Adversarial Attacks on 3D Geometries: A Spectral Perspective

Tal Peer, Idan Cohen, Tomer Zvi

May 2025

Abstract

Although current 3D attack methods achieve high success rates, they predominantly operate in the data space via point-wise manipulations [12, 9, 13]], potentially overlooking critical geometric structures. In this project, we examine an alternative approach—performing adversarial attacks in the *graph spectral domain* on the mesh's vertices (i.e., the underlying point cloud). This approach perturbs the spectral coefficients of a graph transform, targeting frequency components associated with meaningful geometric deformations [6, 7, 4]. Experimental results on ModelNet10 reveal that PointNet++ suffers a drastic accuracy drop from 80% to 16% under our attack, while MeshNet retains 83% accuracy. These findings underscore the need for further research into the structural robustness of mesh-based classifiers and the efficacy of frequency-aware adversarial perturbations. GitHub repo: 3DAttackOnFrequency

Model	**Setting**	**Accuracy**	**Macro F1**	**Weighted F1**
PointNet++	Clean	80.0%	80.0%	80.0%
PointNet++	Adversarial	16.0%	17.0%	15.0%
MeshNet	Clean	90.0%	89.0%	89.0%
MeshNet	Adversarial	83.0%	79.0%	82.0%

Table 1: Classification results on ModelNet10 Dataset.

1 Introduction

In this project, we propose to attack 3D models from the graph spectral domain perspective. By attacking the mesh's vertices (the underlying point cloud), we aim to perturb graph transform coefficients in the spectral domain that correspond to varying certain geometric structures [6] [7]. Specifically, leveraging on graph signal processing, we first adaptively transform the coordinates of points onto the spectral domain via graph Fourier transform (GFT) for compact representation. Then, we analyze the influence of different spectral bands on the geometric structure, based on which we propose to perturb the GFT coefficients via a learnable graph spectral filter. Considering the low-frequency components mainly contribute to the rough shape of the 3D object [8] [7], we further apply a low-frequency constraint to limit perturbations within imperceptible high-frequency components. Finally, the adversarial point cloud is generated by transforming the perturbed spectral representation back to the data domain via the inverse GFT and processing the attack-point cloud back to its mesh representation. Note that this method keeps the faces indices, while the influenced characteristics from the attack are the mesh vertices (points) and faces areas. The attack is performed over the normalized point cloud, and points are not sampled (mostly due to problems occur in the face matching phase).

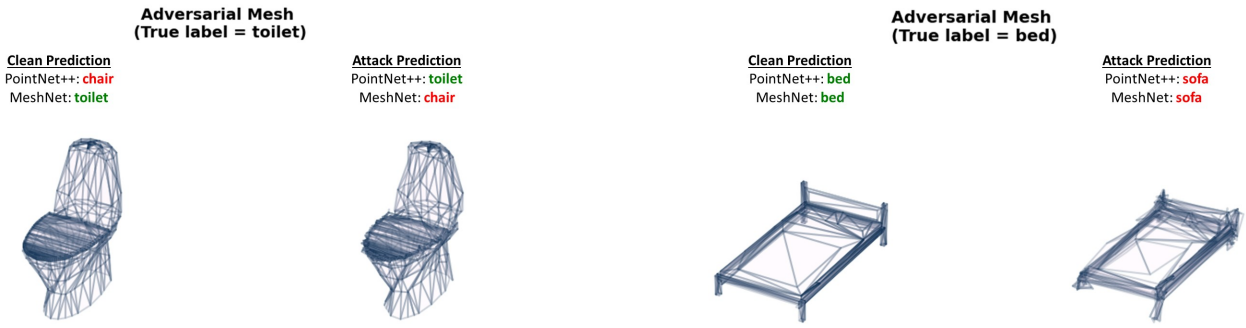
2 Method

Notations and Problem Definition

Let a 3D shape be represented by a triangle mesh $X = (V, F)$, where $V \in \mathbb{R}^{n\times 3}$ is the set of vertices and F is the set of triangular faces. Each mesh is associated with a ground-truth label $y_{\text{true}} \in \mathcal{Y} = \{1, 2, \ldots, 10\}$.

To construct the input for a point cloud classifier, we uniformly sample a fixed number of points from the surface of the mesh, resulting in a point cloud $P = \{p_i\}_{i=1}^{n} \subseteq V$, where $P \in \mathbb{R}^{n\times 3}$. A point cloud classification model $f(\cdot)$ takes P as input and produces a class probability vector $f(P) \in \mathbb{R}^c$, with predicted label:

$$F(P) = \arg\max_{i\in\mathcal{Y}} f(P)_i.$$



To generate adversarial examples, we apply a small perturbation $\Delta \in \mathbb{R}^{n\times 3}$ to the sampled points, yielding $P_{\text{adv}} = P + \Delta$. The attack is considered successful if:

$$F(P_{\text{adv}}) \neq y_{\text{true}}.$$

To assess the transferability of the attack to mesh-based models, we reconstruct a perturbed mesh $X_{\text{adv}} = (V_{\text{adv}}, F)$ by replacing the original vertex set V with the adversarial point cloud P_{adv}, while keeping the face structure F unchanged. This adversarial mesh is then passed to a mesh classifier $H(\cdot)$ (e.g., MeshNet), which outputs a predicted label:

$$H(X_{\text{adv}}) \neq y_{\text{true}}.$$

Attack on Frequency (AOF)

