Adversarial Attack against 3D Shapes Utilizing their Common Points

Junqiao Chen
School of Software Engineering
East China Normal University
Shanghai, China
10211510469@stu.ecnu.edu.cn

Abstract—**3D shape, whose representation contains 3D point cloud and 3D mesh, has played an important role in many security-sensitive domains with the application of 3D Deep Neural Network (DNN). However, adversarial attack aims to threaten DNNs' security by misleading them into wrong prediction. Existing adversarial attacks are primarily designed for point cloud with few studies on 3D mesh adversarial attack. Such disproportion leads to the limitation of the practicality of 3D adversarial attack. Therefore, we propose a 3D adversarial attack, named AdvSCP, which perturbs on the common points to support both 3D point cloud and 3D mesh. In detail, we first design the generation of adversarial perturbation according to the direction of back-propagated gradient. Then we apply the perturbation on the overlapping coordinate field for two reasons. To enhance the effectiveness and stealthiness of our method, iterative generation on adversarial result and append 3D distance measurement as constraint in loss function are considered. Experiments on AdvSCP achieve strong attack efficiency while maintaining imperceptible stealthiness.**

Keywords-Adversarial Attack; 3D Point Cloud; 3D Mesh; Stealthiness; Gradient

I. Introduction

With the rapid development of 3D sensing technologies, applications of 3D data have been rising in many important fields, such as autonomous driving, virtual reality and medical imaging. 3D Point cloud and 3D mesh have been the primary representations of 3D shapes and are used widely for their ability to capture the geometric and spatial information of physical objects. Thus, the security of 3D shape has become a critical concern.

For more reliable and extensive usage in these scenarios, 3D Deep Neural Networks (3D DNN) have been developed to explore 3D point clouds and meshes[1-4]. However, DNN are vulnerable to purposefully designed attacks, which can be roughly divide into two threats including backdoor attack and adversarial attack. Backdoor attack is a kind of attack where attackers implant a malicious trigger into DNN. Nevertheless, backdoor attack requires to contaminate target DNN's dataset, thereby limiting its applicability.

Adversarial attack is another attack that threatens the security of DNN in the inference stage with looser restrictions in its usage. It perturbs the input data, aiming to generate adversarial samples that are imperceptible in human vision domain but can mislead the target model to produce incorrect outputs. More detailed, the target of the attack can be either targeted, which necessitates specially assigned outputs, or untargeted, where the objective is simply to produce incorrect predictions. Such attacks offer advantages like time efficiency and stealthiness. In consequence, substantial efforts are devoted to studies in 2D image adversarial attack[5-8].

In comparison with 2D images, there are relatively fewer researches concentrating on adversarial attack for 3D shapes because of the different data format. Xiang et al.[9] propose an attack for 3D point cloud by perturbing points. Zhang et al.[10] follow the idea in 2D adversarial images to craft adversarial point clouds with the guidance of gradients[5]. However, such efforts are motivated to attack point clouds, ignoring the application for meshes. Therefore, this paper designs an **a**dversarial **a**ttack against 3D **s**hapes like point cloud and mesh, utilizing their **c**ommon **p**oints to craft perturbation (AdvSCP).

We firstly analyze how to generate perturbation in the adversarial attack. In detail, 3D shapes are perturbed based on the direction of the back-propagated gradient with respect to the input of target 3D DNN. One of the challenges in the attacking process is the incompatibility between point cloud and mesh, from both of which requires attackers to extract common attributes. Thus, the common points, which stand for vertices in mesh, are obvious to be the perturbation target. 3D DNNs for meshes usually dissect the mesh information and construct exclusive features as their inputs. In order to perturb vertices in mesh, the feature construct function g is suggested to transform the gradients of features into the ones on vertices. Furthermore, the imperceptibility of previous 3D point cloud adversarial attacks remains inadequate. Therefore, we aim to deploy 3D distance measurements like L_2 distance as the constraint into the loss function to maintain visual and geometric consistency with the original data.

We conduct experiments to validate the effectiveness of our method. It suggests that AdvSCP achieves an average success attack rate of 96% in several DNNs[1, 11-14] designed for both point clouds and meshes. Meanwhile, the addition of constraint reveals a positive trade-off between attack effectiveness and stealthiness. Our main contributions can be summarized into two parts: (1) An adversarial attack method against 3D point cloud and 3D mesh is proposed. (2) Experiments on several DNNs suggest that AdvSCP achieves well performance in attack efficiency with subtle imperceptibility. Meanwhile, we validate the effectiveness of extra constraint.

