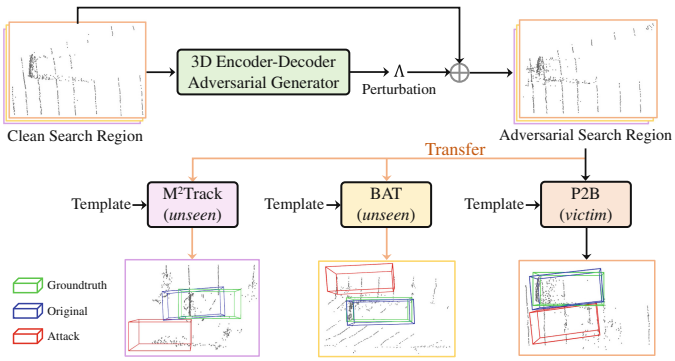

Fig. 1. Illustration of TAN to attack the victim 3D tracker P2B [21] and its transferability to fool other *unseen* approaches including BAT [33] and M^2Track [34]. It is worth noticing that, the two unseen trackers perform better than the victim tracker, while they are deceived by our approach, showing its transferability for attack.

adversarial perturbations, is often ignored and rarely studied, which may cause severe consequences when deploying the tracking system in practice.

The problem of adversarial attack has been firstly explored in [7,25] for 2D images, demonstrating that deep neural network could be deceived via adding small imperceptible perturbations to the original input image. Studying adversary examples to the deep network is beneficial for understanding deep learning and facilitating its robustness [12]. Inspired by this, the research of adversarial attack for the 3D point clouds has been investigated recently. However, current field mainly focuses on attacking point cloud classification task (*e.g.*, [9,10,13,17,18,30,36]), and very little attention is paid to adversarial attack on 3D point cloud tracking. Compared with attack on classification, adversarial attack on point cloud object tracking is different and more challenging. First, instead of handling only one candidate for attack in classification, attack on tracking requires to deal with dense candidates within the research region. Besides, tracking is a temporal video task and all point clouds except for the first one will be attacked, while only one static point cloud is attacked in classification.

Considering the importance of point cloud object tracking and to attract attention to the exploration of adversarial attack on it for facilitating robustness, we introduce a novel framework, dubbed *Transferable Attack Network* (TAN), for attacking 3D object tracking in point cloud. TAN aims at deceiving *not only* the victim tracker *but also* other *unseen* tracking models (see Fig. 1). Specifically, TAN consists of a 3D adversarial generator that is constructed in encoder-decoder style. It takes as input a clean search region point cloud and then outputs the adversarial perturbations to generate a perturbed search region point cloud against the tracker. In order to train TAN, we propose a novel and simple but effective multi-fold drift (MFD) loss, which is the key to achieving high transferability of TAN for attack. Concretely, it considers three different drifts on classification, intermediate feature representation and angle, carefully designed

for 3D tracking. Because these three drifts are common grounds across different point cloud tracking frameworks, our TAN is learned to possess a good transfer capacity from the victim tracker to other unseen models.

In this work, we choose popular state-of-the-art P2B [21] as the victim point cloud tracker, and apply our TAN with proposed MFD loss to generate adversarial perturbation. In extensive experiments on the challenging KITTI [5], we demonstrate that our approach is able to effectively fool P2B [21] by drastically degrading its tracking performance. In addition, we verify the transferability of TAN, learned by attacking P2B and then fixed when transferring, on other recent *unseen* 3D point cloud trackers consisting of BAT [33] and M^2Track [34]. Experimental results show that, even without seeing the architectures of these trackers, the perturbations generated from TAN can successfully fool these models, which poses a new threat to 3D point cloud tracking.

We notice that, there exist attempts to exploring adversarial attack on 2D object tracking in video sequences (*e.g.*, [8,28,31]), but our TAN for adversarial attack on 3D point cloud tracking is considerably different. First, TAN aims to perturb geometric object points, while other 2D attacker works on image pixels. Besides, unlike the 2D case, it is essential to specially consider the sparsity character of point cloud data when designing 3D adversarial attack. Finally, TAN is designated to generate the highly transferable perturbation from the victim tracker to other unseen models, differing from other 2D methods.

