7. Supplementary Material

This section is a supplement for the main part of the paper. In this section, we detail additional formulas for the backgrounds (7.1), demonstrate our Pseudo-codes and stopping criteria (7.2), show more adversarial examples for both OPA and CTA respectively (7.3), visualize the diversity of attacking labels (7.5), discuss the most appropriate hyper-parameter settings (7.7). We also present the attack result OPA on 2D images as a comparable reference (7.8). We provide visualisations of the Activation Maximization (AM) and more attribution distribution of PC networks (7.9 and 7.10 respectively). Finally, we discuss the societal impacts and ethical issues (7.11).

7.1. Background

7.1.1 Point cloud deep neural networks

A PC input can be represented as $P = \{p_0, ..., p_n\}$, where $p_i \in \mathbb{R}^3$ and n is the number of component points. Compared with 2D images, the structural peculiarity of PC data lies in the irregularity: let $R(S)$ be a function that randomly disrupts the order of the sequence S, a PC classifier f must possess such properties: $f(P) = f(R(P))$, which is regarded as a "symmetric system". The pioneer of PC networks is proposed by [33], succeeded by employing an symmetric function $g(S)$ and an element-wise transformer $h(p)$ where $f(P) \approx g(\{h(p_0), ..., h(p_n)\})$ (in their experiments a max-pooling is choosen as $g(S)$). PointNet++ [34], the successor of PointNet, further coalesced hierarchical structures by introducing spatial adjacency via grouping of nearest-neighbors. DGCNN [50] extended the the predecessors by dynamically incorporating graph relationships between multiple layers. All of the point-based methods achieve satisfactory accuracies on acknowledged PC dataset such as ModelNet40 [53].

7.1.2 Integrated Gradients

Gradients-based explainability methods are oriented on generating saliency maps of inputs by calculating gradients during propagation. While vanilla gradients severely suffer from attribution saturation [45], [46] proposes IG which accumulates attributions from an appropriate baseline before the gradients reach the saturation threshold. IG is formulated as:

$$IG_i = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x} d\alpha \quad \text{(S1)}$$

Where x' denotes the given baseline.

7.1.3 Targeted vs. Untargeted attack

For a given classifier f and its logits a, an PC input instance P and an adversarial perturbation A_p:

- *Targeted attack*

$$\text{Minimize} \quad (a[f(P)+A_p]) \quad \text{s.t.} \quad f(P+A_p) \neq f(P) \quad \text{(S2)}$$

- *Untargeted attack*

$$\text{Maximize} \quad (a[f(P+A_p)]) \quad \text{s.t.} \quad f(P+A_p) = T \quad \text{(S3)}$$

Where T is the given target class.

7.1.4 Activation Maximization (AM)

Activation Maximization (AM), first proposed by [13], sets out to visualize a global explanation of a given network through optimizing the input matrix x while freezing all parameters θ such that the selected i^{th} activation neuron at l^{th} layer S_i^l is maximized [31]:

$$x^* = \underset{x}{\operatorname{argmax}} \left(a_i^l(\theta, x)\right) \quad \text{(S4)}$$

7.2. Other implementation details

7.2.1 Stopping Criteria

Theoretically, CTA can keep searching until all positively contributed points are traversed. For algorithmic efficiency, we set specific stopping criteria for OPA and CTA.

OPA: With the introduction of Gaussian random noise for OPA, the optimization process may fall into an everlasting convergence-noise addition loop, a manually configured failure threshold is therefore essential. A recorder R_a is built to record the corresponding prediction activation for each period. We set a global maximum iterations I_{maxg}. The stopping criterion of OPA is fulfilled when

- $I_{cur} > I_{maxg}$ **or** $((Mean(R_a^{k+1}) > Mean(R_a^k)$ **and** $Var(R_a^k) \to 0))$.

Due to the introduction of random Gaussian noise, the optimization process will not fail until the target activation has no fluctuant reaction to the Gaussian noise.

CTA: There are both local and global stopping criteria for CTA. Local criterion stands for terminating the current N_p perturbed points and start the $N_p + 1$ round, which is similar with OPA. Again, we set an activation recorder R_a and a **local** maximum iterations I_{maxl}. The local stopping criterion is fulfilled when:

- $I_{cur} > I_{maxl}$ **or** $Mean(R_a^{k+1}) > Mean(R_a^k)$

Model	PointNet	PointNet++	DGCNN	PointMLP
% of Lg.	41.2%	13.3%	13.4%	8.8%

Table S1. Legality of adversarial examples generated by different models. $\%\,of\ Lg.$ represents the percentage of the legal adversarial examples to the total.

