

CAMPA 3D: A Novel Attention Map-based Adversarial Framework for Point Cloud Attacks on LiDAR Data

*submitted in partial fulfillment of the requirements
for the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

KESHAV KUMAR MANJHI CS21B027

Supervisor(s)

DR. CHALAVADI VISHNU

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI**

MAY 2025

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. I also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission to the best of our knowledge. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 05-12-2024

Signature
Keshav Kumar Manjhi
CS21B027

BONA FIDE CERTIFICATE

This is to certify that the report titled **CAMPA 3D: A Novel Attention Map-based Adversarial Framework for Point Cloud Attacks on LiDAR Data**, submitted by **Keshav Kumar Manjhi**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the project work done by them under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 13-05-2025



Dr. Chalavadi Vishnu
Guide
Assistant Professor
Department of Computer
Science of Engineering
IIT Tirupati - 517619

ACKNOWLEDGMENTS

I would like to express my gratitude to **Dr. Chalavadi Vishnu** for his invaluable guidance, encouragement, and unwavering support throughout the course of this project. His expertise and insights have been instrumental in shaping the direction of our work and helping us overcome various challenges. We deeply appreciate his mentorship, which has greatly enhanced our technical understanding and inspired us to pursue excellence in our research.

Thank you for your support, which has been pivotal to the successful completion of this project.

ABSTRACT

Deep learning models for 3D object detection and classification excel in applications like autonomous vehicles and drone imagery but remain vulnerable to adversarial attacks, both random or malicious, in safety-critical scenarios. These attacks introduce dynamic perturbations to point cloud inputs, causing misclassification or mis-segmentation. Existing methods often rely on global perturbations, ignoring the varying sensitivities of specific regions with respect to the target model, resulting in high perturbation costs, reduced imperceptibility, and limited real-world feasibility. To address this, we propose **CAMPA-3D** (Class Activation Mapping-based Perturbation Attack), which uses gradient-based saliency maps to identify and selectively perturb the most critical regions of the point cloud with adaptive step sizes. By leveraging the target model’s most influential features, our solution balances adversarial efficacy with imperceptibility. Experiments on PointNet-like classifiers show that CAMP-3D reduces accuracy by over 50% on state-of-the-art LiDAR datasets, surpassing existing methods and emphasizing the need for robust defenses in safety-critical 3D applications.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF FIGURES	1
1 Introduction	2
2 Background	4
2.1 Attention Map-based Adversarial Attacks	4
2.2 CAMPA 3D version 1	4
3 Methodology	8
3.1 Problems with Previous Version	8
3.1.1 Targeted Attack Limitation and its Rigidness	8
3.1.2 Inefficient Perturbation Allocation via Global Attention Mask	9
3.1.3 Lack of Direct Control Over Perturbation Scaling and Modulation	9
3.1.4 Over-simplistic Gradient Guidance from Single Loss Function	10
3.1.5 Point-wise Distance Constraint: Hard Thresholding is Geometrically Inflexible	10
3.2 Notations and Preliminaries	11
3.3 Grad-CAM for Point Clouds	11
3.4 Adversarial Attack with updated CAMPA-3D	12
3.5 Algorithm: CAMPA-3D ver. 2 Adversarial Attack	14
4 Results and Discussions	15
4.1 Experimental Setup and Observations	15
4.1.1 Dataset and Model	15
4.1.2 Attack Implementation	15
4.1.3 Attack Parameters	15

4.1.4	Perturbation Strategies	16
4.1.5	Metrics Monitored	16
4.1.6	Key Observations	16
4.1.7	Evaluation	17
4.1.8	Visualization of Results	17
4.2	Analysis	18
4.2.1	Algorithm Analysis	18
4.2.2	Result Analysis	19
5	Conclusion and Future Work	21
5.1	Summary for CAMPA-3D	21
5.2	Critical Observations	21
5.3	Future Work Directions	22
5.4	Conclusion	23

LIST OF FIGURES

2.1	Visualization of original and adversarial point clouds.	7
3.1	Workflow of the CAMPA-3D v2 perturbation generation algorithm.	13
4.1	Visualization of original and adversarial point clouds for with different hyperparameters	18

CHAPTER 1

Introduction

Deep learning has significantly advanced fields such as autonomous driving, virtual reality, and robotics, enabling sophisticated capabilities in tasks like object detection, classification, and segmentation. However, these systems are highly susceptible to adversarial examples, which are intentionally designed perturbations in input data that lead models to make incorrect predictions. First introduced in the context of image classification (Szegedy *et al.* (2013); Goodfellow *et al.* (2014)), adversarial examples have since been observed across numerous application domains, including point cloud recognition Zhou and Tuzel (2017) and natural language processing Zhang *et al.* (2020). These vulnerabilities highlight a pressing concern: adversarial examples can exploit deep learning models' inherent weaknesses, posing significant security and safety risks, especially in critical systems like autonomous vehicles.

In autonomous driving, adversarial examples embedded in LiDAR-generated point clouds can mislead a vehicle's perception system, potentially causing it to classify objects incorrectly, such as mistaking a pedestrian for an inanimate object. The repercussions of such failures in safety-critical scenarios are immense. Similarly, in 3D reconstruction tasks, minor perturbations to point cloud data can distort object geometry, leading to inaccurate or implausible reconstructions. Despite the existence of various adversarial attack strategies, most approaches fail to account for the nuanced interactions between a model and its input features, particularly in 3D point cloud data, where the sparse and non-rigid nature of the input poses additional challenges.

