This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

# Explaining 3D Point Cloud Semantic Segmentation Models Through Adversarial Attacks

Jorge Francisco Ciprián-Sánchez[1], Josafat-Mattias Burmeister[2], Rico Richter[2], Jürgen Döllner[1]
[1]University of Potsdam, Hasso Plattner Institute, Germany
[2]University of Potsdam, Germany
{jorge.cipriansanchez, juergen.doellner}@hpi.uni-potsdam.de,
{burmeister, rico.richter.1}@uni-potsdam.de

## Abstract

*Although deep learning methods for the semantic segmentation of 3D point clouds have been extensively researched, there is limited research on explainable artificial intelligence methods for this domain. Therefore, we extend the Critical Traversal Attack (CTA) approach previously proposed for 3D point cloud classification models to the semantic segmentation task. This approach aims to alter the predictions of a model by perturbing salient input points, which can provide counterfactual explanations. To transfer CTA to semantic segmentation, we reformulate its loss function and success criterion so that the prediction probability of all points belonging to a target class is decreased. Through this method, our study investigates the following aspects of 3D point cloud segmentation models: 1) Sensitivity to perturbations in different input channels. Our results show that the studied architectures are more vulnerable to perturbations in color than spatial features, with color changes for very few points already leading to successful attacks. We further study the per-color-channel sensitivity, finding that models tend to be more sensitive to perturbations in the green channel and more robust to perturbations in the red channel. 2) Degree of perturbation needed to alter the semantic segmentation results for a target class. We find that the degree of perturbation needed for spatial features is considerably large for most architectures, whereas a much smaller degree of perturbation is needed for the color features. 3) Transferability of adversarial attacks across deep learning architectures. Our experiments indicate that, in general, adversarial attacks display mid-to-low transferability for both spatial and color features.*

## 1. Introduction

In recent years, deep learning models for the semantic segmentation of 3D point clouds have been extensively researched [22, 23, 32–36]. However, there is limited research on explainable artificial intelligence (XAI) methods in this domain [13, 27] and, thus, such models remain largely as black boxes, with little information on key characteristics such as sensitivity to different input features and perturbations, both of which can be indicators of the reliability of such models. One way to analyze these characteristics is through adversarial attacks, which attempt to alter the predictions of a model by small perturbations of the model inputs. If successful, the generated adversarial inputs can act as counterfactual examples that are more easily understood by humans [30]. Although there is an increasing number of works that study adversarial attacks for 3D point cloud classification models [11, 12, 17–19, 37, 40], research on the corresponding 3D point cloud segmentation domain is a nascent field, with initial works proposing point-shifting approaches [38] and methods that physically modify the scene from which the 3D point cloud is sampled [42]. Furthermore, to the best of our knowledge, only one work leverages adversarial attacks to produce counterfactuals to investigate the vulnerability of 3D point cloud classification models against input shifts [30], while existing works on 3D point cloud semantic segmentation focus exclusively on performance degradation and object hiding. The present work aims to contribute to this gap in the literature through the following contributions:

1. We extend the CTA approach of Tan and Kotthaus [30] to the domain of semantic segmentation by redefining its optimization goal and success criterion. We define the optimization goal so that the prediction probability of a target class is reduced for all points that have been assigned to that class by the model. We consider an attack successful if the intersection-over-union (IoU) between the points labeled as the target class for the original and the adversarial input is reduced below a certain threshold. In addition, we incorporate the ability to perturbe not only point coordinates, but also surface features such

as color. We refer to this extension as CTA-seg for the remainder of this paper.

2. We use CTA-seg to investigate three key aspects of 3D point cloud segmentation models:
   - **Sensitivity to perturbations of different input features.** Our results show that the architectures studied are more vulnerable to perturbations in color than in spatial features. We find greater robustness towards changes in the red channel, and more sensitivity to changes on the green channel. Linking our analysis with saliency maps, we find that the red channel seems to provide less information for the model predictions. **Degree of perturbation needed to change the semantic segmentation results of a target class.** We analyze the the number of perturbed points and the imperceptibility of the perturbations needed for successful adversarial attacks. We find that small color changes for even a few points can lead to successful attacks, suggesting that the models give more weight to color features in the prediction.
   - **Transferability of the adversarial attacks between different deep learning architectures.** Our results point towards mid-to-low transferability across deep learning architectures for spatial and color features.

## 2. Related Work

Existing adversarial attacks on 3D point clouds can be divided into two categories [18, 30]: (1) distributional attacks, which are based on the independent perturbation of individual points and can lead to scattered, possibly outlier-like perturbations, and (2) shape-based attacks, which attempt to bundle the perturbations so that the perturbed 3D point clouds still represent coherent point clusters. In the present work, we focus on adversarial attacks that shift point features, as point-wise perturbations "enable more intrinsic explorations of the model such as stabilities and decision boundaries" [30]. Following this categorization, our approach can be classified as a distributional attack.

