SAGA: Spectral Adversarial Geometric Attack on 3D Meshes

Tomer Stolik*
Tel Aviv University
tomerstolik@mail.tau.ac.il

Itai Lang*
Tel Aviv University
itailang@mail.tau.ac.il

Shai Avidan
Tel Aviv University
avidan@eng.tau.ac.il

Abstract

A triangular mesh is one of the most popular 3D data representations. As such, the deployment of deep neural networks for mesh processing is widely spread and is increasingly attracting more attention. However, neural networks are prone to adversarial attacks, where carefully crafted inputs impair the model's functionality. The need to explore these vulnerabilities is a fundamental factor in the future development of 3D-based applications. Recently, mesh attacks were studied on the semantic level, where classifiers are misled to produce wrong predictions. Nevertheless, mesh surfaces possess complex geometric attributes beyond their semantic meaning, and their analysis often includes the need to encode and reconstruct the geometry of the shape.

We propose a novel framework for a geometric adversarial attack on a 3D mesh autoencoder. In this setting, an adversarial input mesh deceives the autoencoder by forcing it to reconstruct a different geometric shape at its output. The malicious input is produced by perturbing a clean shape in the spectral domain. Our method leverages the spectral decomposition of the mesh along with additional mesh-related properties to obtain visually credible results that consider the delicacy of surface distortions[1].

1. Introduction

A triangular mesh is the primary representation of 3D shapes, with applications in many safety-critical realms. In the medical field, incorrect perception of the geometric subtleties of an organ can lead to life-threatening errors. In robotics and automotive, a precise understanding of the geometry of obstacles is essential to prevent accidents. The security of facial modeling is also dependent on the accuracy of the processed geometry of the mesh.

Autoencoders (AEs) are one of the most prominent deep-learning tools to process the mesh's geometry. They are designed to capture geometric features which enable dimensionality reduction for both storage and communication purposes [6, 4]. Mesh AEs are also used for segmentation, self-supervised learning, and denoising tasks [16, 19, 7].

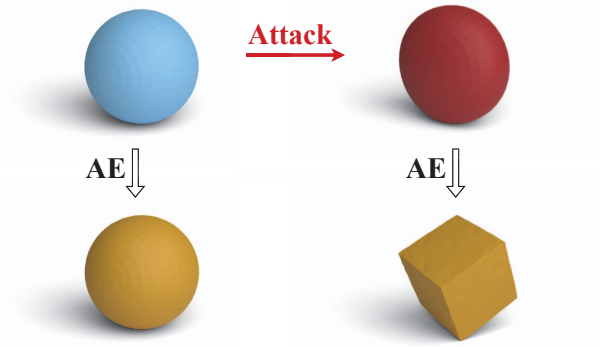

Figure 1. **A result of our geometric mesh attack.** A mesh of a *sphere* (top left) is perturbed into an adversarial example (top right). While the original mesh is accurately reconstructed by an *autoencoder* (AE) (bottom left), our attack fools the AE and changes the output geometry to a *cube*! (bottom right).

Despite their tremendous achievements, neural networks are often found vulnerable to adversarial attacks. These attacks craft inputs that impair the victim network's behavior. Adversarial attacks were extensively studied in recent years, focusing especially on the *semantic* level, where the input to a classifier is carefully modified in an imperceivable manner to mislead the network to an incorrect prediction. Semantic adversarial attacks are abundant in the case of 2D images [8, 20, 3], and recently, semantic attacks on 3D representations have also drawn much attention, both on point clouds [29, 10, 28] and meshes [30, 14, 23, 1].

Nonetheless, the vulnerabilities of networks that process geometric attributes, such as AEs, have not been thoroughly investigated. AEs may be imperative to many practical mesh deployments and their credibility and robustness depend on the study of *geometric* adversarial attacks.

We propose a framework of a geometric adversarial attack on 3D meshes. Our attack, named SAGA, is exemplified in Figure 1. The input mesh of the sphere is perturbed and fed into an AE that reconstructs a *geometrically different* output, *i.e.*, a cube! Ideally, the deformation of the input

[1] https://github.com/StolikTomer/SAGA
*Equal contribution

should be unapparent and yet effectively modify the output geometry.

In our attack, we aim to reconstruct the geometry of a *specific target* mesh by perturbing a clean *source* mesh into a malicious input. We present a white-box setting, where we have access to the AE and we optimize the attack according to its output. A black-box framework is also explored by transferring the adversarial examples to other unseen AEs.

Mesh perturbations include shifts of vertices that affect their adjacent edges and faces and possibly result in noticeable topological disorders, such as self-intersections. Therefore, concealed perturbations must address the inherent topological constraints of the mesh. To cope with the fragility of the mesh surface, we apply the perturbations in the spectral domain defined by the eigenvectors of the Laplace-Beltrami operator (LBO) [5]. Particularly, we facilitate an accelerated attack by operating in a *shared* spectral coordinate system for all shapes in the dataset. The source's distortions are retained by using low-frequency perturbations and additional mesh-related regularizations.

The attack is tested on datasets of human faces [24] and animals [32]. We evaluate SAGA using geometric and semantic metrics. Geometrically, we measure the similarity between shapes by comparing the mean curvature of matching vertices. Semantically, we use a classifier to predict the labels of the adversarial reconstructions, and a detector network to demonstrate the difficulty of identifying the adversarial shapes. We also conduct a thorough analysis of the attack and a comprehensive ablation study.

To summarize, we are the first to propose a *geometric* adversarial attack on 3D meshes. Our method is based on low-frequency spectral perturbations and regularizations of mesh attributes. Using these, SAGA crafts adversarial examples that change an AE's output into a different geometric shape.