To our knowledge, the proposed TAN is the first investigation of transferable adversarial attack on object tracking in 3D point cloud. We hope that it draws researchers' attention on developing more attack methods on point cloud tracking for better understanding this task and the potential risk of its deployment in practical applications. In summary, we make the following contributions:

1) *We introduce a novel Transferable Attack Network (TAN) that effectively fools not only the victim tracker but also other unseen methods. To our knowledge, TAN is the first study of transferable attack on 3D point cloud tracking.*
2) *We propose the Multi-Fold Drift (MFD) loss that specially considers drifts on common grounds across different trackers, consisting of classification, intermediate feature and angle, leading to high transferability of TAN.*
3) *We conduct extensive experiments to verify the effectiveness of TAN in attacking the state-of-the-art P2B and show its high transferability to successfully fool other unseen but stronger trackers.*

2 Related Work

3D Point Cloud Object Tracking. 3D point cloud tracking has been greatly explored in recent years. Inspired by the success in 2D tracking [1,3,14,16], deep Siamese architecture has been exploited in many 3D point cloud trackers. The method of SC3D [6] is the first in applying Siamese network for 3D tracking. Nevertheless, this approach suffers from heavy computation burden because of exhaustive search strategy for candidate generation. To improve SC3D, the work of P2B [21], drawing inspirations from region proposal network (RPN) [22] for 2D Siamese tracking [15], introduces a 3D proposal network to efficiently obtain

candidates for tracking. The approach of BAT [33] proposes a box-aware representation to encode the geometric information of target for point cloud tracking, enhancing the robustness. The work of V2B [11] learns the shape-aware features to deal with the sparsity issue in point clouds for tracking. The work of MLVSNet [27] explores multi-level features for 3D tracking. The work of M^2Track [34] leverages motion information for improving tracking. The algorithm of PTT [24] uses Transformer [26] for augmenting target-aware feature representation for improvements. The work of PTTR [35] explores Transformer to fuse template and search region point clouds for tracking. *Different from* the above approaches that mainly focus on improving 3D tracking accuracy, our method aims at learning adversarial attack against 3D trackers, which facilitates point cloud tracking from another direction.

Adversarial Attack on Point Cloud. Recently, the study of adversarial attack on point cloud has drawn great attention. The work of [30] proposes the first adversarial attack on point cloud classification task. Several attack approaches have been studied and displayed promising attacking performance. The method of [17] extends the adversarial attack for 2D images to point cloud. The work of [9] aims at learning transferable perturbations with auto-encoder reconstruction for 3D point cloud classification. The method of [18] generates adversarial perturbations with joint gradients of original point clouds and outliers. The approach of [13] learns adversarial examples for point cloud classification with small point manipulations. The work of [10] introduces a shape-invariant 3D adversarial attack method for classification. The method of [36] presents a label-guided network to generate adversarial examples for point cloud attack. *Different from* the above methods for point cloud classification, this paper aims at the task of adversarial attack on point cloud object tracking, which is more challenging compared to attacking classification models.

Adversarial Attack on 2D Tracking. The task of attacking 2D object tracking has been largely explored in recent years. The methods in [8,28] propose to generate the adversarial perturbation to attack the tracker via an optimization-based iterative manner. The work of [2] introduces a dual-attention approach to learn the one-shot adversarial attack for tracking. The algorithm of [4] presents an adaptive adversarial attack to fool the aerial object tracker. The work of [31] proposes to deceive a 2D tracker with perturbation from an adversarial generator trained with a cooling-shrinking loss. *Different from* the aforementioned methods for adversarial attack on 2D tracking, our TAN is designated for fooling 3D point cloud tracking with high transferability, which requires significantly different design as discussed before.