### 2.1. Adversarial Attacks on 3D Point Cloud Classifiers

**Distributional Approaches.** Liu et al. [17] transfer the fast gradient sign method and the iterative fast gradient method to point cloud classifiers and extend them by an $l_2$-norm constraint. Xiang et al. [37] attack point cloud classifiers through gradient-based point shifting following the C&W framework [6], exploring both the shifting of existing points and the addition and shifting of newly added points. Hamdi et al. [8] propose an extension of this approach, AdvPC, in which the adversarial examples are optimized to still fool the target model after reconstruction by an autoencoder. Kim et al. [12] develop an $l_0$-optimization-based method to generate adversarial examples with a minimal number of perturbed points and close to the original input. Furthermore, the authors explore the concept of *critical points*, that is, points that characterize the shape of a 3D point cloud and are thus relevant for object classification. Following the concept of critical points, Sun et al. [28] use a gradient-based approach to identify these points, selecting those with the highest saliency scores for the prediction of the target class. They then propose score and contractive loss functions to address the trade-off between the attack goal and the imperceptibility constraint. Tan and Kotthaus [30] propose two optimization-based methods: One Point Attack (OPA) and CTA, incorporating Integrated Gradients (IG) to identify critical points. The authors employ these approaches to study the sensitivity to critical point perturbations of 3D point cloud classification models. In this work, we transfer this approach into the semantic segmentation domain to explore key aspects of such models.

**Shape-based Approaches.** Liu et al. [18] compare distributional attacks with shape-based attacks that modify the underlying surface of a point cloud. They propose the resampling of perturbed examples, the addition of stick-shaped point clusters, and the creation of sinks in the point cloud surface as shape-based attacks. Lou et al. [19] propose HiT-ADV to attack 3D point cloud classifiers through human-imperceptible shape deformations. For this purpose, they identify 3D point cloud regions with high saliency and high surface complexity and smoothly deform them using Gaussian kernel functions. Zheng et al. [40] use the Shapley value to assign an importance score to each region in a 3D point cloud and propose an optimization-based gradient attack that only shifts points in salient point cloud regions.

### 2.2. Adversarial Attacks on 3D Point Cloud Segmentation Models

Research on adversarial attacks for 3D point cloud segmentation models is a nascent field, with the initial works of Zhu et al. [42] and Xu et al. [38] addressing this field. Zhu et al. [42] develop a framework in which objects are placed in the physical space to perform obstacle-hiding and road surface-changing attacks on LiDAR semantic segmentation models used for autonomous driving. Xu et al. [38] propose an optimization-based attack method for performance degradation and object-hiding tasks, presenting norm-unbounded and norm-bounded variations. The authors use the proposed attacks to evaluate the robustness of 3D point cloud segmentation models. Their results show that all examined architectures (PointNet++ [22], RandLA-Net [9], and ResGCN-28 [15]) are vulnerable to adversarial attacks, in particular to the norm-unbounded variation applied to color features.

Although our work is similar to that of Xu et al. [38], it differs in the following points:

1. Since we aim to use CTA-seg as an XAI technique, our

focus lies not on performance degradation of the attack targets w.r.t. the ground truth. Instead, we focus on generating adversarial examples that can successfully alter the *predicted semantic labels*. In this way, the generated adversarial examples can better reflect the model behavior and serve as counterfactual explanations.

2. Xu et al. [38] rely on gradients for critical point selection, which are vulnerable to saturation [1, 30] and can negatively affect the critical point selection. Following Tan and Kotthaus [30], we use IG [29] to obtain our critical point set. In [38], critical point selection is performed only for attacks on spatial features. In this work, we study the use of critical points for attacks on color features as well. Furthermore, we analyze the per-channel sensitivity for the color features.
3. We study the *degree of perturbation* needed to produce a successful adversarial example, i.e., how similar the generated adversarial examples are to the original inputs.
4. A more comprehensive transferability analysis between three widely used architectures is performed. We use PointNet++ [22], KP-FCNN [32] and the state-of-the-art Point Transformer V3 (PTv3) [36], as they represent multi-layer perceptron (MLP)-, convolution-, and transformer-based approaches, respectively.

# 3. Methods

## 3.1. CTA-seg Algorithm

Building on the work by Tan and Kotthaus [30], we define the CTA-seg objective as follows. Let $P \in \mathbb{R}^{N \times D}$ be an input 3D point cloud where $N$ represents the number of points and $D$ the number of per-point features, $f : P \rightarrow Y \in \mathbb{R}^{N \times C}$ the output probabilities of a 3D point cloud semantic segmentation model where $C$ is the number of semantic classes, and $M(a, b) \in \mathbb{R}^{N \times D}$ a perturbation matrix between instances $a$ and $b$. CTA-seg seeks to generate an adversarial example $P' \in \mathbb{R}^{N \times D}$ that solves the following adversarial task:

$$\arg \min_{M(P,P')} (|\{m \in M(P,P') | m \neq 0\}| + \|M(P,P')\|) \quad \text{s.t.} \quad IoU(P_t, P'_t) < \theta, \tag{1}$$

where $P_t$ represents the points assigned to the target class $t$ for the input 3D point cloud $P$ and $P'_t$ the points assigned to $t$ for the adversarial example $P'$ by the semantic segmentation model, and $\theta$ is a threshold on the IoU score between $P'_t$ and $P_t$ below which the attack is considered successful. To address this task, we define the loss function of CTA-seg as follows:

$$L = \alpha \cdot \log(f_t(P')) + \beta \cdot \sum_{i=1}^{T_D} D_i(P, P'), \tag{2}$$

where $\alpha$ represents the optimization rate, $f_t(P')$ the model output probabilities for the target class, $\beta$ a distance-penalizing weight, $T_D$ the total number of distance metrics to be used according to the target features (in our case, spatial distance $D_S$ and color distance $D_C$, see Sec. 3.4). Accordingly, $\sum_{i=1}^{T_D} D_i(P, P')$ is the sum of distance metrics between the original input $P$ and the adversarial example $P'$ that quantify the imperceptibility of the perturbations of the target features.