Our implementation is based on the **AOF (Attack on Frequency)** method introduced by Liu et al. [8], which focuses on perturbing the low-frequency components of the point cloud under a graph spectral basis. We create a spectral-domain attack that learns imperceptible perturbations by operating on the graph Fourier transform (GFT) coefficients of the point cloud's KNN graph. The objective is to optimize the adversarial loss while constraining changes in the spectral domain. The problem is formulated as:

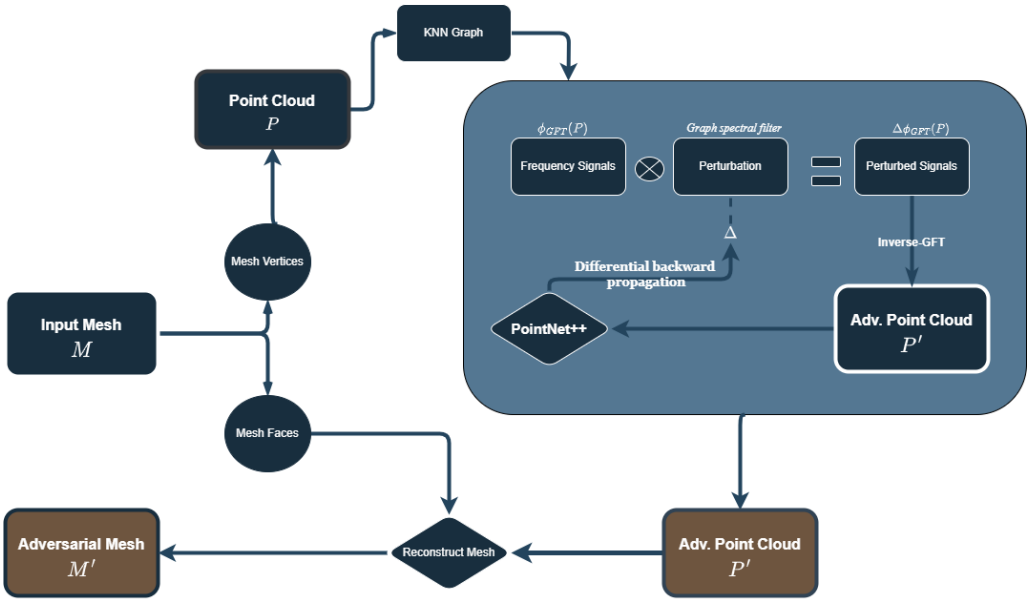
$$\min_{\Delta} \mathcal{L}_{\text{adv}}(P', P, y) \quad \text{s.t.} \quad \|\phi_{\text{GFT}}(P') - \phi_{\text{GFT}}(P)\|_p < \epsilon,$$

where $P' = \phi_{\text{IGFT},\Delta}(\phi_{\text{GFT}}(P))$, and

$$\Delta = \begin{bmatrix} \Delta_{w,1} \cdot \sum_{l=0}^{L-1} \Delta_{h,l} \lambda_1^l \\ \vdots \\ \Delta_{w,n} \cdot \sum_{l=0}^{L-1} \Delta_{h,l} \lambda_n^l \end{bmatrix}.$$

Here, $\mathcal{L}_{\text{adv}}$ denotes the adversarial loss, ϕ_{GFT} and ϕ_{IGFT} are the graph Fourier transform and its inverse, Δ is the learnable spectral perturbation, and ϵ bounds the spectral distortion. The distance is measured in the ℓ_p norm, which in our implementation is instantiated as ℓ_∞.

AOF assumes that the **intrinsic structure** of a 3D object is encoded in the **low-frequency components (LFC)** of its geometric representation, while high-frequency components (HFC) are often model-specific and prone to overfitting. Therefore, rather than perturbing all points indiscriminately, AOF isolates and perturbs only the LFC of the input signal, thereby preserving the overall shape while misleading classifiers in a more transferable manner. In the original paper [8], the motivation for setting the low pass parameter was based on the observation that a significant portion of the energy (75–90%) in GFT coefficients is concentrated in the lowest frequencies (e.g. first 100). Originally, the paper samples 1024 points from each point cloud. In our case, it is required to re-construct the mesh faces, so sampling were too complicated to integrate. We bypassed this obstacle by filtering the dataset to consider meshes with at most 2048 vertices (points), which allowed us to reconstruct the mesh in feasible running times. The attack avoids producing outliers and instead yields subtle shape deformations that preserve the geometric integrity of the object, which are harder to defend against using traditional 3D point cloud filters or denoisers. We found this attribute important especially for the propose of transfer attack to its Mesh form, as the resulted adversarial meshes preserving the original shape characteristics.