II. Related Works

A. 3D Shape Deep Neural Network

Classification is a significant field in 3D data processing. Various deep-learning-based approaches have achieved excellent performance on both point clouds and meshes. PointNet[1] extracts point-wise features with multi-layer perception and ensures the order-invariant property of point clouds with symmetric function and max-pooling. PointNet++[14] further builds the set abstraction layers to get multi-scale local information with sampling and grouping points. Point-BERT[15] follows the idea of BERT[16] to pretrain point clouds transformers with point tokens containing significant local information. Then TNPC[17] utilizes transformer-based framework to capture local and global features. MeshNet[13] is a classical DNN designed for mesh data, which makes use of face and neighbor information from meshes and develops three modules, including spatial descriptor, structural descriptor and mesh convolution to construct the network. RIMeshGNN[3] employs graph neural networks with carefully designed aggregation and pooling layers to solve rotation invariance. MEAN[4] proposes an attention-based approach for 3D mesh to take full advantages of local and global structural information.

B. 3D Adversarial Attacks

In contrast to the thorough studies on adversarial learning for 2D images, the development for 3D shapes is in progress, with three common attack categories based on the perturbation ways: addition, delete and transformation. For point clouds, Xiang et al.[9] add synthetic points by mapping from geometrical pattern to the target class, which firstly applies the adversarial attack on 3D point clouds. Zheng et al.[18] propose a point cloud saliency map to figure out the import points in the prediction of the model and introduce an attack method to delete such salient points. Chen et al.[19] introduce Local Aggressive Adversarial Attack conducted by combinations of strategies to improve the balance between attack efficiency and perceptibility. Li et al.[20] utilize optimal transport on the data manifold to generate transferable adversarial 3D point clouds for cross-network attacks. Xu et al.[21] perform attack directly inside mesh data and propose two adversarial sample generation strategies, vertex-based perturbation and edge-based perturbation. Stolik et al.[22] perturb meshes in the spectral domain to deceive the autoencoder into reconstructing different geometric shapes. Cengiz et al.[23] put forward an adversarial attack method via constraining perturbations to the mesh surface of point clouds.

Existing works on 3D DNN and relative adversarial attack concentrate only on either 3D point cloud or 3D mesh, ignoring the commonality in the data formats as mentioned above. Moreover, the stealthiness of these attack methods is not yet sufficient in practical context.

III. Methodology

A. Threat Model

We consider the common threat model studied in previous works[5, 24, 25]. It focuses on such a scenario that users own their own 3D DNNs and datasets, though they lack computing resource for training and upload on thirty-part platform which offers related services. The attackers on the thirty-part platform then are able to generate attacking examples and misleads the target DNN into wrong predictions. To sum up, it belongs to a white-box attack model so that model structure and possible label information (especially for target attack) are exposed.

B. Problem Definition

A point cloud is a set of unsorted points $P=\{pi\ /\ pi\in R^{n\times D}\}$, which contains points' Cartesian coordinates (x, y, z) and other D-3 affiliated features like color, normal and intensity. A mesh is another kind of expression on 3D data with topological information. It is usually represented by a tuple $M=(V, E, F)$, where $V=\{vi\ /\ vi\in R^3\}$, $E\subset V\times V$ and $F\subset V^3$ stand for vertices, edges and faces respectively. A 3D classification DNN can be described as $f(X, \theta)=c$ aiming to predict the label c. Its input X represents either point clouds P or meshes M, and θ represents parameters of DNN. In the adversarial attacks, we tend to craft perturbation δ, which generate an adversarial example $X^{adv}=X+\delta$. To mislead the target DNN, the objective function is like the following expression:

$$argmax_{\delta}J(f(x+\delta), y)\ \ s.t.||\delta||_p\leq\varepsilon, \tag{1}$$

where ε restrains the perturbation level and $J(\cdot)$ represents the loss function of the attack and decides the algorithm of the attack. Here we propose an efficient untargeted method to ensure the efficiency and stealthiness of adversarial results.