Similar to [30], CTA-seg attempts to perturb the minimum possible number of points of each input to achieve the adversarial task formulated in Eq. (1). It uses a saliency technique to identify a set of critical points (i.e., points that are identified as having a high contribution towards the model prediction, see Sec. 3.2). From these critical points, our algorithm starts with a subsample of $n_p = 1$ of points to be perturbed. The corresponding target features of the selected points are shifted using a gradient-descent approach with a fixed number of steps to minimize the loss defined in Eq. (2). If not successful (local failure), it increases $n_p$ by $k$ points and begins the local optimization loop once again, until either the attack is successful or a global failure condition is reached (see Sec. 3.3).

For the semantic segmentation task, there are two constraints on the loss computation and optimization steps:

1. For the loss calculation, we are only interested in the prediction probabilities for the target class for the points that the model originally assigned to the target class.
2. For the optimization step, we are only interested in the gradients w.r.t. the critical points and from those, only on the ones of the target feature(s) that will be perturbed (e.g., XYZ coordinates or RGB values).

Sec. 7 in the supplementary material elaborates on the implementation details regarding these constraints and shows the detailed pseudocode for CTA-seg.

## 3.2. Selection of Critical Points

In line with [30], we use saliency maps generated by IG [29] for the selection of critical points (i.e., the points to be shifted). For the semantic segmentation task, we use the IG implementation by Schwegler et al. [27] to obtain a saliency map for each CTA-seg input for a given target class. Given that IG computes an attribution value for each per-point feature, we only consider the attribution values corresponding to the target features. Finally, we define the final critical point set as the set of points with attribution values $> 0$.

## 3.3. Stopping Conditions

Local and global stopping criteria are defined to stop the optimization process when certain conditions are met. In our work, local stopping refers to the exit of the optimization loop for a given number of points $n_p$, whereas global stopping refers to the exit of the attack process, whereupon the

attack is considered unsuccessful. An attack is only considered successful when the condition stated in Eq. (1) is fulfilled; i.e., when the IoU of the prediction on the adversarial example decreases below a given threshold for the target class.

For the local stopping conditions, we define a fixed number of maximum local iterations. Furthermore, we track the mean prediction probability for all target points every 25 local iterations - if the mean prediction probability has increased, we consider the local step unsuccessful and continue for the next $n_p + k$ points to be perturbed.

To avoid unfeasible execution times, it is necessary to define a maximum number of global iterations. In contrast to the classification case, each CTA-seg input can contain a variable number of points and points belonging to the target class. To account for this, we estimate the maximum number of global iterations on a per-case basis as follows:

$$maxGlobalIter = \min(|\{p \in P | \text{IG}_t(f(p)) > 0\}|, \\ 3 \cdot |P_t|), \tag{3}$$

that is, the minimum between the number of points for which IG returns an attribution value greater than zero for target class $t$ and three times the number of points of the target class present in the CTA-seg input, as predicted by the model. The latter value is chosen based on the observation that IG often assigns positive attribution values to more points than those of the target class - thus, we posit that points in the surrounding scene may provide relevant context for the segmentation of the target class.

### 3.4. Distance Metrics

To include a regularization in the loss computation that optimizes for the imperceptibility of the perturbations between $P$ and $P'$, it is necessary to define distance metrics for the input features. In this work, we use the bi-directional Hausdorff distance as the spatial distance $D_S$ element in the loss function. Previous works have introduced the use of the bi-directional Hausdorff distance as an imperceptibility metric [12, 30, 37, 40, 41], as it can reduce the generation of outliers [40]. It is defined as follows:

$$D_H(P, P') = \max \left( \max_{p_i \in P} \left( \min_{p'_j \in P'} \|p_i - p'_j\|_2 \right), \\ \max_{p'_j \in P'} \left( \min_{p_i \in P} \|p'_j - p_i\|_2 \right) \right). \tag{4}$$

CTA-seg is capable of shifting not only point coordinates, but also additional input features. For the present work, we consider RGB values, if present, as additional per-point features. For the color distance $D_C$ computation, we select the well-established CIE76 $\Delta E^*_{ab}$ formulation [25], as, for our purpose, it strikes an acceptable balance between perceptual uniformity and simplicity of implementation. The CIELAB color space represents color using three terms: $L^*$ for perceptual lightness and $a^*$ and $b^*$ for the four unique colors of human vision: red, green, blue, and yellow [3]. The CIE76 $\Delta E^*_{ab}$ distance metric is defined as follows:

$$\Delta E^*_{ab} = \sqrt{(L^*_{p_i} - L^*_{p'_i})^2 + (a^*_{p_i} - a^*_{p'_i})^2 + (b^*_{p_i} - b^*_{p'_i})^2}, \tag{5}$$

where, for the present work, $p_i \in P$ and $p'_i \in P'$. We then define $D_C$ metric as follows:

$$D_C = \frac{1}{N} \sum_{i=1}^{N} \Delta E^*_{ab}(p_i, p'_i), \tag{6}$$

where $N$ is the number of points in the CTA-seg input.