Global stopping terminates the optimization of the current instance and registers it as "failed". CTA is designed to shift all the positively attributed points N_{pos} in the worst case which is extremely time-consuming. For practical feasibility, we specify the **global** maximum iterations I_{maxg}. The global stopping criterion for CTA is fulfilled when:

- $I_{cur} > I_{maxg}$ **or** $N_p \geqslant N_{pos}$

where N_{pos} is the total amount of positive attributed points according to the explanation provided by IG.

7.3. More qualitative visualizations for OPA and CTA

We selected 10 representative classes from Modelnet40 that occur most frequently in the real world and demonstrate another 10 adversarial examples for each class generated by OPA and CTA in Fig. S1 and S2 respectively. The perturbed points are colored with red for better visualization. As the success rate of the OPA attack is close to 100%, in order to distinguish the results of CTA from OPA more clearly, we set β in CTA as $(8 \times \alpha)$. This setting makes a good trade-off between success rate, shifting distance and perturbation dimensionality. The detailed experimental results are demonstrated in section 7.7.

7.4. Legality of adversarial examples

In this section we analyze the legitimacy of the generated adversarial examples. For most point cloud datasets, the input coordinate values are normalized to a specific interval to facilitate the prediction of objects at different scales. For ModelNet40, all instances are normalized to the interval of $[-1, 1]$. Therefore, we regard those adversarial examples that are still within the original interval as "legal" and otherwise as "illegal". As the results shown in Table S1, for PointNet, a significant portion of the adversarial examples are still legal. For the rest of the models, however, the vast majority of the examples are outside the legal range. Again, this reveals that PointNet is less robust as grouping modules of adjacent points are missing.

7.5. Label Diversity of adversarial examples

For non-targeted OPA and CTA, the optimization process diminishes the neurons corresponding to the original labels, with no interest in the predicted labels of the adversarial examples. However, we found that observing the adversarial labels helped to understand the particularities of the adversarial examples. Fig. S3 and S4 report the label distribution matrices of untargeted OPA and CTA respectively. As can be seen from Fig. S3, class "radio" is most likely to be the adversarial label, and most of the adversarial examples generated within the same class are concentrated in one of the other categories (e.g. almost all instances from "door" are optimized towards "curtain"). This phenomenon is significantly ameliorated in CTA (see Fig. S4). The target labels are more evenly distributed in the target label matrix, yielding more diversity in the adversarial examples.

7.6. Transferability of random seeds

Random sampling is involved in numerous point cloud networks. Since our method requires very few points to be perturbed, it raises concerns about the transferability of different random sampling seeds. In this subsection we study the impact of random seeds on attack performances. We consider only two networks, PointNet++ and PointMLP, both of which employ Farthest Point Sampling (FPS) [34, 27] (PointNet and DGCNN do not contain this module). First, the above two models are attacked with the random seed r_1. We then predict the generated adversarial examples utilizing the random seeds of $r_{2,3..n}$ ($r_1 \neq r_2 \neq ... \neq r_n$) and record the success rates. Five different seeds are experimented and the results are reported in Table S2. According to the results, PointNet++ and PointMLP attacks almost fail on themselves if the random seeds alter. We consider the reason to be that the original sampled points become adjacent ones or even unrelated to the classification due to the seed variation.

7.7. Hyper-parameter settings

7.7.1 Distance regularization β

For both proposed algorithms, there are two crucial hyper-parameters to be tuned that affect the performance of the attacks, i.e. α and β. α indicates the optimization rate and is empirically set to $1e-6$. β indicates the penalty of perturbation distances, which regularizes the shifting magnitude and preserves the imperceptibility of adversarial examples. In previous experiments, we temporarily set β to 0 to highlight the sparse perturbation dimensions. However, additional investigations suggest that appropriate $beta$ can further improve the performance of the proposed approaches. Fig. S5 demonstrates the performances with different β settings. Interestingly, we found that CTA performs best when $\beta = \alpha$: while maintaining nearly 100% success rate and comparably shifting distances, its average N_p dramatically decreases to 3.04 (different from OPA, CTA employs no random-noise). We strongly recommend restricting β to a reasonable range ($\leq (8 \times \alpha)$) since large β easily leads to an explosion in processing time.