Graph Construction and Spectral Decomposition

Given a point cloud $X \in \mathbb{R}^{N\times 3}$, we construct a k-nearest neighbor graph and define an affinity matrix A via:

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\epsilon^2}\right), & \text{if } i \text{ is neighbor of } j \\ 0, & \text{otherwise} \end{cases}$$

The unnormalized Laplacian is computed as $L = D - A$, where D is the diagonal degree matrix. We perform eigendecomposition $L = V\Lambda V^\top$ and use the eigenvectors V as a graph Fourier basis.

The initial adversarial perturbation Δ is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, with shape $\Delta \in \mathbb{R}^{N\times 3}$.

The Projection of the input coordinates into the spectral domain using:

$$X = X_{\text{LFC}} + X_{\text{HFC}} = V_m V_m^\top X + V_{m:N} V_{m:N}^\top X$$

where V_m denotes the first m eigenvectors.
Only the LFC part is optimized, while the HFC is kept fixed and used for reconstructing the point cloud. As [8] mentioned, 75% of the energy is concentrated within the lower 100 eigenvalues for 1024 points sampled, and 90% of the energy is within the lower 400 eigenvalues. We experimented both constant and propose a new way that sets the cutoff frequency to 12.5% of the point cloud size. This approach led to better attack results.

The adversarial loss function was set as the convex combination of the logits-based loss on both full and low-frequency adversarial point clouds:

$$\mathcal{L}_{\text{AOF}} = (1-\gamma)\cdot \mathcal{L}_{\text{mis}}(X') + \gamma \cdot \mathcal{L}_{\text{mis}}(X'_{\text{LFC}})$$

Here, $\mathcal{L}_{\text{mis}}$ is a Carlini-Wagner-style margin loss encouraging misclassification, and $\gamma \in [0, 1]$ balances the contribution from LFC.

Carlini & Wagner Optimization Scheme

The Carlini & Wagner (C&W) attack is a seminal method that formulates adversarial example generation as a constrained optimization problem, aiming to identify the smallest possible perturbation that causes the model to misclassify. The core objective is to balance the imperceptibility of the perturbation, and a successful misclassification.

Let x be the clean input and x' the perturbed input. The attack minimizes the following composite objective:

$$\mathcal{L}_{\text{mis}}(x') = (1-\gamma)\cdot \text{dist}(x, x') + \gamma \cdot \text{loss}(f(x'), y_t),$$

where:

- $\text{dist}(x, x')$ is the perturbation magnitude (e.g., ℓ_p for some p-norm)
- $\text{loss}(f(x'), y_t)$ is a classification loss (e.g., cross-entropy) between the prediction and a target class y_t
- $\gamma \in \mathbb{R}_+$ regularizes perturbation size against attack success

$\boldsymbol{\Delta_{\text{LFC}}}$ is iteratively optimized using Adam optimizer over a fixed number of steps. After each update, the resulting perturbed point cloud $X' = X_{\text{LFC}} + X_{\text{HFC}}$ is projected back and clipped using an ℓ_∞ constraint to ensure perceptual imperceptibility:

$$X' \leftarrow \text{Clip}_{\ell_\infty}(X', X, \epsilon)$$

3 Experiments

Dataset We evaluate on ModelNet10 dataset, contains CAD models from the 10 categories. Adversarial point clouds are generated using our attack method, using PointNet++. Transferability to MeshNet suffers from heavy computational and transforming issues, especially in the process of aligning the reconstruct faces from the sampled vertices. To overcome this obstacle, we decided to filter the dataset samples that consist of more than 2048 points (201 samples were considered for evaluation).

Name	**# Graphs**	**# Nodes**	**# Edges**	**# Features**	**# Classes**
ModelNet10	4,899	$\sim$ 9,508.2	$\sim$ 37,450.5	3	10

Table 2: Original ModelNet10 dataset.

Point Cloud Experiments

To assess the sensitivity and effectiveness of our spectral attack on point clouds, we conducted ablation experiments varying key hyperparameters. These included the number of optimization steps and iterations, learning rates, and the choice of spectral cutoff, both in fixed and adaptive forms. We also evaluated the impact of different input resolutions by testing both sampled and full-resolution point clouds.

- K (KNN-graph): **30**
- Optimization steps: 5, **10**, 20
- Inner iterations per step: 100, 150, **175**, 200, 250, 375, 450
- Vertex counts: 2048 (sampled), **2048 (full)**
- Learning rate: 1×10^{-3} , 5×10^{-3}
- Spectral cutoff (fixed): 50, **100**, 200
- p-Norm clipping: ℓ_2, ℓ_∞

Both ℓ_2 and ℓ_∞ norms were experimented for clipping. ℓ_∞ norm as it outer-performed over the ℓ_2, which led us to choose this at early stages of the experiments.

Mesh Adjustment Experiments

To evaluate how different mesh configurations affect the transferability and robustness of the attack, we conducted a complementary set of experiments on reconstructed adversarial meshes. These experiments varied spectral cutoff definitions (constant and relative) and tested sensitivity to mesh resolution and optimization schedule.