## 4. Experiments

### 4.1. Dataset

For our experiments, we use the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) dataset [4]. It consists of 3D point clouds of several buildings, covering an area of over $6000\,\text{m}^2$ and approximately $215\,\text{M}$ points [4]. Each point is given a ground truth class label for the segmentation task, covering three broader categories (*Structural*, *Movable*, and generic *Clutter*) with eleven semantic classes in total. The 3D point clouds of the dataset are grouped into six areas. We use areas 1, 2, 4, and 5 for training, area 6 for validation, and area 3 for testing. All of our evaluations are conducted on the test set.

### 4.2. Model Inputs

3D point clouds used in real-world semantic segmentation tasks can span large areas [4, 5, 26] and typically consist of up to billions of points [7]. To input such large-scale 3D point clouds to deep learning models, data processing pipelines typically rely on dividing them into smaller subsets (local neighborhoods) [16], and reducing the density of the 3D point clouds [10], or a combination of both [24, 31]. We define an input for the semantic segmentation models and, by extension, to CTA-seg, as a local neighborhood drawn from such a dataset, downsampled to a fixed maximum number of points. This allows CTA-seg to be integrated into any 3D point cloud semantic segmentation pipeline. For our experiments, we sample all input neighborhoods from the large-scale 3D point clouds of the S3DIS dataset. We set the maximum neighborhood size $s = 2^{14}$ points, a value set due to memory constraints. Neighborhoods containing larger amounts of points are downsampled through random sampling until the desired number of

points is reached. All input neighborhood coordinates are then normalized to a unit sphere, and all RGB values are normalized to the $[0, 1]$ range.

### 4.3. Experimental Setup

We evaluate three aspects of 3D point cloud segmentation models: (1) their sensitivity to perturbations of spatial and color features, (2) the degree of perturbation that needs to be introduced for an attack to be successful, and (3) the transferability of the attacks between architectures. We conduct all experiments on three widely used deep learning architectures: PointNet++ [22], KP-FCNN [32] and PTv3 [36], as they represent MLP-, convolution-, and transformer-based approaches, respectively.

We focus on one class from the *Structural* and one from the *Movable* categories, choosing the ones for which the PTv3 and KP-FCNN models achieve the highest IoU scores, namely *Structural_Door* and *Movable_Chair*. We exclude the *Floor*, *Ceiling* and *Wall* classes from this study since they represent flat surfaces with little color variations.

In line with [30], we take 25 neighborhoods that contain these classes as attack targets. We select neighborhoods where more than 2.5% of the points are classified as the target class, a value found empirically at which recognizable parts of the target classes are present. The learning rate, $\alpha$ and $\beta$ values, and the maximum number of local optimization attempts are taken from [30] (1.0, $1 \times 10^{-6}$, $1 \times 10^{-6}$ and 2500, respectively) and we set the IoU success threshold to 0.5. We find that increasing the amount of perturbed points $n_p$ by one at each local iteration results in very high execution times, as often a larger number of points must be perturbed to affect the IoU score. Thus, we set the increase $k$ of perturbed points to 50, a value found empirically.

All experiments were run on a single NVIDIA DGX A100 GPU with 80 GB of RAM using Python 3.9, PyTorch 2.0.1, and CUDA 11.7. To make our experiments as reproducible as possible, we seed the data sampling processes, the model's initialization, and the selection of CUDA algorithm implementations by the PyTorch framework with a random seed of 0. Nevertheless, the selected CUDA algorithms themselves are implemented in a non-deterministic manner in some cases.

### 4.4. Results

Tab. 1 shows the success rates for all architectures and target features. We define the success rate as follows:

$$\text{Success rate} = \frac{\text{Number of successful attacks}}{\text{Total number of attacks}}. \quad (7)$$

All studied architectures display a high vulnerability to attacks on the color features, while PTv3 and KP-FCNN show more robustness to perturbations on spatial features.

| Class | Architecture | Success Rate | |
|---|---|---|---|
| | | XYZ | RGB |
| Chair | PTv3 | 0.52 | 0.96 |
| | KP-FCNN | 0.64 | 0.92 |
| | PointNet++ | 1.0 | 1.0 |
| Door | PTv3 | 0.64 | 1.0 |
| | KP-FCNN | 0.68 | 1.0 |
| | PointNet++ | 1.0 | 1.0 |

Table 1. Success rates across architectures and target features.

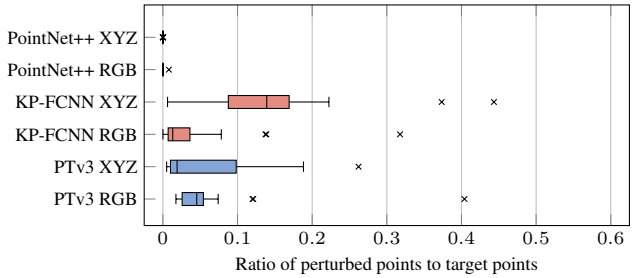



Figure 1. Ratio of perturbed points to target points per successful attack for the *Chair* class.

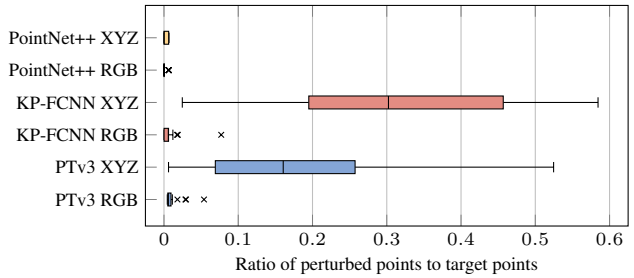



Figure 2. Ratio of perturbed points to target points per successful attack for the *Door* class.