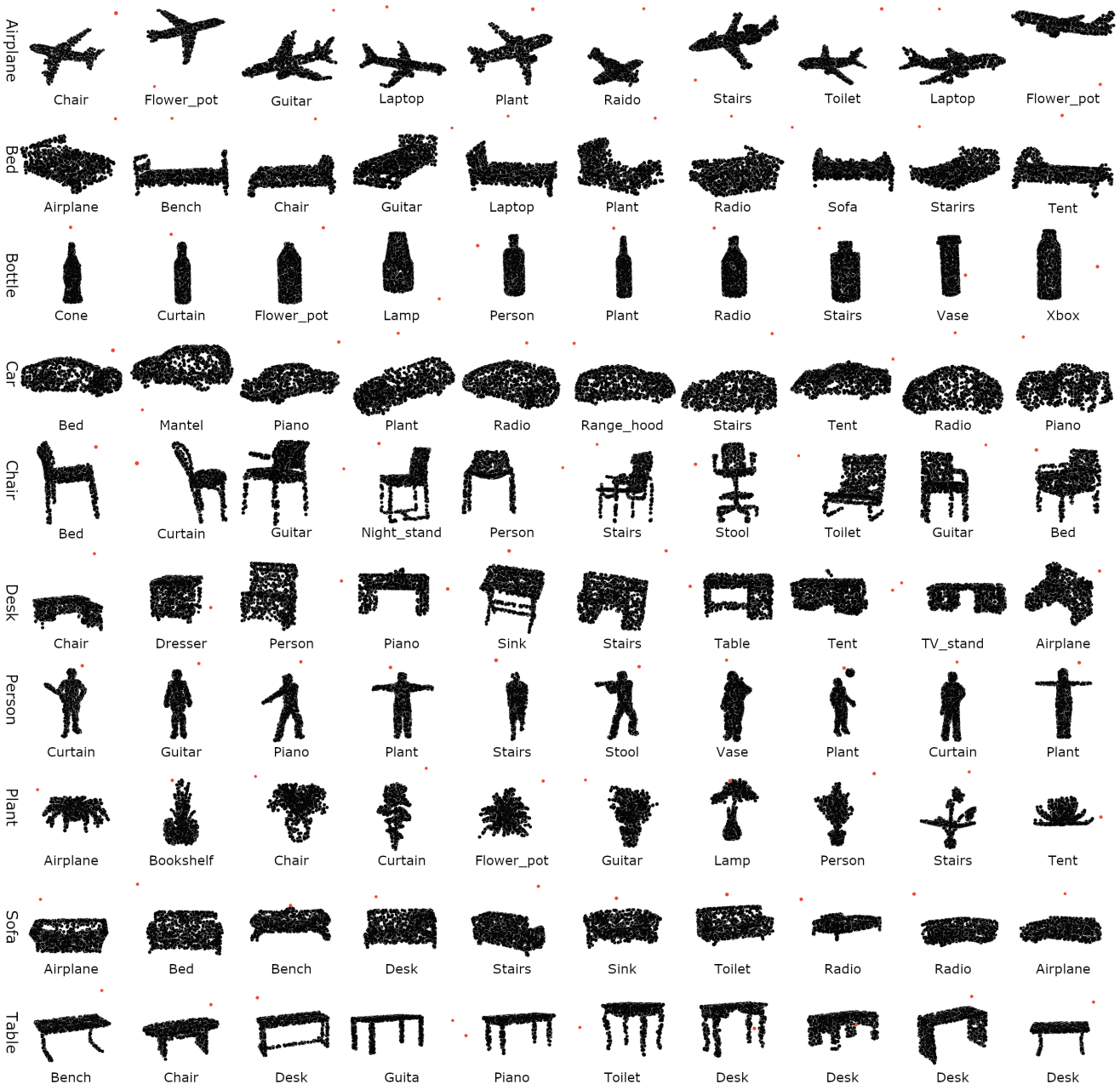

Figure S1. More results from OPA. We chose the 10 representative classes that appear more frequently in the real world. The perturbed points are indicated in red to be noticeable.

	PN	PN++	DGCNN	PointMLP
Acc.%	100	$9.4 \pm 1.15 \times 10^{-3}$	100	$17.8 \pm 1.35 \times 10^{-3}$

Table S2. Mean transferability of random seeds for FPS. Note that only PointNet++ and PointMLP utilize FPS.

7.7.2 Gaussian noise weight W_n for OPA

In particular for OPA, another hyperparameter W_n is set to prevent the optimization process from stagnating at a local optimum. We experimented with various settings of W_n and present the results in Fig. S6. What stands out in the figure is that the appropriate range for W_n is around 10^{-1} to $10^{-0.5}$ where the success rate approximates 100% while maintaining acceptable perturbation distances. Adding Gaussian noise in the optimization process dramatically enhances the attack performance of OPA, with its success rate increasing from 56.1% as a simple-gradient attack

Figure S2. More results from CTA. We also chose the 10 representative classes that appear more frequently in the real world. The perturbed points are indicated in red to be noticeable.

to almost 100%. Interestingly, we observe that a suitable noise weight concurrently reduces the perturbation distance and thus augments the imperceptibility of the adversarial examples. We attribute this to the promotion of Gaussian noise that facilitates the optimizer to escape from saddle planes or local optimums faster, reducing the number of total iterations. However, overweighting deviates the critical point from the original optimization path, which is equivalent to resetting another starting position in 3D space and forcing the optimizer to start iterating again. While there remains a high probability of finding an adversarial example, its imperceptibility is severely impaired.