- Optimization steps: 10, 20
- Inner iterations per step: 175
- Vertex count: 2048 (filtered only, to preserve mesh structure)
- Spectral cutoff (constant): 50, 100
- Spectral cutoff (proportional): 0.1, 0.125 × number of vertices

Attack Settings

Target Models

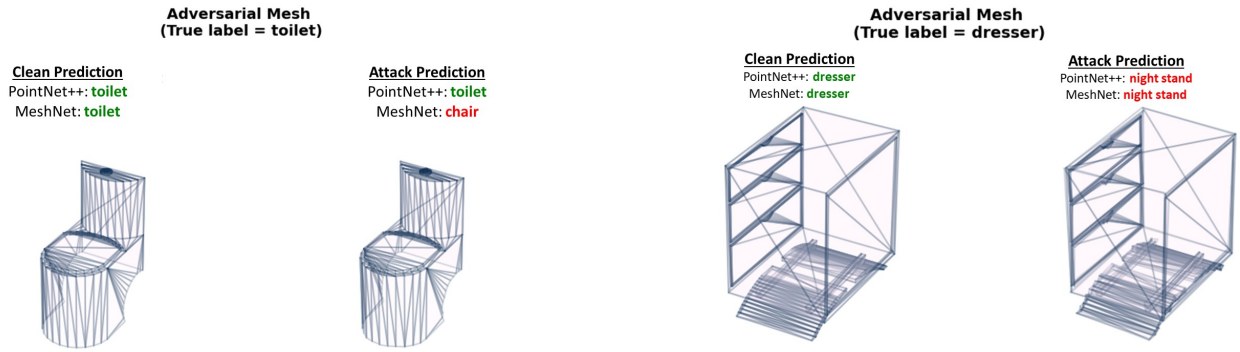
We test our adversarial examples on two representative 3D classifiers: MeshNet [2] and PointNet++ [11]. Both are evaluated using their default architectural configurations. For MeshNet, we follow the original training setup described in the official implementation. For PointNet++, we use the hyperparameter sweep configuration labeled ***bumbling-sweep-11*** from Weights & Biases platform.

Adversarial Point Cloud Generation

We begin with the generation of adversarial point clouds using our spectral-domain attack framework, as described in the previous section. Given original 3D objects from ModelNet10, we apply our AOF-inspired optimization procedure directly in the Laplacian spectral domain. The attack is carried out over the unaltered vertex set of the mesh (i.e., no sampling or resampling of points is performed), and the low-frequency components of the Laplacian eigen-basis are perturbed using a constrained iterative procedure. The optimization is conducted over 10 outer steps and 175 inner iterations using the Adam optimizer with a fixed learning rate of 1×10^{-3}. Perturbations are clipped using an ℓ_∞ constraint with a fixed budget to ensure perceptual similarity between clean and adversarial samples. We use CrossEntropy loss and restrict the perturbation magnitude to a unit-ball of size $\epsilon = 0.1$. The attack runs for 10 epochs, where a successful attack is recorded when the classifier's predicted label diverges from the true label and the distance is smaller from the previous recorded distance.

Point-to-Mesh Transferability

Both clean and adversarial mesh datasets are evaluated using a pretrained MeshNet classifier. The model receives as input the structured face features and topological neighborhood indices and outputs a predicted label per shape. Evaluation is conducted over a filtered subset of 201 test examples that includes only the adversarial meshes with corresponding clean counterparts. We compare model predictions across clean and adversarial versions of each input, measuring top-1 accuracy, macro-averaged F1 score, and class-wise degradation. Since the spectral attack is crafted solely with respect to PointNet++, this evaluation assesses the extent to which adversarial geometry can transfer to an unseen, structurally different model.



4 Results and Analysis

Quantitative results are summarized in Tables **3** and **4**. The PointNet++ model achieves 80% accuracy on clean point clouds, but drops to 16% under attack.