Sec. 8 in the supplementary material shows examples of successful adversarial attacks for spatial and color features.

#### 4.4.1. Perturbation Sensitivity

To analyze the perturbation sensitivity of the models, we first analyze the ratio of perturbed to target points needed to produce a successful attack (Fig. 1 and Fig. 2). When attacking the spatial features, a marked difference in robustness between the classes can be observed, with models showing a more robust behavior for the *Door* class, needing to perturb up to close to 60% of the target points. In contrast, for the *Chair* class, less than 25% of points need to be perturbed to obtain a successful attack at most (excluding outliers). Regarding the color features, all architectures exhibit a high sensitivity, needing to alter less than 10% of the target points to produce a successful attack across both classes. However, PTv3 and KP-FCNN show a higher robustness for the *Chair* class.

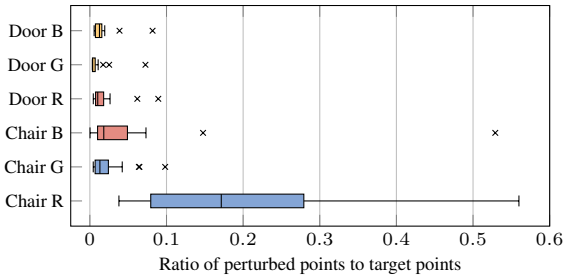



Figure 3. Ratio of perturbed points to target points per successful attack for the red, green, and blue channels for the PTv3 architecture and both target classes.

Given the high sensitivity to perturbations of the color features displayed in these results, we further study the per-color-channel sensitivity of the PTv3 architecture. As shown in Fig. 3, for the *Chair* class, there is a marked difference between the percentage of points that need to be perturbed on the red, green, and blue channels. To achieve a successful attack on the red channel, at most approximately 55% of points need to be shifted. This is a strong contrast with the green and blue channels, where no more than 10% of points need to be shifted at most. We can also notice a similar but less pronounced trend for the *Door* class, where for all channels, less than 5% of points need to be shifted.

#### 4.4.2. Perturbation Degree

To study the degree of perturbation present in the adversarial examples, we first investigate how the number of successful attacks changes when we increase the ratio of perturbed to target points (Fig. 4 and Fig. 5). For both classes, we observe a slower increase in the number of successful neighborhoods w.r.t. the percentage of perturbed points when attacking the spatial features, a behavior more pronounced for the *Door* class. In contrast, when attacking color features, the majority of attacks are successful when perturbing less than 15% of the target points. The *Door* class shows a stronger vulnerability, as all color-based attacks are successful at 10% of perturbed points, regardless of the victim architecture.

Second, we measure the pairwise distance between the original and adversarial 3D point clouds. We use the bi-directional Hausdorff distance to measure the degree of spatial perturbations, whereas for the color features we use the CIE76 $\Delta E^*_{ab}$ distance. To have a criterion by which to determine if a distance is large or not, we establish the following references:

1. Spatial distance: Since all XYZ coordinates are normalized to a unit sphere, we consider the bi-directional Hausdorff distance between a point cloud of size $N = 2^{14}$ where all points are at (0, 0, 0) and one where all points are distributed uniformly on the surface of the unit sphere. The mean distance for this setup is of 1.0.

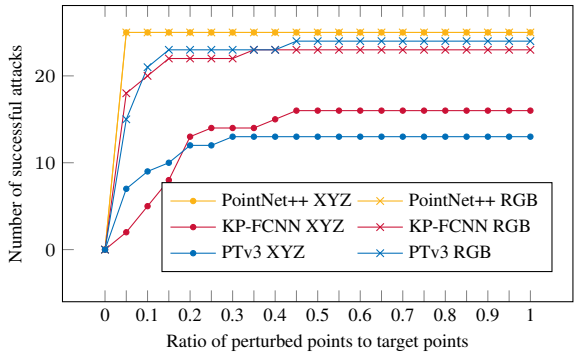



Figure 4. Ratio of perturbed points to target points vs. number of successful attacks for the *Chair* class.

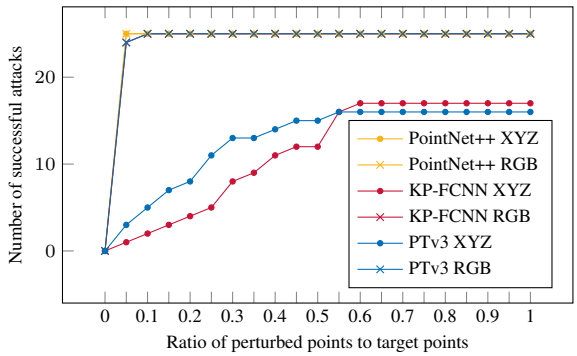



Figure 5. Ratio of perturbed points to target points vs. number of successful attacks for the *Door* class.

2. Color distance: Since the evaluation of color distance is related to the human perception of color, we use the concept of just noticeable difference (JND) [2]. For the CIELAB color space, a CIE76 $\Delta E^*_{ab}$ of approximately 2.9 corresponds to the JND [20].