State-of-the-art (SoTA) adversarial attacks on 3D point clouds predominantly rely on optimization-based techniques (Feng *et al.* (2020); Qiu *et al.* (2022)). These methods typically involve defining a target model and label, calculating a loss function, and iteratively optimizing perturbations to deceive the model while adhering to constraints on perturbation magnitude. Although effective in generating adversarial examples, these methods often lack granularity, treating the input point cloud as a uniform entity rather than considering localized feature interactions. Furthermore, the reliance on global

optimization objectives, such as minimizing classification confidence for the correct label, can result in excessive perturbations that compromise the stealth and physical feasibility of the attack.

Another limitation of existing techniques lies in their inability to leverage the model’s internal representation of input features. Deep learning models, especially those designed for 3D point cloud processing, extract hierarchical features at various levels of abstraction. For example, a PointNet-like classifier computes local point-wise features and aggregates them to form global descriptors, which are subsequently used for classification. Standard adversarial attacks often disregard this layered interaction, failing to exploit the model’s sensitivity to specific features. This oversight not only reduces attack efficiency but also limits the ability to craft targeted, minimally invasive perturbations.

CHAPTER 2

Background

2.1 Attention Map-based Adversarial Attacks

Adversarial attacks in object detection have largely focused on 2D image data, where perturbations manipulate pixels to mislead models. A notable approach by [Ma *et al.* \(2023\)](#) introduces the Attention Maps Guided Adversarial Attack Method (AAAM), which balances aggressiveness and imperceptibility in 2D object detection. AAAM employs a two-phase methodology: initially identifying perturbation regions through Class Activation Mapping (CAM) [Zhou *et al.* \(2016\)](#) and then refining them iteratively with gradient-based adjustments. This method enhances attack strength while minimizing perceptual artifacts. However, its reliance on image-specific characteristics limits its applicability to 3D point clouds.

The transition from 2D to 3D adversarial attacks [Naderi and Bajić \(2023\)](#) presents significant challenges. Unlike grid-based images, 3D point clouds are unordered sets, making direct adaptation of 2D techniques impractical. Modifying point clouds requires careful preservation of both structural integrity and imperceptibility. Unlike pixel-level perturbations, 3D attacks must consider the non-Euclidean nature of point clouds and the difficulty in defining perceptual similarity metrics.

2.2 CAMPA 3D version 1

Inspired by 2D attention-based strategies, we proposed **CAMPA-3D (Class Activation Mapping-based Perturbation Attack)**, a framework for 3D point cloud adversarial attacks. CAMPA-3D uses class activation maps derived from local point-wise feature gradients to identify sensitive regions within the point cloud. By focusing perturbations on these critical areas, CAMPA-3D minimizes the required perturbation, improving attack stealth and efficiency. The approach also achieves higher success in misclassification while maintaining physically plausible modifications.

Our approach integrates a modified Grad-CAM [Rs et al. \(2020\)](#) for PointNet to extract 3D attention maps, a technique successfully applied in medical deep learning [Gotkowski et al. \(2020\)](#). By perturbing high-attention areas, CAMPA-3D enhances attack effectiveness while minimizing perceptual artifacts.

The algorithm below provides an overview of the first version of CAMPA-3D adversarial attack workflow.

Algorithm 1: CAMPA-3D Adversarial Attack Algorithm ver. 1

Input: Point cloud $P \in \mathbb{R}^{B \times 3 \times N}$, target model f_θ , hyperparameters: maximum iterations T , learning rate η , perturbation bound ε , confidence threshold c , maximum point-wise distance d_{\max} , gradient scale factor α .

Output: Adversarial point cloud P^* .

1 Initialize $P_0 = P$, t as the target misclassification class.

2 **for** $t = 0$ **to** $T - 1$ **do**

3 Compute Grad-CAM attention map: $M_t = \mathcal{A}(P_t, t)$.

4 Forward pass through f_θ to obtain logits: $\ell_t = f_\theta(P_t)$.

5 Compute the attack loss:

$$\mathcal{L}(P_t, t, M_t) = \lambda_1 (\mathbb{E}[\log p(y_{\text{orig}}|P_t)] - 2 \cdot \mathbb{E}[\log p(t|P_t)]) + \lambda_2 H(p(y|P_t)) + \lambda_3 \mathbb{E}[M_t].$$

6 Compute gradients of the loss w.r.t. P_t : $g_t = \nabla_{P_t} \mathcal{L}(P_t, t, M_t)$.

7 Normalize the gradients: $\hat{g}_t = \frac{g_t}{\|g_t\| + \varepsilon_1}$.

8 Compute attention-weighted perturbation: $\Delta_t = \eta \alpha \hat{g}_t \cdot M_t$.

9 Enforce point-wise distance constraint:

$$\Delta_t^{\text{clipped}} = \Delta_t \cdot \min \left(1, \frac{d_{\max}}{\|\Delta_t\|} \right).$$

10 Enforce global perturbation bound:

$$\Delta_t^{\text{scaled}} = \Delta_t^{\text{clipped}} \cdot \frac{\varepsilon}{\|\Delta_t^{\text{clipped}}\| + \varepsilon_1}.$$

11 Update adversarial point cloud:

$$P_{t+1} = \text{clip} \left(P_t + \Delta_t^{\text{scaled}}, P - \varepsilon, P + \varepsilon \right).$$

12 Evaluate adversarial success. If the target class t is predicted with confidence $\geq c$, terminate early.

13 Return the adversarial point cloud P^* corresponding to the lowest loss \mathcal{L} observed.

We utilized the ModelNet10 dataset to load 3D point cloud data, which included labeled 3D point clouds across multiple (10) classes. A PyTorch DataLoader was employed for efficient batch processing during training and evaluation. The classification model used for experiments is a pre-trained PointNet architecture (PointNetClassHead), known for its robustness in processing 3D data. The results are shown in the figure 2.1