C. AdvSCP

The framework of AdvSCP is shown in Fig. 1. We divided the core into four parts.

Adversarial Perturbation Generation. We design an adversarial attack method tailored for 3D shapes like point clouds and meshes. It crafts perturbation based on gradient with respect to input X during training. Inspired by FGSM[5], a gradient-based adversarial example generation algorithms in 2D image classification task, we determine the direction of perturbation by gradient and introduce α to represent the step on perturbed direction. (2) describes how to generate δ as follows:

$$\delta=\alpha sign(\nabla_X J(\theta; X, c)), \tag{2}$$

where J means the loss function of DNN.

Adaption on Mesh. The above perturbation generation has been studied in point clouds[10]. In a DNN to classify point clouds, the regular input is always coordinate of points, which means δ is acquired by $\nabla_P J(\theta; P, c)$. However, classification on meshes takes feature as its input, which is extracted by $g(M)$,

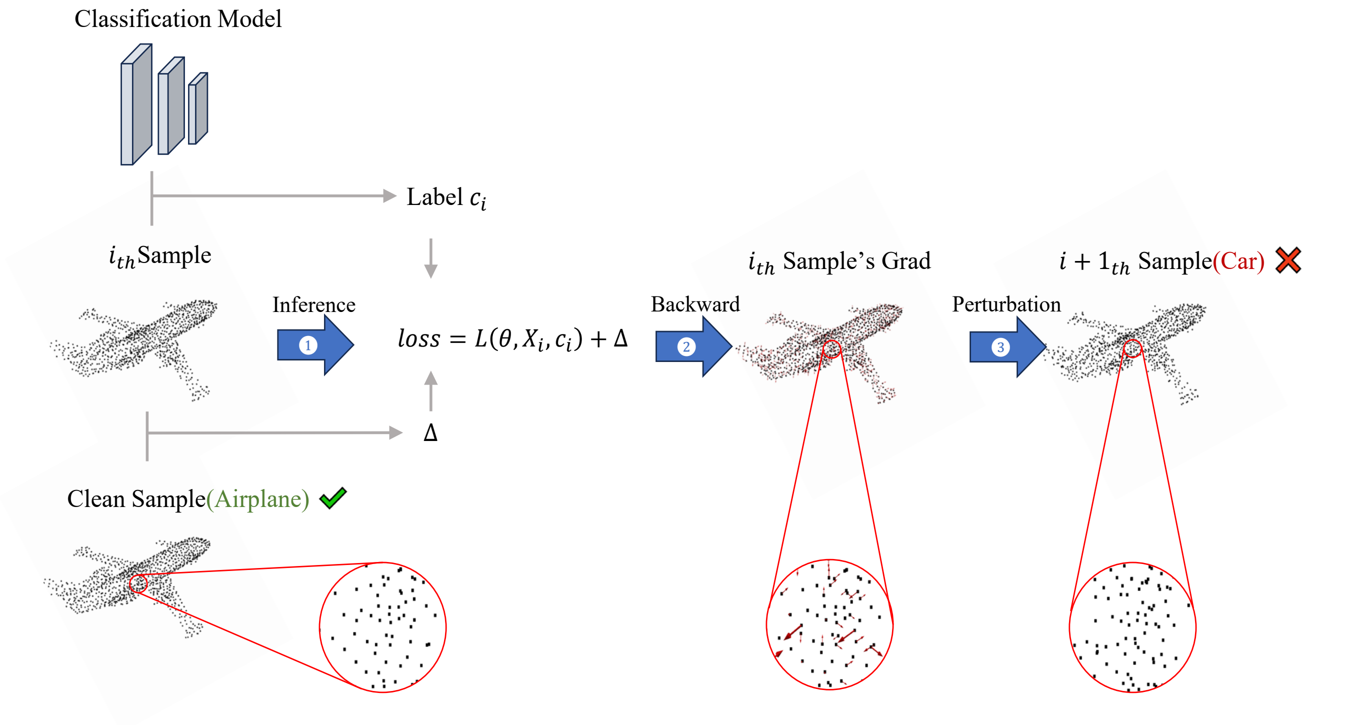


Figure 1. Framework for the i-th iteration on adversarial samples generation. The main three steps are as follows: 1) Infer on classification model and compute distance from clean sample to acquire loss. 2) Calculate the gradient of loss with respect to the i-th sample X_i. 3) Perturb the adversarial sample based on the sign of gradient. After that victim 3D DNN will classify the sample into car rather than its correct label airplane. The workflow shows an adversarial attack on point cloud, however AdvSCP is also applicable to meshes.