Tab. 2 shows the mean spatial and color distances for all experiment configurations. Regarding the adversarial attacks on spatial features, we can observe that the *Door* class requires a larger change in the coordinate values to produce successful attacks. PTv3 and KP-FCNN require perturbations well over the established baseline of 1.0 - meaning that the affected coordinates are placed far outside the unit sphere in many cases. In particular, PTv3 shows the strongest robustness against attacks to spatial features, requiring changes two orders of magnitude higher. In contrast, attacks on PointNet++ require comparatively small perturbations close to the established baseline.

Attacks on the color features show a considerably low color difference compared to the JND criterion, as all experiments show distances well below the 2.9 threshold. For the *Chair* class, the KP-FCNN architecture shows the most robustness to attacks, requiring larger color perturbations.

| Class | Archit. | XYZ Features Hausdorff | Std. | RGB Features CIE76 $\Delta E^*_{ab}$ | Std. |
|---|---|---|---|---|---|
| Chair | PTv3 | 58.9886 | 100.1644 | 0.0765 | 0.0952 |
| | KP-FCNN | 13.4335 | 4.6251 | 0.6013 | 0.9664 |
| | PointNet++ | 0.3655 | 1.2443 | 0.0004 | 0.0012 |
| Door | PTv3 | 286.6936 | 152.6965 | 0.1475 | 0.1532 |
| | KP-FCNN | 15.6496 | 3.2408 | 0.1187 | 0.2336 |
| | PointNet++ | 2.8815 | 3.4049 | 0.0091 | 0.0146 |

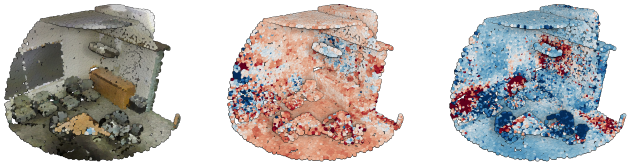
Table 2. Average spatial and color distance between all successful attacks and their corresponding original neighborhoods. *Hausdorff* corresponds to the bi-directional Hausdorff distance, and *Std.* refers to the standard deviation.

| Class | Target Architecture | Source Architecture | Success Rate XYZ | RGB |
|---|---|---|---|---|
| Chair | PTv3 | KP-FCNN | 0.59 | 0.21 |
| | | PointNet++ | 0.0 | 0.0 |
| | KP-FCNN | PTv3 | 0.0 | 0.63 |
| | | PointNet++ | 0.0 | 0.0 |
| | PointNet++ | PTv3 | 1.0 | 1.0 |
| | | KP-FCNN | 1.0 | 0.83 |
| Door | PTv3 | KP-FCNN | 0.0 | 0.0 |
| | | PointNet++ | 0.04 | 0.0 |
| | KP-FCNN | PTv3 | 0.06 | 0.96 |
| | | PointNet++ | 0.07 | 0.04 |
| | PointNet++ | PTv3 | 0.95 | 0.92 |
| | | KP-FCNN | 1.0 | 0.88 |

Table 3. Transferability of adversarial examples.

#### 4.4.3. Transferability

To assess the transferability of the generated adversarial examples between the studied architectures, we employ the following process: First, we select an architecture as an *attack target*. Then, we use the other two architectures as *attack sources*. Next, we input all generated adversarial examples for the source architecture into the target one, registering the success rate for the target on the new inputs. Tab. 1 shows the results of the transferability tests. PointNet++ shows the highest vulnerability, with adversarial examples from PTv3 and KP-FCNN having a success rate between 80 and 100% for all classes and target features. In contrast, the transferability of adversarial examples to PTv3 and KP-FCNN is comparatively low, with a success rate of or close to 0% in many cases. For both classes, KP-FCNN shows a high susceptibility to adversarial examples from PTv3 on the color features and a strong robustness against adversarial examples on spatial features. PTv3 displays success rates close to 0% in most cases - only showing higher success rates on adverasrial examples from KP-FCNN for the *Chair* class.



(a) original (b) XYZ (c) RGB

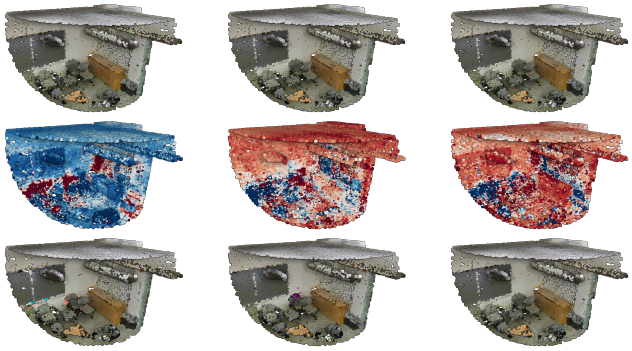
Figure 6. XYZ and RGB IG saliency maps of a sample PTv3 input of the *Chair* class. Blue: Positive attribution value. Red: Negative attribution value.

## 5. Discussion

In this work, we extend CTA [30] into CTA-seg for the 3D point cloud semantic segmentation domain. We use this method to study three aspects of adversarial attacks on 3D point cloud segmentation models: 1) Perturbation sensitivity, 2) Perturbation degree, and 3) Transferability.