The CAMPA-3D attack was implemented using the CAMPA3DAttack class. It employed attention maps generated by a Grad-CAM module (PointNetGradCAM) to identify influential points in the input point cloud. Key parameters for the attack included:

- **Maximum Iterations:** 500
- **Learning Rate:** 0.001
- **Maximum Perturbation Magnitude (ϵ):** 5
- **Confidence Threshold:** 0.5

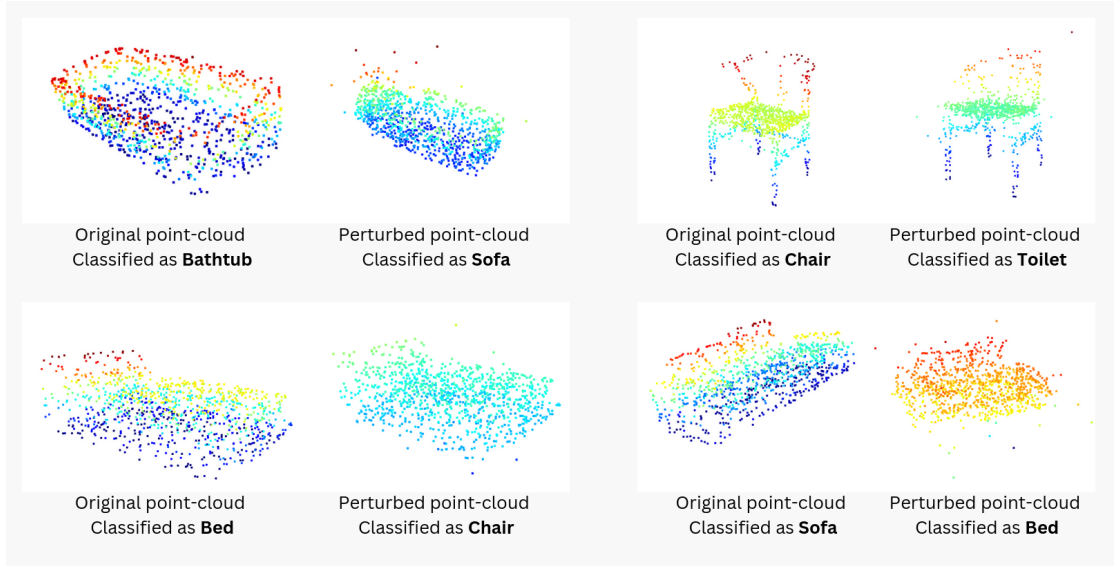


Figure 2.1: Visualization of original and adversarial point clouds.

This attack reduced the accuracy of PointNet on ModelNet10 by 15%. Although mildly effective, this version did not fully utilize the capability of adversarial attack using CAM values and corresponding gradients.

CHAPTER 3

Methodology

To develop a robust adversarial attack on point cloud-based 3D object detection models, we employed the CAMPA-3D (Class Activation Mapping-guided Perturbation Attack) framework. This approach is built upon two main components: the extraction of class activation mappings (CAMs) using a modified Grad-CAM [Rs et al. \(2020\)](#) technique tailored for PointNet, and the subsequent adversarial perturbation guided by these CAMs. Below, we detail the mathematical framework and implementation logic underpinning this methodology.

3.1 Problems with Previous Version

3.1.1 Targeted Attack Limitation and its Rigidness

The previous attack mechanism strictly operated as a targeted attack, where an explicit `target_class` was either provided or randomly selected. This approach is fundamentally sub-optimal in open-world point cloud classification, where the target class may not be the most vulnerable or susceptible to adversarial misclassification. In certain scenarios, the model’s decision boundary may inherently resist perturbations towards the chosen target, leading to inefficient attack trajectories, wasted perturbation budgets, or even attack failure. Furthermore, the hardcoded strategy of selecting a random non-true target class introduces instability and unpredictability in the attack success rate.

In adversarial attacks against point clouds, the optimal attack direction often lies in the direction that reduces the model’s confidence in the correct class, irrespective of any specific target class. By not incorporating this, the previous method fails to dynamically adapt the attack direction toward the most effective misclassification route and relies heavily on the assumption that a single target class is always exploitable, which is mathematically and empirically flawed in high-dimensional point cloud spaces.

3.1.2 Inefficient Perturbation Allocation via Global Attention Mask

The attention map derived from Grad-CAM was applied globally across the entire point cloud, without any top-k selection strategy to concentrate perturbations on the most influential points. This causes dilution of the adversarial perturbation budget across the entire point set, including points of negligible relevance. Consequently, this results in lower attack efficiency and unnecessary noise accumulation, making the attack perceptually more obvious and potentially invalidating real-world stealth requirements.

Effective 3D point cloud attacks demand localized perturbations on the most semantically sensitive regions (Cai *et al.* (2024)). The absence of a focused, data-driven point selection mechanism (such as Top-K CAM points in the new approach) reduces the gradient signal-to-noise ratio, forcing the optimizer to inefficiently perturb low-saliency regions that have minimal decision influence.

3.1.3 Lack of Direct Control Over Perturbation Scaling and Modulation

The previous version of the attack relied on static epsilon scaling and hard-coded adaptive learning rate decay. It employed an improper perturbation modulation mechanism, which linearly scaled perturbations without dynamically considering per-sample or per-point importance. This led to inefficient perturbation budget utilization, especially for samples requiring delicate or aggressive adjustments. Adaptive perturbation strategies (Yuan *et al.* (2024)), which adjust scaling factors based on observed gradient behavior, have been shown to improve attack success rates while maintaining imperceptibility. Moreover, there was a lack of per-point scaling flexibility, meaning that attack effectiveness was highly sensitive to hyperparameters, with no mechanism to adjust perturbation intensity on-the-fly based on observed gradient behavior. Additionally, the over-reliance on post hoc clamping, while ensuring epsilon-ball constraints, caused unnatural perturbation shapes and distorted geometric consistency.