3 The Proposed Methodology

In this section, we detail the Transferable Attack Network (referred to as TAN). The overall pipeline for training TAN is shown in Fig. 2. As in Fig. 2, TAN contains a 3D adversarial generator in encoder-decoder architecture. During training, it receives the clean search region point cloud and outputs perturbation used

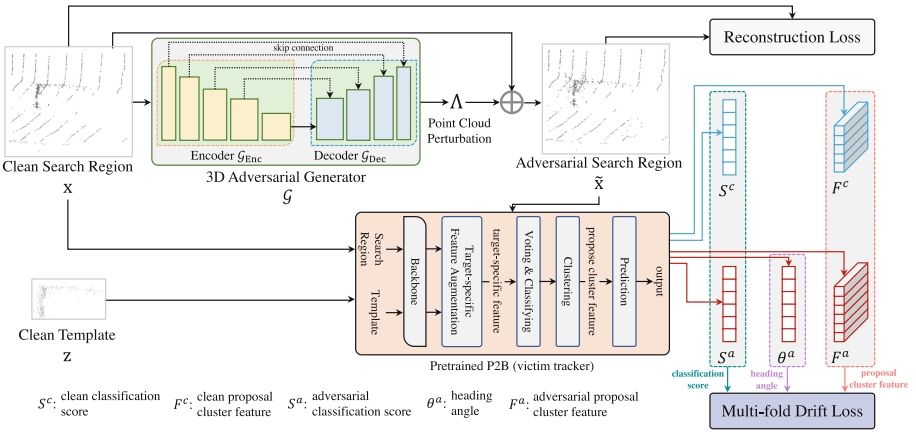

Fig. 2. Illustration of TAN, which consists of a 3D adversarial generator and the victim tracker P2B, and is trained by our multi-fold drift (MFD) loss for high transferability

to generate adversarial search region point cloud. The adversarial search region together with the clean template are fed to the victim tracker for obtaining the adversarial classification scores S^a, heading angles θ^a and intermediate features F^a for proposals. Since the goal of TAN is to drift final tracking result away from the proposal with the highest confidence obtained by the victim tracker, we feed clean search region point cloud to the victim tracker, which, with the clean template, is used to generate clean classification scores S^c and intermediate features F^c for proposals, guiding attack with MFD loss as described later.

In this work, we adopt P2B [21], a recent state-of-the-art 3D tracker, as our victim. But note, other Siamese 3D trackers could also be used in TAN. The brief architecture of P2B is shown in Fig. 2 (the orange block). It receives a pair of template and search region point clouds, and then fuses the template feature into the search region feature to obtain target-specific feature, which goes through voting and classification, and clustering to obtain proposal cluster feature for final prediction. We refer readers to [21] for more details regarding P2B.

3.1 3D Encoder-Decoder Adversarial Generator

The 3D adversarial generator $\mathcal{G}$ is used for generating the perturbation to obtain the adversarial example. As shown in Fig. 2, $\mathcal{G}$ is composed of an encoder $\mathcal{G}_{\text{Enc}}$ and a decoder $\mathcal{G}_{\text{Dec}}$. In this work, we adopt PointNet++ [20] as encoder due to its simplicity and excellent performance in many tasks. It is worth noting, other 3D point cloud backbones (*e.g.*, [32]) could be used as well. For the decoder part, following U-Net architecture [23] for segmentation, we stack a set of interpolating and pointnet [19] layers. To introduce more information from encoder to decoder, we apply skip connections between them.

The generator $\mathcal{G}$ takes the clean search region x as input and outputs the corresponding perturbation Λ as follows,

$$\Lambda = \mathcal{G}_{\text{Dec}}(\mathcal{G}_{\text{Enc}}(\text{x})) \tag{1}$$

The perturbation Λ is the learned point cloud offset that has the same size as of x. Afterwards, it is used to shift the points in x for generating the adversarial search region $\tilde{\mathrm{x}}$ for attack. Mathematically, the adversarial search region $\tilde{\mathrm{x}}$ is obtained as follows,

$$\tilde{\mathrm{x}} = \Lambda + \mathrm{x} \tag{2}$$

3.2 Multi-fold Drift Loss

After obtaining the adversarial search regions $\tilde{\mathrm{x}}$ in Eq. (2), we feed it together with the clean search region x and template z to the pretrained victim 3D tracker P2B (see again Fig. 2) to obtain classification scores, heading angles and features of proposals, as follows,

$$\{S^c, \theta^c, F^c\} = \mathrm{P2B}(\mathrm{z}, \mathrm{x}) \qquad \{S^a, \theta^a, F^a\} = \mathrm{P2B}(\mathrm{z}, \tilde{\mathrm{x}}) \tag{3}$$