where g is the feature construction function on mesh information for the specific DNN. The feature is not compatible with either point cloud's coordinates or other features built by different 3D DNNs. As a result, this paper aims to develop a method to create perturbation adaptable to both point clouds and meshes. We choose to perturb on vertices of meshes for two reasons: (1) Perturbation on vertices is naturally consistent with the operation on point clouds, both of which contains coordinates information. Besides, perturbation on coordinates intrinsically displays an explicit geometric interpretation. (2) Diverse methods introduce their exclusive features to conduct the classification. However, it is difficult to represent such features in a unified manner. Thus, we transform the focusing domain into the original mesh representation M and leverage information on V for integration.

Moreover, two aspects of optimization are factored in consideration to improve the efficiency and stealthiness of our method. The first optimization is to utilize iteration to generate adversarial examples, while the second one is to introduce constraint on loss function.

Iterative Generation on Perturbation. We propose a method to iteratively construct adversarial examples through multiple steps, rather than completing the process in a single iteration. It allows the adversarial examples to better adapt to complex decision boundaries, reducing the risk of prematurely getting trapped in local minimal. The iteration operation performs as follows:

$$X_{i+1}=X_i+\alpha sign(\nabla_{X_i}J(\theta; X_i, c)), \tag{3}$$

where X_i means the adversarial example after the i-th iteration. The hyperparameter k represents the maximum number of iterations.

Constraint on Loss. In order to restrain the range of perturbation into an acceptable level, we introduce regularization constraint on the loss function J. In detail, we deploy 3D distance metrics like L_2 distance between the adversarial examples and the original ones as constraint and define its weight parameter λ shown in (4) as:

$$L(\theta, X_i, X_0, c) = J(\theta, X_i, c)+\lambda||X_i—X_0||_2. \tag{4}$$

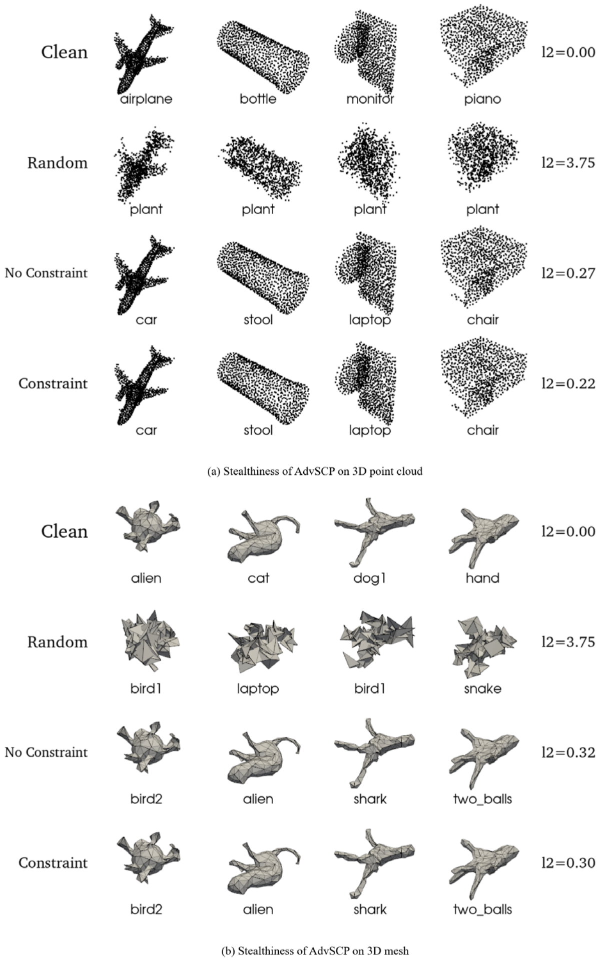
By applying the constraint, we tend to enhance the stealthiness of adversarial results with more subtle discrepancies to human vision.