7.8. OPA on 2D image neural network

We extend our OPA to 2D image neural networks for a rough comparison of its sensitivity to critical points with that of 3D networks. We trained a simple ResNet18 network with the MNIST handwriting dataset, which achieves

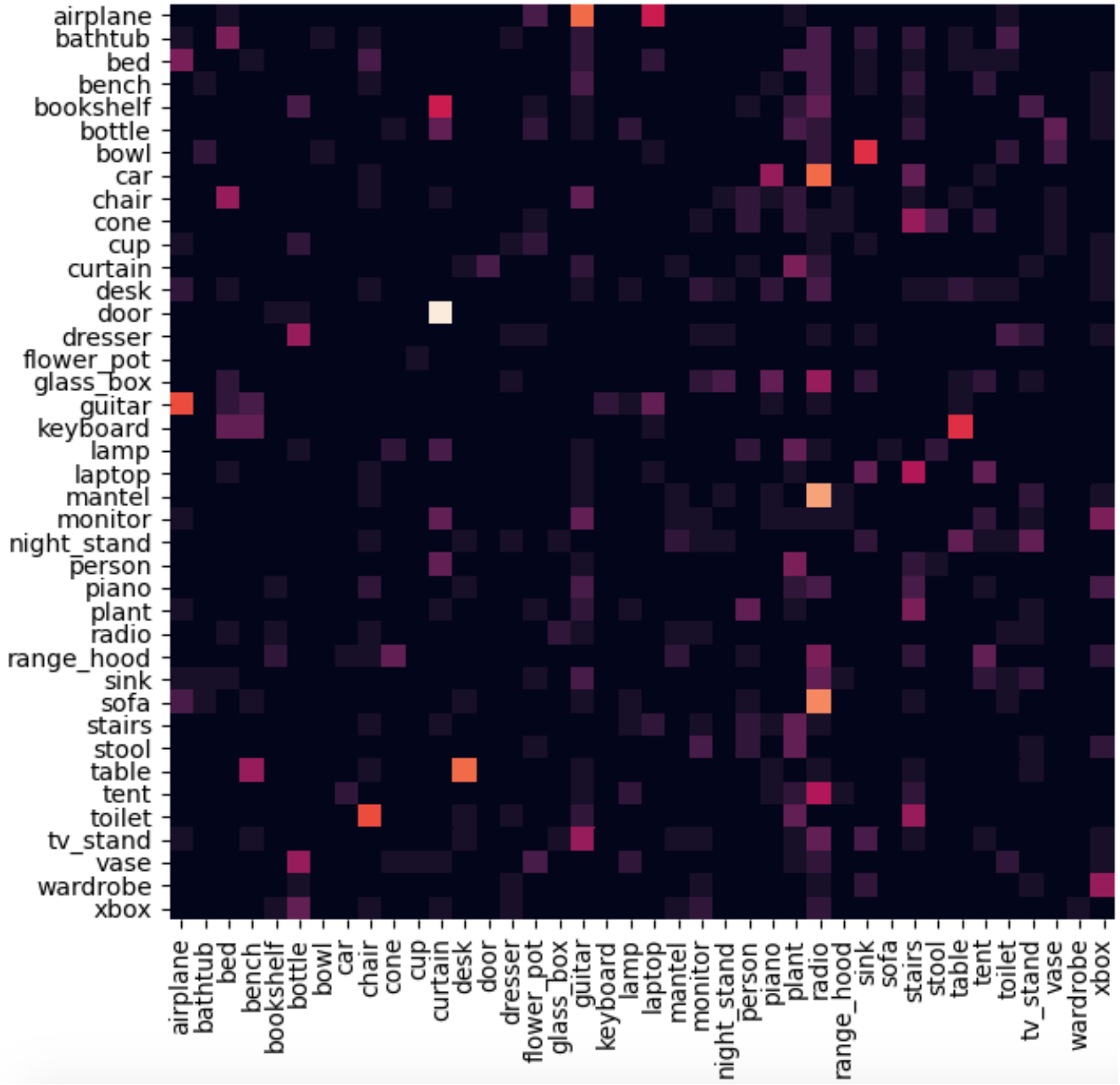

Figure S3. Heat map of successful attacks by OPA across labels. Rows indicate from which category the adversarial examples come and the columns indicate to which category they are predicted. The brighter the square, the more examples that fall into the corresponding category.