where S^c, θ^c (**note**, θ^c is not used) F^c and S^a, θ^a, F^a are classification scores, angles, and cluster features of proposal, respectively. $\mathrm{P2B}(\cdot, \cdot)$ represents inference execution of tracker P2B. With the predictions for x and $\tilde{\mathrm{x}}$, we can compute multi-fold drift (MFD) loss $\mathcal{L}_{\mathrm{MFD}}$, containing a center-aware classification drift loss $\mathcal{L}_{\text{cen-cls}}$, a feature drift loss $\mathcal{L}_{\mathrm{fea}}$ and a heading angle drift loss $\mathcal{L}_{\mathrm{ang}}$, to train TAN, as described in the following.

(1) Center-aware Classification Drift Loss. Since the highest classification score usually directly determines the target position in P2B (and also in other trackers), we introduce a center-aware classification drift loss $\mathcal{L}_{\text{cen-cls}}$ to drive the target on predicted by the clean classification score S^c away on the adversarial classification score S^a. Different than existing 2D attack approaches [4,31] that simply decrease distance between positive and negative classification scores, our center-aware drift loss aims at reducing the classification distance between the region centered at target within a radius r and the outside area beyond a radius γ ($\gamma > r$) on the adversarial classification score, which ensures that the predicted target can be shifted far away. Mathematically, $\mathcal{L}_{\text{cen-cls}}$ is expressed as follows,

$$\mathcal{L}_{\text{cen-cls}} = \max_{p \in R_r^{\text{cen}}} (S^a[p]) - \max_{p \in R_\gamma^{\text{out}}} (S^a[p]) \tag{4}$$

where $(S^a[p])$ is the classification score for proposal cluster p, and R_r^{cen} and R_γ^{out} represent the regions centered at the proposal cluster with highest score with radius r and outside region beyond radius γ based on clean classification results. In particular, they are mathematically expressed as follows,

$$R_r^{\text{cen}} = \{p \mid \|p - p^{\text{cen}}\|_2 \leq r\} \qquad R_\gamma^{\text{out}} = \{p \mid \|p - p^{\text{cen}}\|_2 > \gamma\} \tag{5}$$

where $p^{\text{cen}} = \arg\max_p S^c[p]$ denotes the target proposal cluster with the highest confidence based on S^c.

For point cloud tracking, we observe that, due to the sparsity nature, point cloud in R_γ^{out} may be almost empty. In this case, it is difficult the find adversarial

distractors to fool the tracker. To deal with this, we consider another region R_r^{out} that is closer to R_r^{cen} in the classification drift loss, and modify Eq. (4) as follows,

$$\mathcal{L}_{\text{cen-cls}} = \max_{p \in R_r^{\text{cen}}} (S^a[p]) - (\max_{p \in R_r^{\text{out}}} (S^a[p]) + \max_{p \in R_\gamma^{\text{out}}} (S^a[p]))/2 \tag{6}$$

where $R_r^{\text{out}} = \{p \mid \|p - p^{\text{cen}}\|_2 > r\}$. Note that, the new classification drift loss in Eq. (6) weights more importance on find adversarial distractors in R_γ^{out} because R_r^{out} contains R_γ^{out}, which indicates the preference of adversarial distractors far away from the target center. With our center-aware classification drift loss, we can decrease and meanwhile increase the confidence of target and background regions, causing confusion on classification to fool the tracker.

(2) Feature Drift Loss. Besides attacking directly the tracker on classification, we consider attack on the intermediate features as well, because feature space is a common ground across different backbones of various trackers. In specific, we introduce a feature drift loss $\mathcal{L}_{\text{fea}}$ that drives the adversarial features away from the clean features, resulting in drift on the final classification from another direction. Mathematically, $\mathcal{L}_{\text{fea}}$ is defined as follows,

$$\mathcal{L}_{\text{fea}} = -\|F^c[p^c] - F^a[p^a]\|_2 \tag{7}$$

where p^c and p^a denote the max-score proposals based on clean and adversarial classification results S^c and S^a in R_r^{cen}. $F^c[p^c]$ and $F^c[p^c]$ represent their features, respectively. With feature drift loss, we can push adversarial features away from normal clean features, further enhancing attack and transferability.