D. Model Attacking

The hyperparameter iteration may be set to be overly large. Therefore, we can alternatively exit early once an example has successfully misled DNN. In view of network's performance, those samples that are wrongly categorized firstly in attacking process are ignored. The adversarial attack will generate adversarial examples corresponding to the victim DNN after attacking process. After that, attackers can easily mislead DNN with generated adversarial examples. In detail, the victim DNN will misclassify the adversarial input into wrong class.

IV. Experiments

In this session, a series of experiments are carried out to verify our method. We firstly elaborate our experiments' setup and then compare our AdvSCP with random perturbation to validate the feasibility. After that, the performance of our method in terms of an untargeted attacking framework is estimated. Furthermore, in order to demonstrate the indispensability of our method, the ablation study on the corresponding hyperparameters is conducted. The visualization result of AdvSCP is illustrated in Fig 2.



(a) Stealthiness of AdvSCP on 3D point cloud

(b) Stealthiness of AdvSCP on 3D mesh

Figure 2. Visualization of our AdvSCP method, which presents results of point cloud and mesh in (a) and (b). It shows that AdvSCP can mislead the target 3D DNN with barely perceptible change for human eyes. Moreover, the constraint enhances adversarial samples' indiscernibility in terms of L_2 distance.

A. *Experiments Setting*

Our experiments are designed with datasets that correspond to the different data format. Attacks are carried out on ModelNet40/10[26], ShapeNet[27] for point cloud, while on Shrec16, Cubes and Manifold40 for mesh. To ensure fair comparison, the dataset used in a certain model is in accordance with model's origin experiment design.

ModelNet40/10 and ShapeNet are both used for point cloud model PointNet[1], PointNet++[14] and DGCNN[11] in evaluating attack experiments. ModelNet40 consists of 12,311 CAD models classified into 40 classes, 9,843 of which are for training and others are for testing. ModelNet10 is a simplified version of ModelNet dataset in 10 classes with 3,991 training models and 908 testing models, bringing the overall count to 4,899 models. ShapeNet is also a large-scale dataset of 3D point cloud. We choose a subset of ShapeNetCore for part segmentation, which includes 16 classes with 12,137 training samples and 2,874 testing samples. For 3D mesh, MeshCNN[12] is attacked on Shrec16, and MeshNet[13] on Manifold40. In addition, these two DNNs are further attacked on Cubes. Shrec16 comprises 600 meshes from16 classes, and the training data and test data are 480 and 120, respectively. Manifold40 repairs the models in ModelNet40 to make them watertight manifolds and possesses the same scale. Cubes is a triangular mesh dataset for classification, which contains 3,722 training meshes and 659 test meshes in 22 classes.

At the beginning of the experiments, we firstly train all DNNs on corresponding datasets, where the sampling size is standardized to 1,024. L_2 distance is utilized to quantify the imperceptibility of the crafted adversarial data. Then we evaluate the performance of the attacking methods by **a**ttack **s**uccess **r**ate (ASR), which is the fraction of the adversarial examples that are misclassified by target models to all attacked examples. The hyperparameters of different models on various datasets are dynamically adjusted based on experimental results to maximize ASR and minimize L_2 distance.

B. *Effectiveness of AdvSCP*

We firstly demonstrate the feasibility of AdvSCP by comparing with random perturbations. The attack success rates are strived to maintain alignment as shown in Table 1. The result indicates that in order to achieve an effective attack, random perturbation obviously changes the clean samples considerably on both point clouds and meshes while AdvSCP keeps the variation slight. Shown from Fig. 2, random perturbation almost ruins the samples into unrecognizable objects. It further validates the feasibility of gradient-based perturbation on 3D shapes' common points.

Table 1. The comparison of AdvSCP with random perturbation.

	AdvSCP		Random	
Models	ASR	L_2 Distance	ASR	L_2 Distance
PointNet++	0.98	0.27	0.91	3.75
MeshCNN	0.98	0.31	0.95	3.75

Further experiments are performed to evaluate the performance of our AdvSCP. Shown in Table 2, our method achieves high ASR against all attacked models on various datasets. In particular, attacking for MeshCNN on Shrec16 dataset reaches an ASR of 99% with only 0.306 L_2 distance. As shown in Fig. 2, AdvSCP generates adversarial samples which own hardly noticeable differences with the clean data in human vision. Though the over-performance is great, the L_2 distances of different models after attacking reveal gaps, which implies the challenge to deceive a certain model. We notice that the complexity of DNN is positively correlated with the difficulty of the attack, as DGCNN is the most complicated for its dynamics.