an accuracy of 99% on the test set. We select 1000 samples from the test set as victims to be attacked with OPA. The quantitative results and parts of the adversarial examples are demonstrated in table S3 and Fig. S7 respectively. In Fig. S7, the original instances and their adversarial results are listed on the first and the second row respectively. With the removal of a pixel in a critical location, a small number of test images successfully fooled the neural network. However, from a quantitative viewpoint (table S3), shifting one critical point almost fails to fool the ResNet18 network (1.2% success rate for ResNet18-LG). We believe the reasons are: (1) 2D images are restricted within the RGB/greyscale space, thus there exists an upper bound on the magnitude of the perturbation, while 3D point clouds are infinitely extendable; (2) Large-size convolutional kernels (≥ 2) learn local features of multiple pixels, which mit-

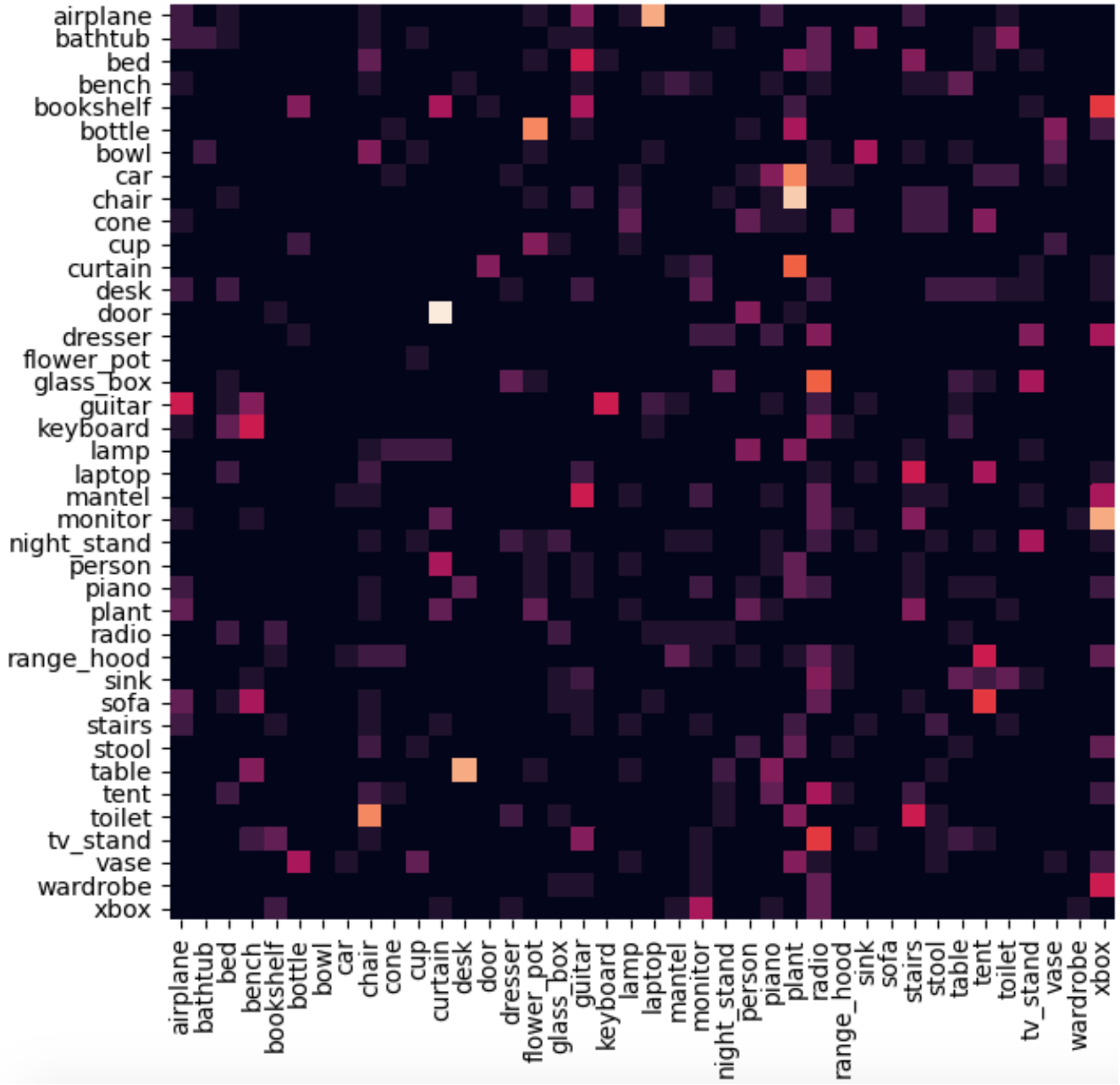

Figure S4. Heat map of successful attacks by CTA across labels. Rows indicate from which category the adversarial examples come and the columns indicate to which category they are predicted. The brighter the square, the more examples that fall into the corresponding category.

igates the impact of individual points on the overall prediction. According to observation (1), we temporarily remove the physical limitation during attacks to investigate the pure mechanism inside both networks and report the results in ResNet18-AL of table S3. Though the attack success rate climbs to 51.7%, there is still a gap with PointNet (98.7%). Even with the "legality" restriction, OPA still maintains a success rate of 41.2% on PointNet. PointNet encodes points with 1×1 convolutional kernels, which is analogous to an independent weighting process for each point. The network inclines to assign a large weight to individual points due to the weak local correlation of adjacent points and therefore leads to vulnerable robustness against perturbations of critical points.