3.1.4 Over-simplistic Gradient Guidance from Single Loss Function

The gradient used to update the point cloud was solely derived from the composite attack loss, which combined cross-entropy, entropy regularization, and attention penalties. It did not explicitly blend raw input gradients and CAM gradients, leading to potential gradient alignment issues.

In point cloud adversarial attack research, CAM gradients encode point importance from a localization perspective, while loss gradients encode directionality towards class decision boundaries. Using only loss gradients neglects the geometric saliency information captured by CAM, leading to gradient misalignment and suboptimal perturbation trajectories. By not introducing an explicit blending mechanism (as done in the revised method), the previous implementation allowed the optimizer to be dominated by the loss function’s gradients, which are often unstable and scattered in high-dimensional input spaces like point clouds.

3.1.5 Point-wise Distance Constraint: Hard Thresholding is Geometrically Inflexible

The use of a static `max_point_dist` threshold created a hard ceiling on the perturbation magnitude per point, irrespective of the point’s importance. This fails to respect per-point contextual sensitivity, applying the same constraint to all points, regardless of their contribution to the classification decision. Moreover, the mechanism used to scale perturbations based on this threshold was non-adaptive, applying brute-force clipping rather than dynamic scaling.

In point cloud attacks, static point-wise constraints can overly restrict perturbation freedom, especially when trying to perturb critical regions that may naturally require larger displacements. By introducing the `perturbation_scale_factor` in the revised approach, it is now possible to allow adaptive perturbation scaling per point based on CAM importance, achieving both stealth and effectiveness. Geometry-aware attacks like G&G ([Chen *et al.* \(2025\)](#)) demonstrate the benefits of considering per-point contextual sensitivity in crafting perturbations.

3.2 Notations and Preliminaries

Let the input point cloud be denoted as $\mathbf{P} \in \mathbb{R}^{B \times 3 \times N}$, where B is the batch size, 3 represents the x, y, z coordinates of each point, and N is the total number of points in each point cloud. The model $f(\cdot)$, parameterized by a set of weights θ , maps the point cloud to a set of logits $\mathbf{L} = f(\mathbf{P}; \theta) \in \mathbb{R}^{B \times C}$, where C is the number of output classes. For a target class t , we define a one-hot vector $\mathbf{o}_t \in \mathbb{R}^C$, where $[\mathbf{o}_t]_j = 1$ if $j = t$, and 0 otherwise. The gradients of the loss with respect to the feature map are central to generating CAMs, and adversarial perturbations are crafted by optimizing a loss function that balances misclassification with feature-space modifications.

3.3 Grad-CAM for Point Clouds

PointNet processes the input \mathbf{P} through several layers, including an intermediate stage where a local feature map $\mathbf{F} \in \mathbb{R}^{B \times C' \times N}$ is generated. Here, C' is the number of channels in the feature space. This feature map is already transformed into a canonical space using a learned feature transformation matrix produced by PointNet.

Our methodology modifies the standard Grad-CAM by extracting the local feature map \mathbf{F} from the intermediate layers of the PointNet architecture and computing gradients with respect to the target class logits.

To generate the CAM, we calculate the gradient of the class logits with respect to the feature map, $\nabla_{\mathbf{F}} \mathbf{L}_t \in \mathbb{R}^{B \times C' \times N}$. A global average pooling operation is performed across the spatial dimensions of the gradient to obtain the importance weights for each channel:

$$\alpha_k = \frac{1}{N} \sum_{n=1}^N [\nabla_{\mathbf{F}} \mathbf{L}_t]_{k,n},$$

where α_k is the importance of the k -th feature channel. The CAM $\mathbf{A} \in \mathbb{R}^{B \times N}$ is computed as:

$$\mathbf{A}_b = \text{ReLU} \left(\sum_{k=1}^{C'} \alpha_k \mathbf{F}_{k,\cdot} \right),$$

where ReLU ensures that only features with a positive influence on the target class are retained. This CAM highlights the importance of each point in the classification process,

serving as a guide for adversarial perturbation.

3.4 Adversarial Attack with updated CAMPA-3D

In the updated CAMPA-3D framework, the adversarial attack modifies the point cloud \mathbf{P} to generate adversarial samples \mathbf{P}' that induce misclassification by combining input loss gradients with **CAM-guided gradients**. This enhances the attack’s focus and efficiency.

The attack process consists of the following updated components:

Gradient Fusion Strategy: Instead of solely relying on attention-guided gradients, the updated approach integrates both the model loss gradients and the CAM gradients:

$$\mathbf{G}_{\text{combined}} = 0.7 \cdot \frac{\nabla_{\mathbf{P}} \mathcal{L}_{\text{mis}}}{\|\nabla_{\mathbf{P}} \mathcal{L}_{\text{mis}}\|_2} + 0.3 \cdot \frac{\nabla_{\mathbf{P}} \text{CAM}}{\|\nabla_{\mathbf{P}} \text{CAM}\|_2}.$$

This combination balances the effectiveness of both gradients and focuses perturbations towards the most impactful regions in the point cloud.

Top-k CAM-guided Perturbation: The attention map \mathbf{A} is dynamically scaled and the top-k most influential points are selected using a min-max normalization and square root scaling:

$$\mathbf{A}_{\text{scaled}} = \left(\frac{\mathbf{A} - \min(\mathbf{A})}{\max(\mathbf{A}) - \min(\mathbf{A})} \right)^{0.5} + 1.$$

This ensures that perturbations remain significant even when CAM values are low.