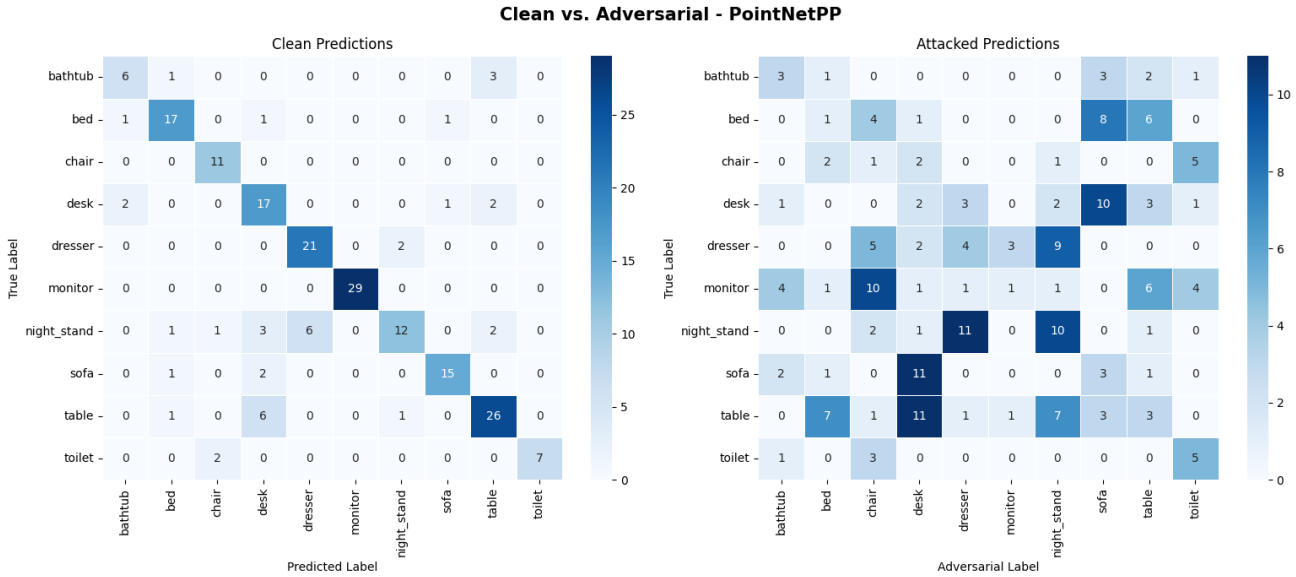
(a) Clean PointNet++

Class	Precision	Recall	F1-score	Support
bathtub	0.67	0.60	0.63	10
bed	0.81	0.85	0.83	20
chair	0.79	1.00	0.88	11
desk	0.59	0.77	0.67	22
dresser	0.78	0.91	0.84	23
monitor	1.00	1.00	1.00	29
night_stand	0.80	0.48	0.60	25
sofa	0.88	0.83	0.86	18
table	0.79	0.76	0.78	34
toilet	1.00	0.78	0.88	9
Accuracy			0.80	201
Macro avg	0.81	0.80	0.80	201
Weighted avg	0.81	0.80	0.80	201

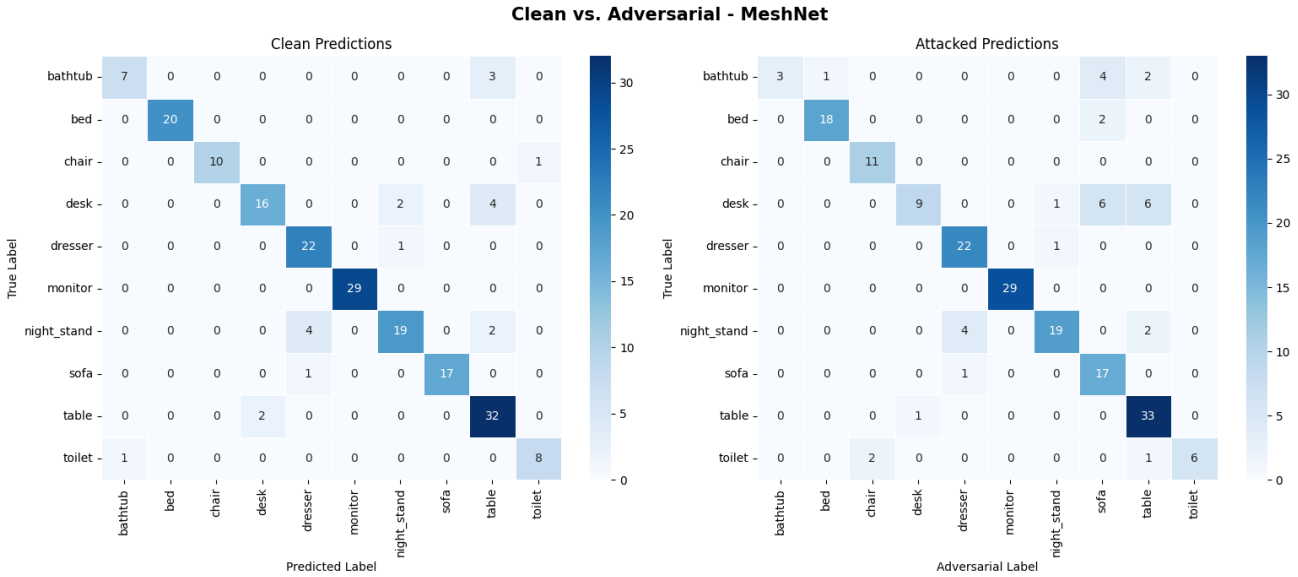
(b) Attacked PointNet++

Class	Precision	Recall	F1-score	Support
bathtub	0.27	0.30	0.29	10
bed	0.08	0.05	0.06	20
chair	0.04	0.09	0.05	11
desk	0.06	0.09	0.08	22
dresser	0.20	0.17	0.19	23
monitor	0.20	0.03	0.06	29
night_stand	0.33	0.40	0.36	25
sofa	0.11	0.17	0.13	18
table	0.14	0.09	0.11	34
toilet	0.31	0.56	0.40	9
Accuracy			0.16	201
Macro avg	0.17	0.20	0.17	201
Weighted avg	0.17	0.16	0.15	201

Table 3: PointNet++ classification metrics on clean and adversarial samples.