(3) Angle Drift Loss. For point cloud tracking, the target is represented with a rotated 3D bounding box. A heading angle in the range $[-\pi, \pi]$ is utilized to indicate the direction of the target. Considering that a perturbed angle can also decrease tracking performance (*e.g.*, success, which is measured by the Intersection over Union), especially in the case of extremely spare point cloud, and bring potential risk for downstream tasks such as motion planning, we introduce an angle loss $\mathcal{L}_{\text{ang}}$ that aims at drifting the heading angle to targeted directions. In specific, we design the $\mathcal{L}_{\text{ang}}$ to enforce the heading angle of the tracking result to drift to the horizontal (*i.e.*, 0 or π). Mathematically, $\mathcal{L}_{\text{ang}}$ is defined as follows,

$$\mathcal{L}_{\text{ang}} = \|h(\theta^a[p^a])\|_2 \tag{8}$$

where p^a is the max-score proposal based on adversarial classification results S^a, and $h(\cdot)$ denotes the piecewise functions as follows,

$$h(x) = \begin{cases} x \cdot \text{sgn}(x) & 0 \le x < \pi/2 \quad \text{or} \quad -\pi/2 \le x < 0 \\ \pi - x \cdot \text{sgn}(x) & -\pi \le x < -\pi/2 \quad \text{or} \quad \pi/2 \le x < \pi \end{cases} \tag{9}$$

where $\text{sgn}(\cdot)$ denotes the sign function.

With the above $\mathcal{L}_{\text{cen-cls}}$, $\mathcal{L}_{\text{fea}}$ and $\mathcal{L}_{\text{ang}}$, $\mathcal{L}_{\text{MFD}}$ is computed as follows,

$$\mathcal{L}_{\text{MFD}} = \lambda_{\text{cen-cls}} \cdot \mathcal{L}_{\text{cen-cls}} + \lambda_{\text{fea}} \cdot \mathcal{L}_{\text{fea}} + \lambda_{\text{ang}} \cdot \mathcal{L}_{\text{ang}} \tag{10}$$

where $\lambda_{\text{cen-cls}}$, λ_{fea} and λ_{ang} represent the weights to balance the loss. Our MFD loss aims to drift the tracking result from multiple common perspectives across different frameworks, allowing effective attack on both victim and unseen models.

3.3 Overall Loss

Overall Loss. In addition to $\mathcal{L}_{\text{MFD}}$ that drifts the target, a distance loss $\mathcal{L}_{\text{dist}}$ is utilized so that the adversarial perturbation is imperceptible to naked eyes. It is defined using L_2 norm as follows,

$$\mathcal{L}_{\text{dist}} = \|\tilde{\mathrm{x}} - \mathrm{x}\|_2 \tag{11}$$

The overall loss $\mathcal{L}_{\text{TAN}}$ to train TAN is then expressed as follows,

$$\mathcal{L}_{\text{TAN}} = \mathcal{L}_{\text{MFD}} + \lambda_{\text{dist}} \cdot \mathcal{L}_{\text{dist}} \tag{12}$$

The training of TAN is performed in an end-to-end manner. Algorithm 1 illustrates its training process. Once the training has been completed, TAN will be directly applied for generating the adversarial search region given the clean search region to fool the victim tracker. Besides, it can be transferred to deceive other unseen approaches as described later.

Algorithm 1. Training Process of TAN

Input: z: clean template; x: clean search region; P2B: pretrained tracker;
1: Initialize 3D generator $\mathcal{G}$, load P2B and freeze its parameters;
2: **repeat**
3: Get clean z and x;
4: Get adversarial perturbation by feeding x to $\mathcal{G}$ (Eq. (1));
5: Get adversarial search region $\tilde{\mathrm{x}}$ (Eq. (2));
6: Get predictions and features by feeding z, x and $\tilde{\mathrm{x}}$ to P2B (Eq. (3));
7: Compute $\mathcal{L}_{\text{MFD}}$ (Eq. (10)), $\mathcal{L}_{\text{dist}}$ (Eq. (11)) and overall loss $\mathcal{L}_{\text{TAN}}$ (Eq. (12));
8: Compute gradient of $\mathcal{L}_{\text{TAN}}$ w.r.t the weights of $\mathcal{G}$ and update these weights;
9: **until** convergence
Output: trained $\mathcal{G}^*$