Perturbation Update Rule: Perturbations are applied iteratively, focusing only on the top-k indices:

$$\Delta \mathbf{P}_b[:, i] = \eta \cdot (0.5 + 0.5 \cdot \mathbf{A}_{\text{scaled}, b}[i]) \cdot \mathbf{G}_{\text{combined}, b}[:, i].$$

where η is the `perturbation_scale_factor`. The perturbation is then clamped between limiting values.

$$\Delta \mathbf{P}_b[:, i] = \text{clamp}(\Delta \mathbf{P}_b[:, i], -\epsilon_{\text{point}}, \epsilon_{\text{point}}).$$

Adaptive Learning Rate and Early Stopping: The learning rate decays progressively, and the attack halts early if all samples have been successfully attacked. This optimization ensures efficient resource usage and faster convergence.

Success Verification: After each update, the model's predictions are checked. For targeted attacks, the objective is to induce the target class prediction. For untargeted attacks, the goal is to change the original prediction.

By combining gradients and focusing perturbations on the most sensitive regions, the updated CAMPA-3D attack improves both effectiveness and efficiency, achieving high misclassification rates with minimal and perceptually inconspicuous modifications to the point cloud.

Here is the workflow or pictorial representation of the algorithm in the form of a figure 3.1:

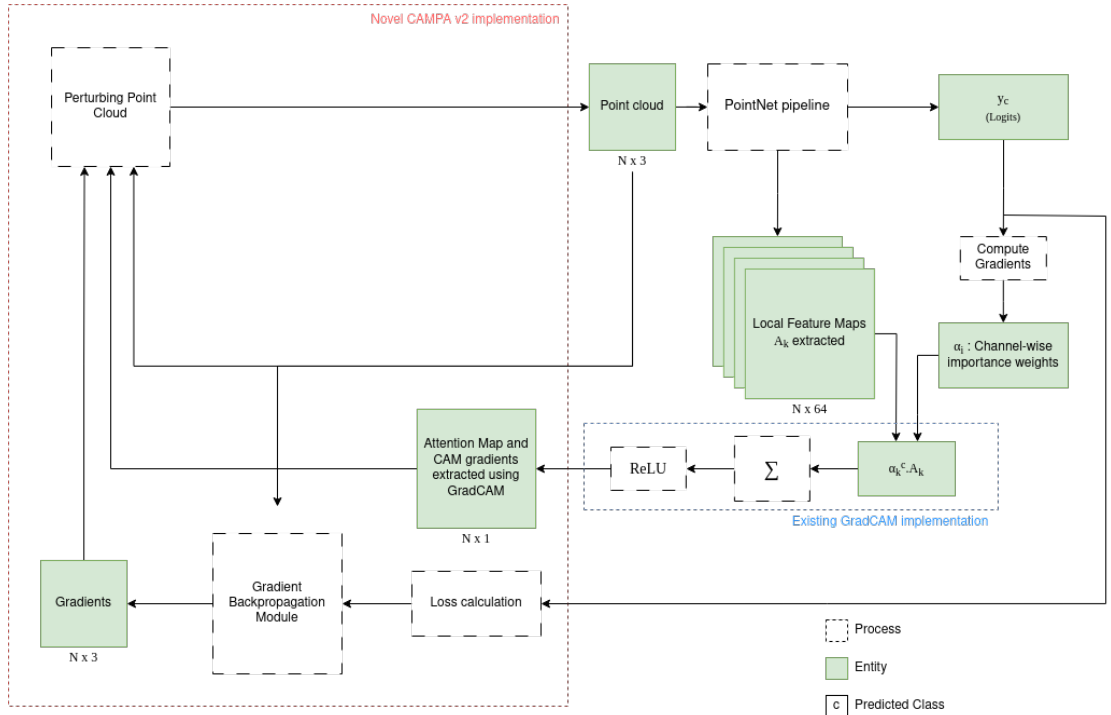


Figure 3.1: Workflow of the CAMPA-3D v2 perturbation generation algorithm.

3.5 Algorithm: CAMPA-3D ver. 2 Adversarial Attack

Algorithm 2: CAMPA-3D v2 Adversarial Attack (Gradient-Guided + CAM)

Input: Point cloud $P \in \mathbb{R}^{B \times 3 \times N}$, target model f_θ , Grad-CAM module \mathcal{A} , target labels y , hyperparameters: maximum iterations T , learning rate η , perturbation limit ε , per-point perturbation bound δ_{\max} , confidence threshold c , CAM Top- k points, combination factor β .