Our results show that, of the studied architectures, PTv3 and KP-FCNN are considerably more robust to perturbations of spatial than color features, requiring the shift of a considerably larger amount of target points to produce successful attacks. This could have two causes: First, it could indicate that the models give higher weight to color features in their prediction. Second, it could be that the pattern recognition mechanisms of the models are more robust and generalizable for spatial features. These results corroborate the findings by Xu et al. [38] on the higher sensitivity to perturbations on the color features. However, our results and the results of [38] are not directly comparable, since our evaluation considers the change of *model predictions* compared to the initial predictions, while [38] study the performance degradation w.r.t. the ground truth labels. To explore this behavior, we analyze the IG saliency maps for the spatial and color features. In most cases, we observe a higher number of positively attributed points for the color features (Fig. 6), which supports the first hypothesis that models give higher weights to color in their prediction. To further separate the influence of both aspects, future studies could test whether models that only use XYZ as input are more sensitive to attacks on spatial features.

Furthermore, our experiments show a difference in the robustness of PTv3 across individual color channels, suggesting that models might learn to prioritize or ignore individual color channels for specific objects. To further explore this behavior, Fig. 7 shows the saliency maps generated by IG for the individual color channels for the *Chair* class. We observe a much higher number of positively attributed points for the red channel, which points to a higher contribution of them towards the prediction. However, the high-attribution points are mostly located in distant and unrelated parts of the scene - whereas for the green and blue channels, the high-attribution points are much more focused



(a) red channel (b) green channel (c) blue channel

Figure 7. Per-color channel IG saliency maps for a sample adversarial attack on the $Chair$ class with PTv3. From top to bottom: original input, saliency map, adversarial example. Blue: Positive attribution value. Red: Negative attribution value.

on similar regions. This behavior might indicate that the red channel does not provide enough information to the models, thus affecting the capability of IG to identify critical points.

For the PointNet++ architecture, both attacks on XYZ coordinates and RGB values achieve an almost perfect success rate. This might be because PointNet++ hierarchically aggregates local features using max-pooling layers, making it more vulnerable to outliers in the input point clouds. This is in line with the findings of Tan and Kotthaus [30], who find that max-pooling layers negatively impact the robustness of PointNet [21] against adversarial attacks. The higher robustness of PTv3 and KP-FCNN to perturbations of spatial features is also reflected in the degree of perturbation needed to produce successful attacks. The number of perturbed points and mean Hausdorff distances between original and adversarial examples are considerably large, meaning that the adversarial examples are outside of the learned data distribution, diminishing their value as counterfactual examples. Increasing the value of $\beta$ in the loss function might help to produce adversarial examples closer to the inputs; however, it is likely that the success rate would also decrease. For color features, on the other hand, the use of critical points allows us to generate counterfactual examples with a low degree of perturbation, making it easy to observe the points necessary to change the model predictions. However, it is not as intuitive to visualize color-based decision boundaries as it is with XYZ coordinates, as small color changes are often difficult to perceive. Research on visualization techniques for such counterfactual examples would further benefit their interpretability.

Regarding the transferability of the adversarial attacks, we observe that adversarial examples from PointNet++ are ineffective against KP-FCNN and PTv3. This is probably because attacks on PointNet++ contain considerably less perturbations than those for KP-FCNN and PTv3, potentially reducing their transferability. Furthermore, we observe that adversarial examples from KP-FCNN and PTv3 are effective on PointNet++. Xu et al. [38] observe a similar behavior, with attacks from ResGCN-28 [15] being successful for PointNet++. However, our experiments show that there is a low transferability of attacks between the KP-FCNN and PTv3 architectures. These results suggest that PointNet++ is generally vulnerable to input shifts; nevertheless, the transferability across different architectures is put into question. A larger transferability study covering a wider range of architectures would further help to shed light on this matter. Furthermore, such a study could also analyze the transferability between more similar architectures; e.g., different versions of Point Transformer [35, 36, 39].

**Known Limitations.** In this study, we evaluate a limited number of classes from an indoor dataset. Further research on a larger sample of classes and datasets is needed to ascertain the generalizability of our findings. Such experiments could provide further insight into the vulnerability of the models w.r.t. attacks on different classes and explore if observable patterns emerge (e.g., similar behavior for classes with similar characteristics). Research on adversarial attacks on the laser reflection intensity feature is also of interest since this feature is widely used in LiDAR 3D point cloud datasets [5, 14, 26]. A limitation of CTA-seg is that it relies on the saliency scores produced by an underlying saliency technique; in our case, IG. Thus, the success of our attack strategy does not solely depend on the attacked model but also on the performance of the saliency method. Further studies could evaluate CTA-seg with other saliency methods (e.g., pGS-CAM [13]). On the other hand, this opens the possibility of utilizing CTA-seg as an evaluation tool for the underlying saliency method.

## 6. Conclusions

By extending a point-shifting adversarial attack method from the 3D point cloud classification domain to semantic segmentation, this study explores the robustness of 3D point cloud segmentation models against adversarial examples. Our results show that all analyzed architectures are vulnerable to small perturbations of color features for even a small number of points, indicating a need to reduce the over-reliance of models on these features. Furthermore, our findings indicate a medium-to-low transferability of adversarial attacks between transformer- and convolution-based approaches. This suggests that the robustness against adversarial attacks may be improved by using ensembles of multiple architectures. Although further research is needed to fully ascertain the generalizability of our findings to a wider sample of objects and datasets, this work demonstrates the potential of adversarial attack approaches to explain the behavior of 3D point cloud semantic segmentation models.