4 Experiments

Implementation. We implement our TAN by PyTorch on a PC machine with 3 Nvidia RTX A6000 GPUs. The victim tracker is P2B [21] (CVPR 2020), which is applied as it is during training. We train TAN using Adam optimizer. The learning rate is set to 0.0001 with a decay of 0.95. The parameters r and γ used in the center-aware classification drift loss are empirically set to 0.35 and 0.65, respectively. The weights $\lambda_{\text{cen-cls}}$, λ_{ang}, λ_{fea} and λ_{dist} in feature drift and overall losses are set to 2.5, 1, 1 and 5, respectively.

Dataset and Evaluation Metrics. We conduct experiments on the most popular 3D point cloud tracking dataset KITTI [5]. The dataset settings for training and testing follows [21]. For point cloud tracking, one-pass evaluation [29] of Success and Precision is applied to measure performance. The Success is defined using IoU between predicted box and groundtruth box, while Precision is defined as AUC for errors from 0 to 2m. Since we aim to degrade tracking performance, we use drop rates of Success and Precision to measure attack performance.

Table 1. Attack of TAN on P2B [21] using Precision and Success drop rates.

		Precision			Success		
		Ori. (%)	Att. (%)	Drop Rate	Ori. (%)	Att. (%)	Drop Rate
P2B (*victim*)	Car	74.4	41.4	44.4%	59.9	33.6	43.9%
	Pedestrian	51.7	21.6	58.2%	31.2	11.6	62.8%
	Van	47.4	27.9	41.1%	40.7	24.1	40.8%
	Cyclist	41.1	19.6	52.3%	28.7	14.4	49.8%
	Average	53.7	27.6	48.5%	40.1	20.9	47.9%

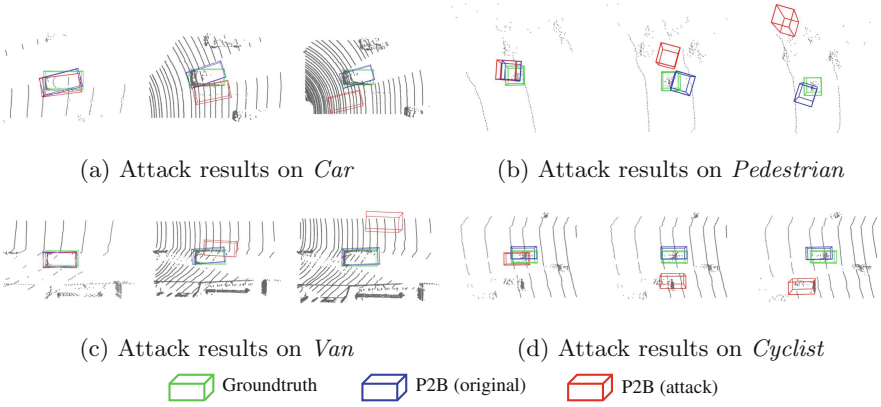

Fig. 3. Qualitative attack effects of our method on P2B in terms of different categories.

4.1 Overall Attack Performance

Table 1 reports the overall attack performance of TAN on P2B[1]. P2B is one of the recent state-of-the-art 3D trackers with excellent result. Despite this, our TAN is able to significantly degrade its performance. Specifically, our TAN decreases the Success scores of P2B on *Car*, *Pedestrian*, *Van* and *Cyclist* from 59.9%, 31.2%, 40.7% and 28.7% to 33.6%, 11.6%, 24.1% and 14.4%, respectively, with drop rates of 43.9%, 62.8%, 40.8% and 49.8%. On average, we degrade the Success score from 40.1% to 20.9% with an enor-

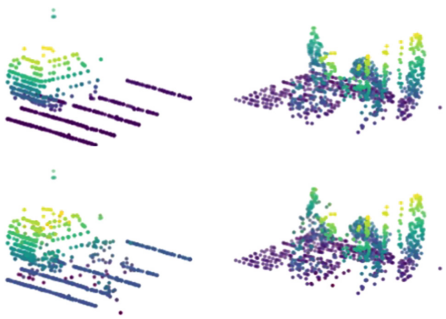

Fig. 4. Visualization of clean search regions (top) and corresponding adversarial search regions (bottom).