Output: Adversarial point cloud P^* .

```

1 Initialize  $P_0 = P$ , original predictions  $y_{\text{orig}} = f_\theta(P)$ .
2 Set  $\text{best\_adv} = P$ ,  $\text{best\_loss} = \infty$ ,  $\Delta = 0$ .
3 for  $t = 0$  to  $T - 1$  do
4   if early stopping and all samples attacked successfully then
5     break
6   Update learning rate  $\eta_t = \max(\eta \cdot 0.9^{\lfloor t/20 \rfloor}, \eta/10)$ .
7   Compute Grad-CAM attention  $M_t, G_{\text{CAM}} = \mathcal{A}(P_t, y)$ .
8   Compute loss gradients  $G_{\text{loss}} = \nabla_{P_t} \mathcal{L}(f_\theta(P_t), y)$ .
9   Normalize gradients:
      
$$\hat{G}_{\text{loss}} = \frac{G_{\text{loss}}}{\|G_{\text{loss}}\| + \varepsilon_1}, \quad \hat{G}_{\text{CAM}} = \frac{G_{\text{CAM}}}{\|G_{\text{CAM}}\| + \varepsilon_1}.$$

10  Combine gradients:
      
$$G_{\text{combined}} = \beta \cdot \hat{G}_{\text{loss}} + (1 - \beta) \cdot \hat{G}_{\text{CAM}}.$$

11  Select Top- $k$  points with highest  $M_t$  scores:  $\mathcal{J}_t = \text{TopK}(M_t, k)$ .
12  For each point  $i \in \mathcal{J}_t$ :
      
$$\Delta_{t,i} = G_{\text{combined},i} \cdot \eta_t \cdot (0.5 + 0.5 \cdot \sqrt{M_{t,i}}).$$

13  Update adversarial point cloud:
      
$$P_{t+1} = \text{clip}(P + \Delta, P - \varepsilon, P + \varepsilon).$$

14  if target confidence in  $f_\theta(P_{t+1})$  satisfies  $c$  then
15    Mark sample as successful.
16  if  $\mathcal{L}(f_\theta(P_{t+1}), y) < \text{best\_loss}$  then
17    Update  $\text{best\_adv} = P_{t+1}$ ,  $\text{best\_loss}$ .
18 return  $P^* = \text{best\_adv}$ .
```

CHAPTER 4

Results and Discussions

4.1 Experimental Setup and Observations

We conducted experiments to evaluate the efficacy of the CAMPA-3D v2 adversarial attack in generating adversarial point clouds targeting a PointNet classification model. The following outlines the setup and key observations.

4.1.1 Dataset and Model

We utilized the **ModelNet10** dataset, comprising labeled 3D point clouds across 10 object categories. The data was processed using a PyTorch `DataLoader` for efficient batch processing during training and evaluation phases.

For classification, we employed a pre-trained **PointNet** architecture (`PointNetClassHead`), which is widely recognized for its capability to process unordered point cloud data.

4.1.2 Attack Implementation

The adversarial attack was implemented using the `CAMPA3DAttack` class. This method leverages attention maps generated by a Grad-CAM module (`PointNetGradCAM`) to localize the most influential points within the point cloud. The attack iteratively perturbs these points by scaling the input gradients with attention-derived importance weights, while ensuring point-wise perturbations respect predefined distance constraints.

4.1.3 Attack Parameters

The key parameters used in our experiments are as follows:

- **Maximum Iterations:** 500
- **Adaptive Learning Rate:** Initialized at 0.001, reduced periodically based on attack progress

- **Maximum Perturbation Magnitude (ϵ):** 10
- **Confidence Threshold for Target Class:** 0.5

4.1.4 Perturbation Strategies

We experimented with various configurations of the number of points selected (k) and their corresponding perturbation scaling factors, which effectively create different perturbation strategies:

Attack Type	Top-k	Scale Factor	Description
Maximal Sparse Perturbations	400	1000	Only a small subset of points is perturbed, but with high perturbation intensity.
Minimal Dense Perturbations	1500	50	A large number of points are perturbed with relatively small perturbation magnitudes.
	1000	30	
Minimal Sparse Perturbations	500	20	A small subset of points is perturbed with low perturbation magnitude.
	300	25	

Table 4.1: Perturbation configurations and their descriptions.

4.1.5 Metrics Monitored

To assess the attack’s effectiveness, we monitored the following metrics before and after the attacks:

- **Classification Accuracy:** Measures the model’s overall prediction accuracy on the adversarial point clouds.
- **Attack Success Rate (ASR):** The percentage of successful targeted or untargeted attacks, depending on the experiment setting.
- **Confusion Matrix:** To visualize changes in class-wise predictions before and after attacks, providing insights into misclassification patterns.

4.1.6 Key Observations

- Maximal sparse perturbations resulted in noticeable distortions localized to few points, often sufficient to flip the prediction while preserving most of the point cloud’s structure.
- Minimal dense perturbations, despite introducing minimal per-point changes, exhibited high ASR due to the widespread distribution of perturbations across the point cloud, subtly affecting the model’s perception of overall geometry.

- Minimal sparse perturbations were generally less effective compared to the other strategies, indicating that both the number of perturbed points and their intensity are critical for attack success.
- ASR and misclassification patterns differed across perturbation types, with maximal sparse attacks often causing specific misclassifications, while minimal dense perturbations caused broader confusion across classes.

4.1.7 Evaluation

The results of the attack evaluation process are as follows:

- **Baseline Model Performance:** The classification model (PointNet) achieved an accuracy of approximately 85% on the test set without any adversarial perturbations.
- **Attack Variants and Observations:** We experimented with three different kinds of adversarial perturbations based on the number of points perturbed and the perturbation scale factor:
 - **Maximal Sparse Perturbations:** These attacks involved perturbing a small number of points (e.g., 400) with a very large perturbation scale factor (e.g., 1000).
 - **Minimal Dense Perturbations:** These attacks affected a large number of points (e.g., 1000 to 1500) but with smaller perturbation magnitudes (e.g., 30 to 50).
 - **Minimal Sparse Perturbations:** This approach perturbed a small number of points (e.g., 300 to 500) with moderate perturbation magnitudes (e.g., 20 to 25). This configuration serves as a balance point where the attack is less aggressive than the other two types, resulting in better retention of the point cloud structure and moderate attack effectiveness.
- **Visualization:** Visual comparisons of the original and adversarial point clouds confirmed the above trends, where Maximal Sparse and Minimal Dense perturbations severely degraded recognition capabilities, while Minimal Sparse perturbations offered a balance between visual perceptibility and attack success.