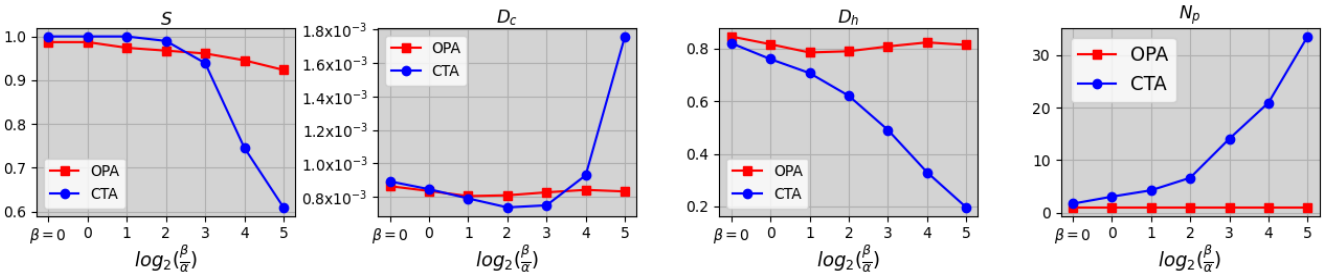

Figure S5. Performance (success rate, Chamfer and Hausdorff distances and the number of shifted points respectively) of OPA and CTA in different settings of hyper-parameters. The x-axis indicates the logarithm of the quotient of β and α where the first tick denotes $\beta = 0$.

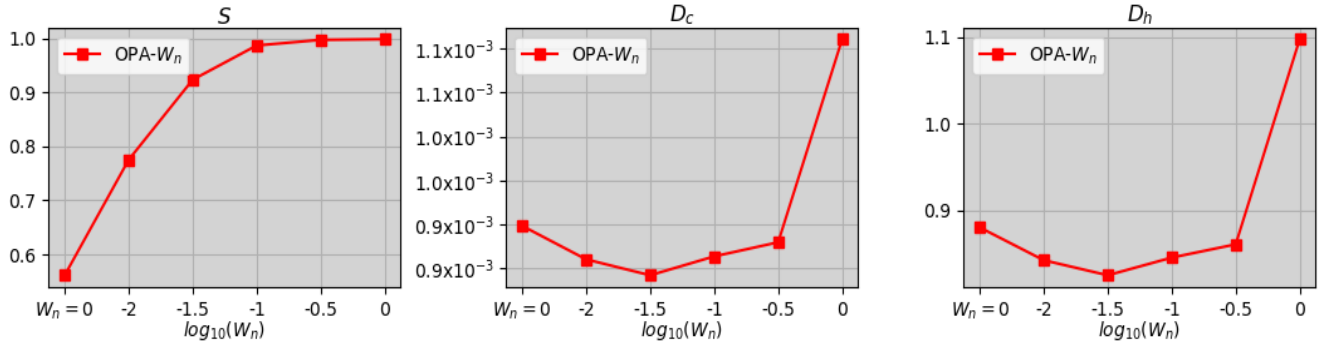

Figure S6. Performance (success rate, Chamfer and Hausdorff distances respectively) of OPA in different settings of weights for Gaussian noise. The x-axis indicates the logarithm of W_n where the first tick denotes $W_n = 0$.

	S	D_c	D_h
ResNet18-LG	1.2	4.93×10^{-2}	8.67×10^{-1}
ResNet18-AL	51.7	1.48	4.08×10^{1}
PointNet-AL	98.7	8.64×10^{-4}	8.45×10^{-1}
PointNet-LG	41.2	/	/

Table S3. OPA attack performance comparisons between ResNet18 and PointNet. ResNet18-LG indicates the "legal" attack within the range restriction of the greyscale value ($0 \sim 255$), while ResNet18-AL indicates a purely numerical attack possibly with no legality restriction. PointNet-LG and PointNet-LG denote legal and illegal attacks, respectively, as in Section 7.4.