In contrast, MeshNet maintains high performance on the clean mesh set (90%) and exhibits only a moderate decline (83%) on the adversarial meshes. Class-wise analysis reveals that spectral perturbations transfer effectively for certain categories (e.g., “chair”, “desk”), while others remain relatively robust (e.g., “monitor”). Notably, MeshNet’s resilience is most apparent in geometrically stable classes such as “bed” and “sofa”, where low-frequency perturbations are less likely to impact high-level structural cues. These results support our hypothesis that mesh-based classifiers provide partial defense against low-frequency adversarial perturbations.



(a) Clean MeshNet

Class	Precision	Recall	F1-score	Support
bathtub	0.88	0.70	0.78	10
bed	1.00	1.00	1.00	20
chair	1.00	0.91	0.95	11
desk	0.89	0.73	0.80	22
dresser	0.81	0.96	0.88	23
monitor	1.00	1.00	1.00	29
night_stand	0.86	0.76	0.81	25
sofa	1.00	0.94	0.97	18
table	0.78	0.94	0.85	34
toilet	0.89	0.89	0.89	9
Accuracy			0.90	201
Macro avg	0.91	0.88	0.89	201
Weighted avg	0.90	0.90	0.89	201

(b) Attacked MeshNet

Class	Precision	Recall	F1-score	Support
bathtub	1.00	0.30	0.46	10
bed	0.95	0.90	0.92	20
chair	0.85	1.00	0.92	11
desk	0.90	0.41	0.56	22
dresser	0.81	0.96	0.88	23
monitor	1.00	1.00	1.00	29
night_stand	0.90	0.76	0.83	25
sofa	0.59	0.94	0.72	18
table	0.75	0.97	0.85	34
toilet	1.00	0.67	0.80	9
Accuracy			0.83	201
Macro avg	0.87	0.79	0.79	201
Weighted avg	0.86	0.83	0.82	201

Table 4: MeshNet classification metrics on clean and adversarial samples.

5 Conclusion

Our results show that **PointNet++ is highly vulnerable** to adversarial perturbations generated using the AOF method. Notably, classes such as *bed*, *chair*, *monitor*, and *desk* suffer dramatic degradation (F1-scores around 0.05–0.08).

Despite being tested under the same attack, **MeshNet exhibits significantly greater robustness**, with only a 7% drop in accuracy. This suggests that MeshNet's reliance on richer local geometric and topological features confers partial resistance to such attacks, even when they are not explicitly designed for mesh classifiers.

These findings highlight shared vulnerabilities across 3D representations, while also revealing architectural differences in adversarial robustness between PointNet++ and MeshNet.

6 Future Work

Our findings suggest that **transferable adversarial attacks on 3D models** represent a promising direction for further exploration. Throughout the project, we sought to leverage the graph structure—and in particular, the community structure—of 3D geometries, hypothesizing that spectral-domain attacks may yield better results than existing spatial-domain methods. Based on our experience and preliminary results, several research avenues merit

further investigation:

- **Hyperparameter tuning**: Exploring alternative configurations and systematic methods for selecting the spectral low-pass cutoff used in the attack.
- **Model evaluation**: Extending the evaluation to a wider range of 3D model architectures, particularly under spectral-domain attacks and possible defenses [8, 7].
- **Sampling strategies**: Studying the impact of various point and face sampling techniques on attack effectiveness and transferability (see Figures 4 and 3).
- **Graph-theoretic centrality measures**: Utilizing graph centrality measures such information flow [4], betweenes or community detection on the sampled surface to guide more targeted attacks (Figure 4a).
- **Random walks**: Applying random walk-based dynamics [3, 1, 5] over mesh vertices as an alternative sampling strategy (Figures 4b and 4c).
- **GNN attack**: Explore different approaches based on existing attacks for Graph Neural Networks, as well as message passing methodology to explore the mesh surface and geometric characteristics.[10]

Many of the above directions were preliminarily tested—particularly sampling methods and graph-theoretic analyses—but were not integrated into the final pipeline due to the complexity of aligning them with mesh-based data formats and models in the given time window.

Overall, this project enabled us to synthesize ideas from adversarial learning, spectral graph theory, and geometric deep learning. We are grateful for the opportunity to explore this interdisciplinary space.