## Acknowledgements

This work was partially funded through grants by the Systems Design Research School of the Hasso Plattner Institute and by the Federal Ministry of Education and Research, Germany through grant 033L305A ("TreeDigitalTwins") and grant 01IS22062 (AI research group "FFS-AI"). We thank the anonymous reviewers for their valuable feedback.

## References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 3

[2] American Psychological Association. Apa dictionary of psychology, 2018. 6

[3] Panagiotis Antoniadis. How to compute the similarity of colours, 2024. 4

[4] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543. IEEE, 2016. 4

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pages 9296–9306. IEEE, 2019. 4, 8

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 2

[7] Jürgen Döllner. Geospatial artificial intelligence: Potentials of machine learning for 3d point clouds and geospatial digital twins. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1):15–24, 2020. 4

[8] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. AdvPC: Transferable adversarial perturbations on 3D point clouds. In *Computer Vision – ECCV 2020*, pages 241–257. Springer, 2020. 2

[9] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[10] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8338–8354, 2022. 4

[11] Yuanmin Huang, Mi Zhang, Daizong Ding, Erling Jiang, Zhaoxiang Wang, and Min Yang. CausalPC: Improving the robustness of point cloud classification by causal effect identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19779–19789. IEEE, 2024. 1

[12] Jaeyeon Kim, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Minimal adversarial examples for deep learning on 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7797–7806. IEEE, 2021. 1, 2, 4

[13] Abhishek Kuriyal and Vaibhav Kumar. Towards explainable lidar point cloud semantic segmentation via gradient based target localization, 2024. 1, 8

[14] Michael Kölle, Dominik Laupheimer, Stefan Schmohl, Norbert Haala, Franz Rottensteiner, Jan Dirk Wegner, and Hugo Ledoux. The hessigheim 3d (h3d) benchmark on semantic segmentation of high-resolution 3d point clouds and textured meshes from uav lidar and multi-view-stereo. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 1:11, 2021. 8

[15] Guohao Li, Matthias Müller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 8

[16] Nan Li, Olaf Kähler, and Norbert Pfeifer. A comparison of deep learning methods for airborne lidar point clouds classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:6467–6486, 2021. 4

[17] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing*, pages 2279–2283. IEEE, 2019. 1, 2

[18] Daniel Liu, Ronald Yu, and Hao Su. Adversarial shape perturbations on 3D point clouds. In *Computer Vision – ECCV 2020 Workshops*, pages 88–104. Springer, 2020. 2

[19] Tianrui Lou, Xiaojun Jia, Jindong Gu, Li Liu, Siyuan Liang, Bangyan He, and Xiaochun Cao. Hide in thicket: Generating imperceptible and rational adversarial perturbations on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24326–24335. IEEE, 2024. 1, 2

[20] Marc Mahy, Luc Van Eycken, and André Oosterlinck. Evaluation of uniform color spaces developed after the adoption of cielab and cieluv. *Color Research & Application*, 19(2): 105–121, 1994. 6

[21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, pages 77–85. IEEE, 2017. 8

[22] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2, 3, 5

[23] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[24] Dario Rethage, Johanna Wald, Jürgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the 15th Eu-*

*ropean Conference on Computer Vision, Part IV*, pages 625–640. Springer International, 2018. 4

[25] Alan R. Robertson. The cie 1976 color-difference formulae. *Color Research & Application*, 2(1):7–11, 1977. 4

[26] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, 2018. 4, 8

[27] Markus Schwegler, Christoph Müller, and Alexander Reiterer. Integrated gradients for feature assessment in point cloud-based data sets. *Algorithms*, 16(7), 2023. 1, 3

[28] Yiming Sun, Feng Chen, Zhiyu Chen, and Mingjie Wang. Local aggressive adversarial attacks on 3D point cloud. In *Proceedings of The 13th Asian Conference on Machine Learning*, pages 65–80. PMLR, 2021. 2

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017. 3

[30] Hanxiao Tan and Helena Kotthaus. Explainability-aware one point attack for point cloud neural networks. In *Winter Conference on Applications of Computer Vision*, pages 4570–4579, 2023. 1, 2, 3, 4, 5, 7, 8

[31] Hugues Thomas. *Learning new representations for 3D point cloud semantic segmentation*. Theses, Université Paris sciences et lettres, 2019. 4

[32] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pages 6410–6419. IEEE, 2019. 1, 3, 5

[33] Hugues Thomas, Hubert Tsai, Timothy D. Barfoot, and Jian Zhang. Kpconvx: Modernizing kernel point convolution with kernel attention. In *CVPR*, 2024.

[34] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[35] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 8

[36] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 1, 3, 5, 8

[37] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9128–9136. IEEE, 2019. 1, 2, 4

[38] Jiacen Xu, Zhe Zhou, Boyuan Feng, Yufei Ding, and Zhou Li. On adversarial robustness of point cloud semantic segmentation. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 531–544, 2023. 1, 2, 3, 7, 8

[39] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 8

[40] Shijun Zheng, Weiquan Liu, Yu Guo, Yu Zang, Siqi Shen, Chenglu Wen, Ming Cheng, Ping Zhong, and Cheng Wang. Sr-adv: Salient region adversarial attacks on 3d point clouds for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 25(10):14019–14030, 2024. 1, 2, 4

[41] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1961–1970, 2019. 4

[42] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. Adversarial attacks against lidar semantic segmentation in autonomous driving. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, page 329–342, New York, NY, USA, 2021. Association for Computing Machinery. 1, 2