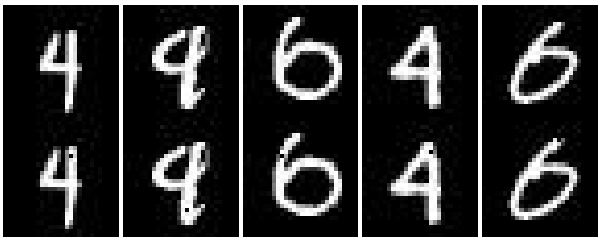

Figure S7. Successful attack examples of ResNet18-LG by OPA. The first and second rows are input images and adversarial examples respectively.

7.9. Maximized activation

The proposed OPA was motivated by a fruitless AM (see Sec 7.1.4 for introduction) attempt for PC networks. Fig. S8 displays an example from 1000-steps AM results of PointNet. More examples with different initializations are depicted in Fig. S10. We conduct the AM experiments with various initializations including zero, point cluster generated by averaging all test data [30] and a certain instance from the class "Car". What stands out in the visualization is that the gradient ascent of the PC neural network's activations appears to depend solely on the magnitude of the outward extension subject to the extreme individual points (the middle figure). We further investigate the explanations of the AM generations utilizing IG and the analysis reveals that almost all the positive attributions are concentrated on the minority points that were expanded (the right figure). Fig. S9 provides a quantitative view of how target activation ascends with the shifting of input points and we introduce Gini coefficient [12] to represent the "wealth gap" of the Euclidean distance among all points. Interestingly, as the target activation increments over the optimization process, the Gini coefficient of Euclidean distances steepens to 1 within few steps, indicating that the fastest upward direction of the target activation gradient corresponds with the extension of a minority of points.

For fairness and persuasion, we also conduct AM experiments with various initializations. Fig. S10 shows AM initialized with zeros and the point cluster generated by averaging all test data [30].

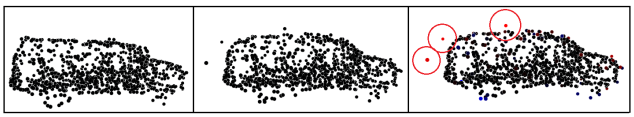

Figure S8. AM results initialized with a certain instance. The first, second and third columns demonstrate the initialized set of points, the AM output results after 1000 optimization steps and the salience map explanation of the corresponding output explained by IG, respectively. Note that the majority of positively attributed points (bright red) are exactly the expanded ones.

	r_D	r_P
OPA	98.6	100
CTA	99.2	45.6

Table S4. Success rates of defense against the proposed attacks by outlier removal. r_D denotes the success rate of the input instance being detected as adversarial examples and r_P denotes the percentage of perturbation points detected by the defense module correctly.

7.10. Visualization of the attribution distributions

As a supplementary of table 6, we demonstrate the complete pie diagrams of the attribution distributions of the aforementioned four pooling structures in S11.

7.11. Societal impacts and ethical issues

This work proposes two adversarial approaches, which pose a potential threat to the security of PC networks. Motivationally, however, this paper aims to illuminate the distribution of attributions of PC networks rather than specifically targeting the attack method of the model. Practically, our proposed approaches can be more easily defended visually or algorithmically compared to related studies aiming at imperceptibility. Table S4 presents the results of defending against the proposed attacks by a simple outlier removal algorithm. Adversarial samples generated by OPA and CTA are detected with almost 100% success rate. We thus argue that the proposed attacks do not pose a serious threat to existing networks.

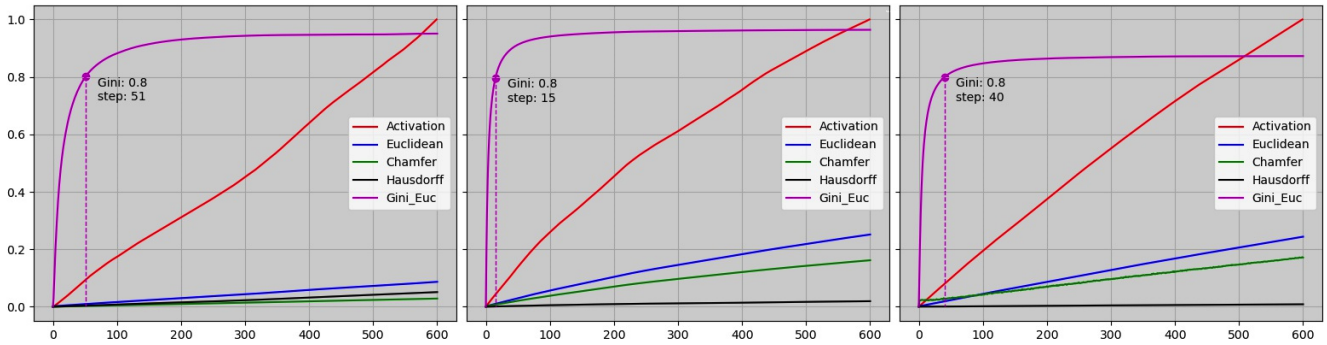

Figure S9. Correlation between the ascending target activation and the various distances of the optimized example from the original initializations: zero (left), the average of the test set (middle) and a certain instance (right). Activations are normalized in order to be visible together with other factors. X-axis denotes the optimization steps and y-axis denotes corresponding values in the legend. The marked points are the steps in the optimization process where the Gini coefficient of the attribution first reaches 0.8.