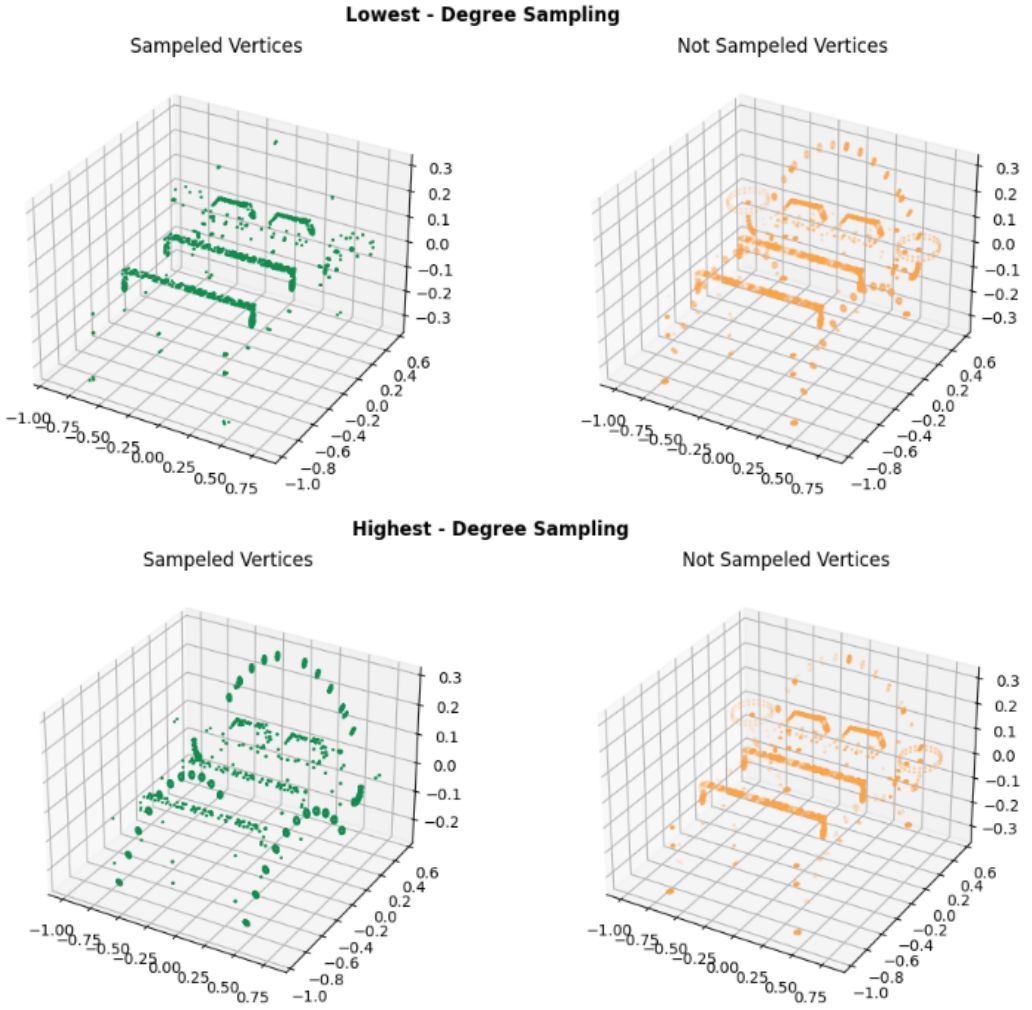
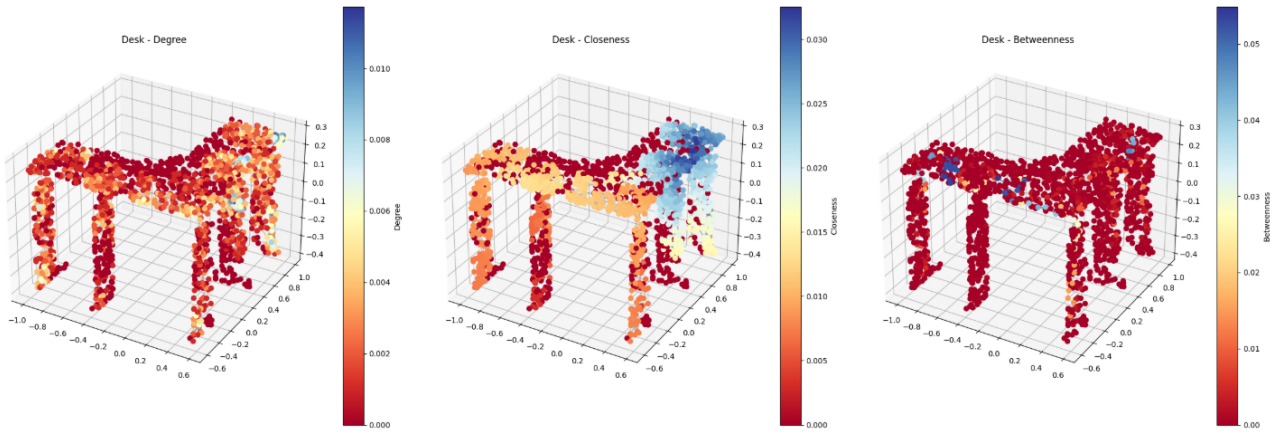
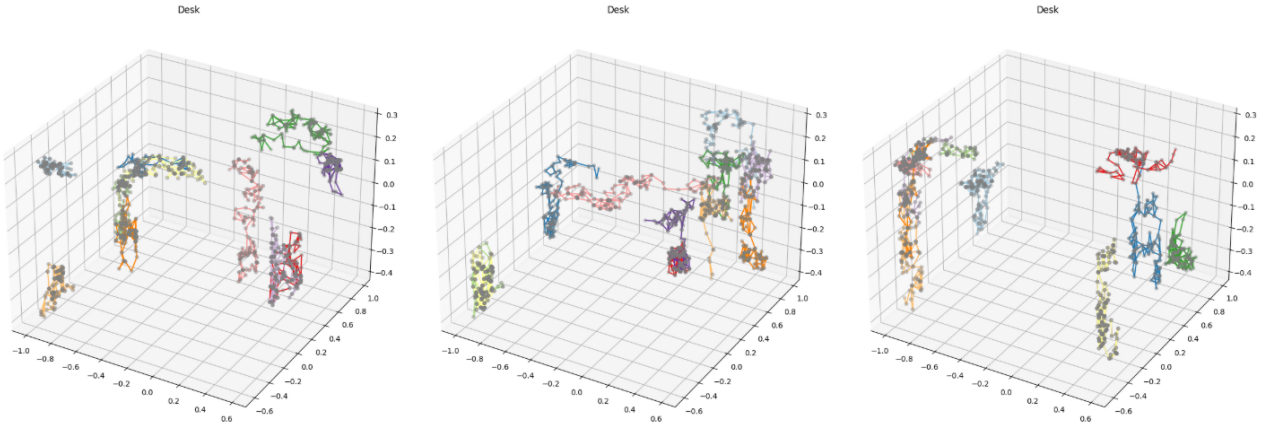