C. Stealthiness of AdvSCP

Evaluation on the effect of constraint is conducted by comparing AdvSCP with constraint in loss and the one without constraint. Results in Table 2 illustrate that the constraint decreases the L_2 distance of adversarial examples with little reduction on ASR in most cases (12/13). Especially, L_2 distance of PointNet++ on ModelNet40 decreases from 0.569 to 0.496 when its ASR slightly drops 0.03. Furthermore, ASR and L_2 distance of MeshCNN on Cubes drop from 1.0 and 0.356 to 0.96 and 0.313, respectively. With further evaluation based on Fig. 2, result suggests that our method achieves notable stealthiness in both qualitative and quantitative aspects. Moreover, the introduction of constraint further reduces the discrepancy with the clean data.

Table 2. Performance on several 3D DNNs and datasets with comparison between original AdvSCP and the ablation one without constraint.

Dataset	Models	With constraint		No constraint	
		ASR	L_2 Distance	ASR	L_2 Distance
ModelNet40	PointNet	**0.97**	**0.45**	1.00	0.57
	PointNet++	0.95	0.92	1.00	0.97
	DGCNN	0.98	1.83	0.99	1.90
ModelNet10	PointNet	0.96	0.80	0.98	1.01
	PointNet++	0.99	1.02	1.00	1.07
	DGCNN	0.97	2.51	0.98	2.74
ShapeNet	PointNet	0.96	**0.66**	0.99	0.75
	PointNet++	**0.97**	1.21	0.97	1.30
	DGCNN	0.95	3.56	0.98	4.00
Shrec16	MeshCNN	**0.99**	**0.31**	1.00	0.31
	MeshNet	0.99	0.31	1.00	0.32
Cubes	MeshCNN	**0.96**	**0.31**	1.00	0.36
Manifold40	MeshNet	0.84	0.44	0.99	0.49

D. Ablation Study

We perform ablation studies to assess the influence of two hyperparameters: k and α. The ablation experiments are conducted on ModelNet10 and Shrec16 for point clouds and meshes, respectively, with results illustrated across Figs. 3 to 6. Though the experiments are conducted separately on the two data formats, the consequences show consistent. However, the impact of the two hyperparameters on attacking performance varies in degree. ASR on k exhibits a more stable growth trend with a smoother increasement of L_2 distance. Moreover, α has a stronger influence on both ASR and L_2 distance. The shift of α on PointNet++ even causes an apparent drop on ASR according to Fig. 5. It suggests that the augmentation of k rather than α is better for it is more controllable and generates more imperceptible adversarial results.

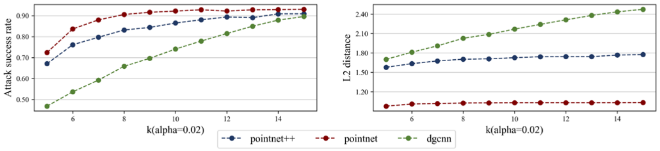


Figure 3. The influence of k on attack success rate and L_2 distance for point cloud.

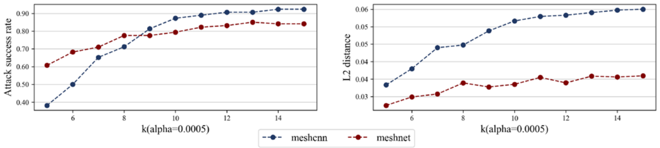


Figure 4. The influence of k on attack success rate and L_2 distance for mesh.

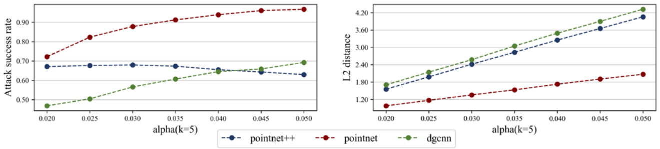


Figure 5. The influence of α on attack success rate and L_2 distance for point cloud.