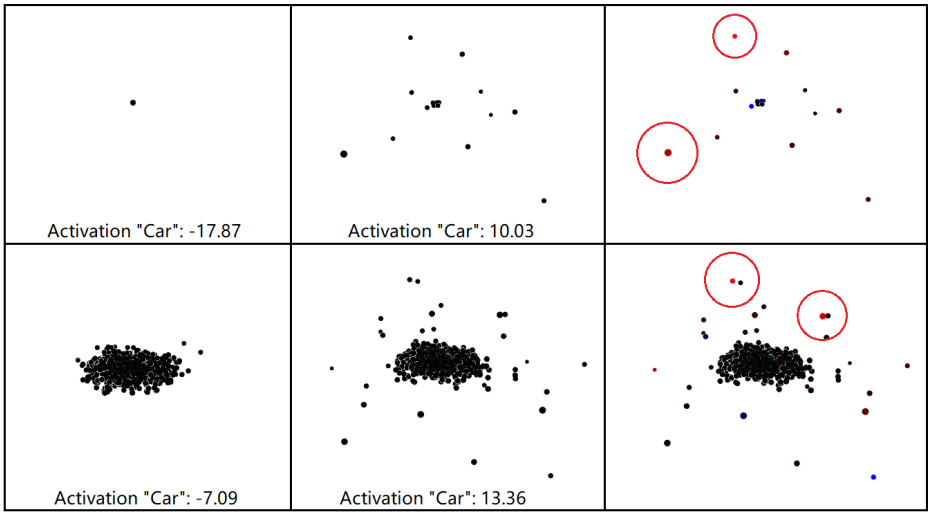

Figure S10. AM results initialized with zeros (the first row) and the point cluster generated by averaging all test data (the second row) respectively. The first, second and third columns demonstrate the initialized set of points, the AM output results after 1000 optimization steps and the salience map explanation of the corresponding output explained by IG, respectively. In the explanation, red points indicate the degree of positive attributions.

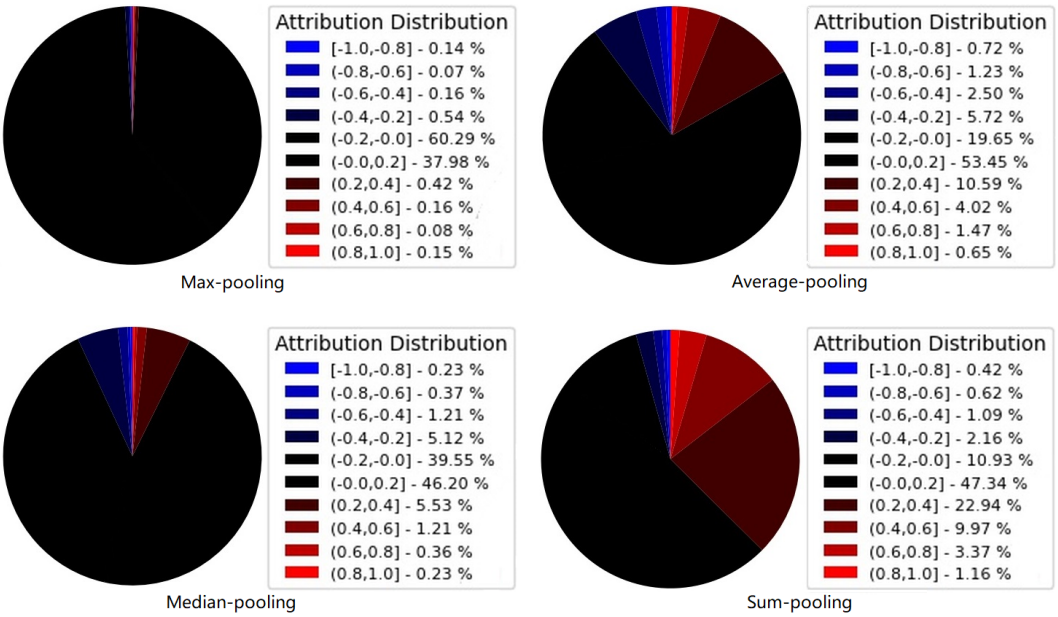

Figure S11. The distributions of attributed points of PointNet structured with max, average, median and sum-pooling layers as the global feature extraction layer respectively.