[1] In this work, we choose to train P2B on different categories ourselves, because many models are not provided. The training settings are the same as in official P2B. Due to difference in PyTorch version and machine, the results may be slightly different.

Table 2. Ablation study on MFD loss using average Precision and Success scores.

$\mathcal{L}_{\text{cen-cls}}$	$\mathcal{L}_{\text{fea}}$	$\mathcal{L}_{\text{ang}}$	Average precision			Average success		
			Ori. (%)	Att. (%)	Drop Rate	Ori. (%)	Att. (%)	Drop rate
✓			53.7	32.5	39.5%	40.1	24.2	39.7%
✓	✓		53.7	28.4	47.1%	40.1	23.1	42.4%
✓	✓	✓	53.7	27.6	48.5%	40.1	20.9	47.9%

Table 3. Transfer of TAN to other unseen trackers BAT [33] and M^2Track [34].

		Precision			Success		
		Ori. (%)	Att. (%)	Drop Rate	Ori. (%)	Att. (%)	Drop Rate
BAT (*unseen*)	Car	74.3	47.8	35.7%	62.0	36.0	41.9%
	Pedestrian	71.7	32.3	55.0%	43.5	25.6	41.1%
	Van	62.3	24.9	60.0%	51.8	21.9	57.7%
	Cyclist	47.1	37.1	21.2%	29.5	19.2	34.9%
	Average	63.9	35.5	44.4%	46.7	25.7	45.0%
M^2Track (*unseen*)	Car	81.1	35.8	55.9%	67.2	34.4	48.8%
	Pedestrian	89.5	48.7	45.6%	60.8	35.6	41.4%
	Van	65.8	45.7	30.5%	52.7	36.8	30.2%
	Cyclist	93.4	76.9	17.7%	71.5	46.7	34.7%
	Average	82.5	51.8	37.2%	63.1	38.4	39.1%

mous drop rate of 47.9%, which shows the effectiveness of our approach in deceiving the 3D point cloud tracker. In addition to the quantitative analysis, we qualitatively demonstrate the attack effect. As shown in Fig. 3, we can observe that the proposed TAN can effectively drift the tracker. Moreover, we show the clean and learned adversarial search region points in Fig. 4, from which we can see they are visually similar and the perturbations are imperceptible.

4.2 Ablation Study

This section studies the impact of different drift losses in MFD loss. Table 2 shows the results. From Table 2, we can see that, when using drift loss $\mathcal{L}_{\text{cen-cls}}$, the drop rates for Precision and Success are 39.5% and 39.7%, respectively. When incorporating feature drift loss $\mathcal{L}_{\text{fea}}$ in MFD, we improve the drop rates to 47.1% and 42.4% with 7.6% and 2.7% gains. Together with the angle drift loss $\mathcal{L}_{\text{ang}}$, we obtain the best drop rates 48.5% and 49.7%. These experiments show that each drift loss in MFD is beneficial for improving drop rates for better attack.

4.3 Transfer to Unseen Trackers

TAN aims at high transferability to unseen trackers. To verify this, we conduct experiments by applying TAN learned with P2B to unseen trackers BAT [33] and

M^2Track [34]. BAT introduces a box-aware representation to enhance the target features for improvement. M^2Track explores motion cues to boost performance. Note, both BAT and M^2Track perform better than the victim model P2B.

Since not all models on KITTI are provided, we train our own models for BAT and M^2Track and use them for experiments. Table 3 shows the attack performance by transferring TAN to other trackers. From Table 3, we can observe that, by applying TAN to BAT and M^2Track, the average Success scores are decreased from 46.7%/63.1% to 23.3%/38.4%, achieving drop rates of 45.0% and 39.1%, respectively. Likewise, we can see that, the average Precision scores are significantly reduced from 63.9%/82.5% to 35.5% and 51.5% with drop rates of 44.4% and 37.2%, which shows the high transferability of TAN to other unseen trackers and poses a new threat to 3D point cloud tracking.