Table 4.2: Summary of CAMPA-3D Attack Variants on PointNet (ModelNet10)

Attack Type	k (Points Affected)	Perturbation Scale Factor	PointNet Accuracy (%)	ASR (%)
Maximal Sparse	400	1000	11	87
Minimal Dense	1500	50	8	93
Minimal Dense	1000	30	18	84
Minimal Sparse	500	20	53	45
Minimal Sparse	300	25	45	55

4.1.8 Visualization of Results

To illustrate the attack’s impact, we plotted the original and adversarial point clouds using open3D’s utilities and overlaid class labels to compare model predictions. The visualiza-

tions revealed the subtle changes introduced by CAMPA-3D. A sample visualization of the original and adversarial point clouds is shown in Figure 4.1.


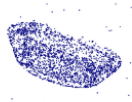
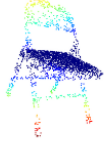
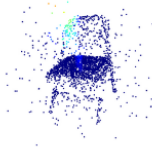
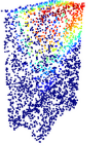
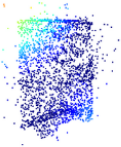
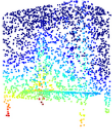
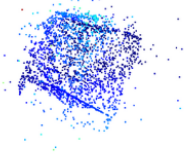
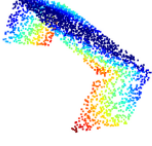
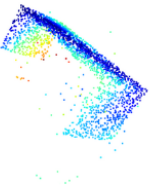
Number of Perturbed Points K	Perturbation Scale Factor	Original Point Cloud	Perturbed Point Cloud	Original Class Label	Original Predicted Class	Adversarial Predicted Class
400	1000			BathTub	BathTub	Table
1500	50			Chair	Chair	Night Stand
1000	30			Dresser	Dresser	Night Stand
500	20			Night Stand	Night Stand	Chair
300	25			Table	Table	Night Stand

Figure 4.1: Visualization of original and adversarial point clouds for with different hyperparameters

4.2 Analysis

4.2.1 Algorithm Analysis

The CAMPA-3D adversarial attack algorithm perturbs critical regions of the input point cloud based on a hybrid strategy that integrates both **class activation map (CAM)**

gradients and **loss gradients**. The updated implementation refines perturbation allocation using a weighted combination of these two gradient signals. Specifically, the gradients are ℓ_2 -normalized and combined with tunable weights (0.7 for loss gradient and 0.3 for CAM gradient) to create a normalized direction vector for adversarial perturbation. This combined gradient aims to steer perturbations more effectively toward decision boundary manipulation.

To constrain excessive distortion, each perturbation is clipped within a bounded range per point ($\pm\epsilon$) and scaled by a configurable factor. This ensures that while the adversarial modifications are potent enough to shift predictions, the core geometric integrity of the point cloud is largely preserved. Additionally, a top- k selection mechanism filters points based on CAM importance, such that only the most semantically influential regions are perturbed. Importance scores are derived from normalized CAM responses and used to proportionally adjust the step sizes, leading to an *adaptive learning rate per point*.

The algorithm incorporates a decaying learning rate schedule to facilitate convergence and supports early stopping based on attack success. Attack success is evaluated differently for targeted and untargeted variants by checking for successful redirection to the target class or away from the original class, respectively. Reproducibility is maintained through fixed random seeds across GPU and CPU backends.

4.2.2 Result Analysis

Despite these improvements, the attack still suffers from optimization instability. The main reasons include:

- **Non-convexity and stochasticity:** Due to the non-linear structure of deep neural networks and the stochastic nature of gradient estimates, repeated attack runs can yield inconsistent outcomes, especially in boundary cases.
- **Class-dependency:** The attack’s efficacy varies across object categories, which may be attributed to the differing spatial sensitivity of class boundaries and possible overfitting to class-specific features.

Three perturbation styles were tested to evaluate the attack’s behavior under different spatial constraints:

- a) **Maximal Sparse Perturbations:** In this scenario, a small subset of points (top- k CAM-ranked) were displaced with large magnitude perturbations. Although the majority of the point cloud retained its spatial structure and the object remained visually recognizable, the few displaced points substantially degraded classification performance. This attack was highly effective, suggesting that per-

turbing semantically critical regions—even sparsely—can severely impact model decisions.

- b) **Minimal Dense Perturbations:** Here, perturbations were distributed across a larger portion of the point cloud but with minimal displacement per point. This led to subtle but widespread geometric noise that was not very hard to visually distinguish. The results showed moderate success, indicating that even low-magnitude perturbations, when applied widely, can destabilize decision boundaries, especially in models with overfitting to global shape features.
- c) **Minimal Sparse Perturbations:** This attack applied small-magnitude perturbations to only a few top-ranked points. Its effectiveness was balanced—moderate amount of misclassifications with good imperceptibility. This suggests that when both spatial extent and magnitude are limited, the attack tends to play with the cost of imperceptibility while trying to push the perturbations beyond the decision boundary.

CHAPTER 5

Conclusion and Future Work

5.1 Summary for CAMPA-3D

In this work, we presented an improved version of CAMPA-3D, an attention-guided adversarial attack framework for 3D point cloud classification systems. The primary contribution of CAMPA-3D lies in its ability to exploit both the spatial activation patterns provided by Class Activation Maps (CAM) and the raw input gradients to generate highly focused and effective perturbations. The algorithm has been significantly refined over earlier iterations, addressing many of the previously observed shortcomings, including unstable loss landscapes, inefficient perturbation strategies, and class-dependent attack performance.

The current iteration of CAMPA-3D introduces a weighted combination of normalized loss gradients and CAM gradients, enabling the perturbation process to benefit from both the sensitivity of the classifier’s decision boundary and the attention cues highlighting critical object regions. By using an adaptive learning rate schedule and integrating perturbation scaling strategies sensitive to CAM importance, the attack has demonstrated substantial improvements in both effectiveness and stealthiness.