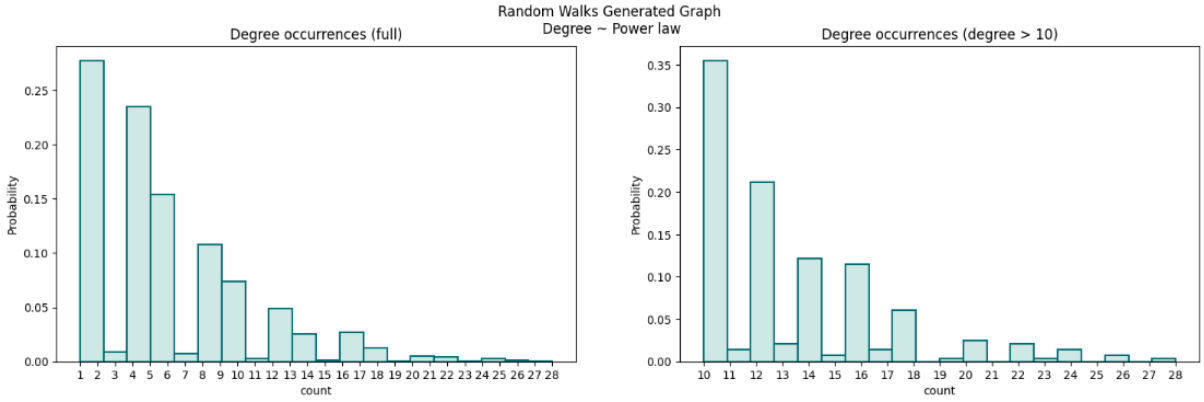
Figure 3: Comparison of degree-based sampling strategies.



(a) Centrality measures (including additional variants) on the random walk graph of the "Desk" mesh. 32 walks of length 128 were computed.



(b) Random walks (32, each of length 128) performed over the "Desk" mesh structure.



(c) Degree distribution of the random walk-induced graph.

Figure 4: Graph-theoretic analysis and sampling strategies based on random walks.

References

[1] Amir Belder, Gal Yefet, Ran Ben-Itzhak, and Ayellet Tal. Random walks for adversarial meshes. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery.

[2] Yifan Feng, Yifan Zhang, Zizhao Zhao, and Rongrong Ji. Meshnet: Mesh neural network for 3d shape representation. In *AAAI Conference on Artificial Intelligence*, 2019.

[3] Robert M. Gray and Lee D. Davisson. *Introduction*, page 1–9. Cambridge University Press, 2004.

[4] Shiyu Hu, Daizong Liu, and Wei Hu. Improving the transferability of 3d point cloud attack via spectral-aware admix and optimization designs, 2024.

[5] Gregory F. Lawler and Vlada Limic. *Random Walk: A Modern Introduction*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010.

[6] Binbin Liu, Jinlai Zhang, Lyujie Chen, and Jihong Zhu. Boosting 3d adversarial attacks with attacking on frequency. *CoRR*, abs/2201.10937, 2022.

[7] Daizong Liu, Wei Hu, and Xin Li. Point cloud attacks in graph spectral domain: When 3d geometry meets graph signal processing, 2023.

[8] Wendi Liu, Yuesheng Lu, and Huan Zhang. Aof: Attack on frequency for transferable adversarial attack against 3d point cloud classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17045–17054, 2022.

[9] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. Towards more practical adversarial attacks on graph neural networks, 2021.

[10] Bo Pang, Zhongtian Zheng, Yilong Li, Guoping Wang, and Peng-Shuai Wang. Neural laplacian operator for 3d point clouds, 2024.

[11] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017.

[12] Chengchao Xiang, Xiao Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[13] Jinlai Zhang, Lyujie Chen, Binbin Liu, Bo Ouyang, Qizhi Xie, Jihong Zhu, Weiming Li, and Yanmei Meng. 3d adversarial attacks beyond point cloud, 2021.