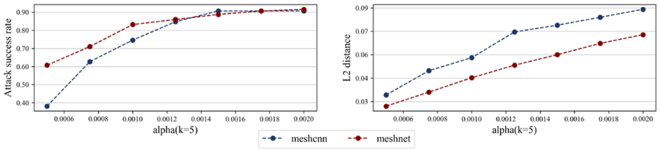


Figure 6. The influence of α on attack success rate and L_2 distance for mesh.

V. Conclusions

In this paper, we propose an adversarial attack based on the back-propagated gradient, named AdvSCP. In detail, it crafts perturbation on coordinates, aiming to attack both 3D point cloud and 3D mesh. Moreover, iterative generation of adversarial samples and constraint on loss function are performed to improve the performance. Experiments suggest that AdvSCP achieves strong attack efficiency and imperceptible stealthiness.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652-660, 2017.

[2] H. Liu and S. Tian, "Deep 3D point cloud classification and segmentation network based on GateNet," The Visual Computer, vol. 40, no. 2, pp. 971-981, 2024.

[3] B. Shakibajahromi, E. Kim, and D. E. Breen, "Rimeshgnn: A rotation-invariant graph neural network for mesh classification," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3150-3160, 2024.

[4] J. Dai, R. Fan, Y. Song, Q. Guo, and F. He, "MEAN: An attention-based approach for 3D mesh shape classification," The Visual Computer, vol. 40, no. 4, pp. 2987-3000, 2024.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[6] Y. Dong et al., "Boosting adversarial attacks with momentum," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9185-9193, 2018.

[7] J.-J. Huang et al., "DeMPAA: Deployable multi-mini-patch adversarial attack for remote sensing image classification," IEEE Transactions on Geoscience and Remote Sensing, 2024.

[8] Y. Song et al., "PB-UAP: Hybride Universal Adversarial Attack for Image Segmentation," ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5, 2025.

[9] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9136-9144, 2019.

[10] Y. Zhang, G. Liang, T. Salem, and N. Jacobs, "Defense-pointnet: Protecting pointnet against adversarial attacks," 2019 IEEE International Conference on Big Data (Big Data), pp. 5654-5660, 2019.

[11] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," ACM Transactions on Graphics (tog), vol. 38, no. 5, pp. 1-12, 2019.

[12] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," ACM Transactions on Graphics (ToG), vol. 38, no. 4, pp. 1-12, 2019.

[13] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, pp. 8279-8286, 2019.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in neural information processing systems, vol. 30, 2017.

[15] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 19313-19322, 2022.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp. 4171-4186, 2019.

[17] W. Zhou, Y. Zhao, Y. Xiao, X. Min, and J. Yi, "TNPC: Transformer-based network for point cloud classification," Expert Systems with Applications, vol. 239, p. 122438, 2024.

[18] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," Proceedings of the IEEE/CVF international conference on computer vision, pp. 1598-1606, 2019.

[19] Z. Chen, F. Chen, Y. Sun, M. Wang, S. Liu, and Y. Ji, "Local aggressive and physically realizable adversarial attacks on 3D point cloud," Computers & Security, vol. 139, p. 103539, 2024.

[20] Z. Li, X. Du, N. Lei, L. Chen, and W. Wang, "NoPain: No-box Point Cloud Attack via Optimal Transport Singular Boundary," arXiv preprint arXiv:2503.00063, 2025.

[21] H. Xu, F. He, L. Fan, and J. Bai, "D3AdvM: A direct 3D adversarial sample attack inside mesh data," Computer Aided Geometric Design, vol. 97, p. 102122, 2022.

[22] T. Stolik, I. Lang, and S. Avidan, "SAGA: Spectral adversarial geometric attack on 3D meshes," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4284-4294, 2023.

[23] B. Cengiz, M. Gülşen, Y. H. Sahin, and G. Unal, "ε-Mesh Attack: A Surface-based Adversarial Point Cloud Attack for Facial Expression Recognition," 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1-9, 2024.

[24] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity," ACM Comput. Surv., vol. 55, no. 8, p. Article 163, 2022, doi: 10.1145/3547330.

[25] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li, "Adversarial attack and defense: A survey," Electronics, vol. 11, no. 8, p. 1283, 2022.

[26] Z. Wu et al., "3d shapenets: A deep representation for volumetric shapes," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1912-1920, 2015.

[27] A. X. Chang et al., "Shapenet: An information-rich 3d model repository," arXiv preprint arXiv:1512.03012, 2015.