5 Conclusion

In this paper, we propose a Transferable Attack Network (TAN) against 3D point cloud trackers. In specific, we design a 3D adversarial generator in TAN and train it using a novel multi-fold loss that considers classification, intermediate feature and angle for perturbation generation. Because the multi-fold loss is carefully designed for general purpose, it enables high transfer of TAN to other trackers. In our experiments, we show that TAN is able to successfully attack a recent state-of-the-art 3D tracker P2B. Besides, we validate its transferability to other unseen models by degrading their performance. Our results show the vulnerability in recent 3D trackers, which prompts us to design robust methods for safety-critical applications. In addition, the study of properties of adversarial examples, has the potential to be leveraged to enhance the robustness.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: ECCVW (2016)
2. Chen, X., et al.: One-shot adversarial attacks on visual tracking with dual attention. In: CVPR (2020)
3. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: CVPR (2019)
4. Fu, C., Li, S., Yuan, X., Ye, J., Cao, Z., Ding, F.: AD2attack: adaptive adversarial attack on real-time UAV tracking. In: ICRA (2022)
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The Kitti vision benchmark suite. In: CVPR (2012)
6. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3d Siamese tracking. In: CVPR (2019)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
8. Guo, Q., et al.: SPARK: spatial-aware online incremental attack against visual tracking. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 202–219. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_13

9. Hamdi, A., Rojas, S., Thabet, A., Ghanem, B.: AdvPC: transferable adversarial perturbations on 3D point clouds. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12357, pp. 241–257. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58610-2_15
10. Huang, Q., Dong, X., Chen, D., Zhou, H., Zhang, W., Yu, N.: Shape-invariant 3d adversarial point clouds. In: CVPR (2022)
11. Hui, L., Wang, L., Cheng, M., Xie, J., Yang, J.: 3d Siamese voxel-to-BEV tracker for sparse point clouds. In: NeurIPS (2021)
12. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: NIPS (2019)
13. Kim, J., Hua, B.S., Nguyen, T., Yeung, S.K.: Minimal adversarial examples for deep learning on 3d point clouds. In: ICCV (2021)
14. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: SiamRPN++: evolution of Siamese visual tracking with very deep networks. In: CVPR (2019)
15. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with Siamese region proposal network. In: CVPR (2018)
16. Lin, L., Fan, H., Zhang, Z., Xu, Y., Ling, H.: SwinTrack: a simple and strong baseline for transformer tracking. In: NeurIPS (2022)
17. Liu, D., Yu, R., Su, H.: Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In: ICIP (2019)
18. Ma, C., Meng, W., Wu, B., Xu, S., Zhang, X.: Efficient joint gradient based attack against SOR defense for 3d point cloud classification. In: ACM MM (2020)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
20. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: NIPS (2017)
21. Qi, H., Feng, C., Cao, Z., Zhao, F., Xiao, Y.: P2B: point-to-box network for 3d object tracking in point clouds. In: CVPR (2020)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
23. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
24. Shan, J., Zhou, S., Fang, Z., Cui, Y.: PTT: point-track-transformer module for 3d single object tracking in point clouds. In: IROS (2021)
25. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv:1312.6199 (2013)
26. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
27. Wang, Z., Xie, Q., Lai, Y.K., Wu, J., Long, K., Wang, J.: MLVSNet: multi-level voting Siamese network for 3d visual tracking. In: ICCV (2021)
28. Wiyatno, R.R., Xu, A.: Physical adversarial textures that fool visual object tracking. In: ICCV (2019)
29. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: CVPR (2013)
30. Xiang, C., Qi, C.R., Li, B.: Generating 3d adversarial point clouds. In: CVPR (2019)
31. Yan, B., Wang, D., Lu, H., Yang, X.: Cooling-shrinking attack: blinding the tracker with imperceptible noises. In: CVPR (2020)
32. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)

33. Zheng, C., et al.: Box-aware feature enhancement for single object tracking on point clouds. In: ICCV (2021)
34. Zheng, C., et al.: Beyond 3d Siamese tracking: a motion-centric paradigm for 3d single object tracking in point clouds. In: CVPR (2022)
35. Zhou, C., et al.: PTTR: relational 3d point cloud object tracking with transformer. In: CVPR (2022)
36. Zhou, H., et al.: LG-GAN: label guided adversarial network for flexible targeted attack of point cloud based deep networks. In: CVPR (2020)