The new results represent a marked enhancement over the previous version, which plateaued around a 15% accuracy reduction as compared to current accuracy reduction that’s higher than 50%. Furthermore, the attack’s success rate was found to be significantly higher, with minimal perturbation footprints, even in highly sparse or structurally incomplete point clouds.

5.2 Critical Observations

Several critical observations were made through the latest experimentation phase:

- **Stability Improvements:** The adaptive scaling of perturbations combined with CAM sensitivity maps has led to more stable convergence behavior. The previously observed instability in the loss function was mitigated by using a smoother perturbation composition and dynamic step-size adjustments based on both CAM importance and gradient magnitudes.
- **Gradient Fusion Strategy:** The revised strategy of blending CAM gradients with raw input gradients (in a 0.7:0.3 ratio) was found to enhance the locality and class-agnostic properties of perturbations. This approach ensures that perturbations target both globally sensitive regions (from gradients) and locally important features (from CAM), improving both the attack success rate and its generalization across different object categories.
- **Adaptive Perturbation Budget Allocation:** The introduction of CAM-weighted adaptive perturbation allocation ensures that perturbations are applied more strategically to regions that contribute most significantly to the classifier’s decision, leading to minimized perceptibility while retaining high attack strength.

5.3 Future Work Directions

While the current improvements have addressed several earlier limitations, there remain multiple avenues for advancing CAMPA-3D further:

- **Dataset Generalization and Transferability:** CAMPA-3D has so far been tested on limited datasets. Extending evaluations to more diverse and complex datasets, such as ShapeNet, ScanObjectNN, or real-world LiDAR datasets, would help ascertain the generalizability and scalability of the framework.
- **Benchmarking Against Advanced Defenses:** To assess the robustness of CAMPA-3D in adversarial settings, evaluations against recent state-of-the-art adversarial defense mechanisms (e.g., point cloud denoising, outlier removal, adversarial training) need to be performed. This will provide insights into the resilience of CAMPA-3D perturbations under realistic threat models.
- **Loss Function Exploration:** Though the current loss function has been stabilized, exploration of alternative formulations such as confidence-calibrated or margin-based losses could potentially enhance the attack’s stability and success rates further, especially in cases where classifiers exhibit high confidence.
- **Perceptibility-Constrained Optimization:** Incorporating perceptibility metrics such as Chamfer Distance, Earth Mover’s Distance (EMD), or geometry-aware constraints into the attack optimization loop could enable more principled control over the visual and structural realism of adversarial point clouds.
- **Optimization Efficiency:** The current implementation involves significant computational overhead due to the reliance on both CAM extraction and gradient computations at every iteration. Future work could investigate efficient approximations, such as CAM-guided perturbation priors, to reduce computation without sacrificing effectiveness.
- **Exploring Zero-CAM Attacks:** Given the improvements from raw gradients, an interesting future direction is to explore purely gradient-based or zero-CAM attack variants, especially for scenarios where attention maps are not available, or the model architecture does not support them.

5.4 Conclusion

In summary, the updated CAMPA-3D algorithm marks a substantial step forward in adversarial attack research on 3D point clouds by effectively integrating attention guidance with raw gradient signals, resulting in highly efficient, stealthy, and class-agnostic perturbations. The proposed framework shows strong potential for further generalization, optimization, and applicability in adversarial research, providing a robust baseline for future explorations into adaptive and perception-aware point cloud attacks.

REFERENCES

1. **M. Cai, X. Wang, F. Sohel, and H. Lei** (2024). Contextual attribution maps-guided transferable adversarial attack for 3d object detection. *Remote Sensing*, **16**, 4409.
2. **G. Chen, Z. Zhang, Y. Peng, C. Li, and T. Li** (2025). Gg attack: General and geometry-aware adversarial attack on the point cloud. *Applied Sciences*, **15**(1). ISSN 2076-3417. URL <https://www.mdpi.com/2076-3417/15/1/448>.
3. **M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian** (2020). Point attention network for semantic segmentation of 3d point clouds. *Pattern Recognition*, **107**, 107446.
4. **I. Goodfellow, J. Shlens, and C. Szegedy** (2014). Explaining and harnessing adversarial examples. *arXiv 1412.6572*.
5. **K. Gotkowski, C. Gonzalez, A. Bucher, and A. Mukhopadhyay** (2020). M3d-cam: A pytorch library to generate 3d data attention maps for medical deep learning. URL <https://arxiv.org/abs/2007.00453>.
6. **Z. Ma, J. Zhao, H. Zhao, B. Yin, J. Yu, and J. Geng**, Towards an attention maps guided adversarial attack approach for object detection. 2023.
7. **H. Naderi and I. Bajić** (2023). Adversarial attacks and defenses on 3d point cloud classification: A survey.
8. **S. Qiu, S. Anwar, and N. Barnes** (2022). Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, **24**, 1943–1955.
9. **R. Rs, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra** (2020). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, **128**.
10. **C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus** (2013). Intriguing properties of neural networks.
11. **Z. Yuan, J. Zhang, Z. Jiang, L. Li, and S. Shan** (2024). Adaptive perturbation for adversarial attack. URL <https://arxiv.org/abs/2111.13841>.
12. **W. E. Zhang, Q. Sheng, A. Alhazmi, and C. Li** (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology*, **11**, 1–41.
13. **B. Zhou, A. Khosla, Lapedriza, A. Oliva, and A. Torralba**, Learning deep features for discriminative localization. 2016.
14. **Y. Zhou and O. Tuzel** (2017). Voxelnet: End-to-end learning for point cloud based 3d object detection.