2. Related Work

Spectral mesh analysis. The vertices and triangular faces of a mesh define a discrete approximation of a 2D surface [17]. The spectral analysis of continuous 2D manifolds is derived from the Laplace-Beltrami operator (LBO), which is a generalization of the Laplacian from the Euclidean setting to curved surfaces. The eigenfunctions of the LBO form an orthogonal basis that spans signals upon the shape's surface.

Taubin [26] was the first to introduce the spectral analysis of meshes by exploring the notion of a discrete LBO. Pursuing research [17, 13] suggested using the classic cotangent scheme [21] to construct the LBO. In this case, the operator is more robust against differences in mesh discretization. Consequently, the LBO eigenvectors are approximate samples of the continuous eigenfunctions on the vertices of the mesh [13]. Based on this analysis, we utilized the spectral basis of the mesh to perform our attack.

Mesh autoencoders. Nowadays, a prevailing 3D learning technique employs AE networks that learn to encode geometric shapes into a latent space and reconstruct them. Marin *et al*. [15] used a multilayer-perceptron (MLP) AE to establish a latent representation of the mesh, and then exploited it in an additional pipeline to recover a shape from its LBO spectrum.

A popular mesh AE was presented by Ranjan *et al*. [24], where spectral convolution layers and mesh sampling methods achieved promising results on human face data. Further work suggested using spiral convolution operators [2], while Zhou *et al*. [31] used a fully convolutional architecture with a spatially varying kernel to handle irregular sampling density and diverse connectivity. All the mentioned AEs operate on the mesh vertices, assuming a known connectivity, to successfully reconstruct the surface. We used Marin's AE [15] as our victim model, and we explore the attack transferability to the CoMA AE [24].

3D adversarial attacks. In recent years, the research of adversarial attacks on 3D data has expanded, focusing almost entirely on semantic attacks that aim to malfunction classifiers. The literature on semantic adversarial attacks of point clouds is vast. A common approach [29, 10] is to refer to the perturbation as shifts or additions of outlier points in the 3D Euclidean space.

On the contrary, semantic mesh attacks often leverage properties derived from the connectivity of the vertices. Belder *et al*. [1] introduced the concept of random walks on the mesh surface to create adversarial examples. Other papers [14, 23] addressed semantic attacks in the spectral domain. Mariani *et al*. [14] used band-limited perturbations and extrinsic restrictions to cause misclassifications. Rampini *et al*. [23] suggested a universal attack by applying a purely intrinsic regularization on the spectrum of the adversarial shape.

The work most similar to ours is the geometric point cloud attack proposed by Lang *et al*. [12]. To our knowledge, this is the only geometric attack on 3D shapes. Lang *et al*. demonstrated the ability to reproduce a different geometry by feeding an AE with a malicious input shape. However, that work focused on point clouds. It used vertex displacements in the 3D Euclidean space and exploited the lack of connectivity and order to construct adversarial examples.

In contrast, our work is oriented to 3D meshes. Unlike point clouds, meshes have topological constraints. Hence, swaps of vertices' locations or local shifts of vertices are highly noticeable. We leverage the connectivity to operate in the spectral domain where we control global attributes across the shape and better preserve the geometry of the original surface.

3. Method

We attack an autoencoder (AE) trained on a collection of shapes from several semantic classes. In each attack, we use a single source-target pair, where the source and target shapes are selected from different classes. Our goal is to find a perturbed version of the source, with minimal distortion, that misleads the AE to reconstruct the target. Ideally, the source's perturbations should be invisible while still altering the AE's output to the geometry of the target shape.

Given an attack setup of a source shape and a target class, we choose, as a pre-processing step, the nearest neighbor shape from the target class in the sense of a Euclidean norm of the difference between matching vertices. Since the AE is sensitive to the geometry of its input, selecting a target that is geometrically similar to the source benefits the attack and reduces the potential magnitude of the perturbation.

In the upcoming subsections, we present a preliminary spectral analysis followed by a description of the spectral domain in which the attack is performed. Then, we define the problem statement and elaborate on the perturbation parameters, the loss function, and the evaluation metrics.

3.1. Preliminaries

Manifolds. A geometric shape can be described as a 2D Riemannian manifold $\mathcal{X}$ embedded in the 3D Euclidean space $\mathbb{R}^3$ [17]. Let $\Delta_{\mathcal{X}}$ be the Laplace-Beltrami operator (LBO) of the manifold $\mathcal{X}$, which is a generalization of the Laplacian operator to the curved surface. The LBO admits an eigendecomposition of the shape into a set of discrete eigenvalues $\{\lambda_i\}$, known as the spectrum of the shape, and a set of eigenfunctions $\{\phi_i\}$, as follows:

$$\Delta_{\mathcal{X}}\phi_i = \lambda_i\phi_i. \tag{1}$$

The eigenfunctions $\{\phi_i\} : \mathcal{X} \to \mathbb{R}$ form an orthogonal spectral basis of scalar functions. Thus, the Euclidean embedding values of the manifold in the x, y, z axes can be represented as three linear combinations of the spectral basis using a set of corresponding *spectral coefficients* $\{\alpha_{i,x}\}, \{\alpha_{i,y}\}, \{\alpha_{i,z}\}$.

Mesh graphs. A continuous manifold of a 3D shape can be discretized into a triangular mesh graph $M = (V, F)$. $V \in \mathbb{R}^{n\times 3}$ is the vertices matrix, in which each of the n vertices is assigned a 3D Euclidean coordinate. $F \in \mathbb{R}^{m\times 3}$ is the triangular faces matrix consisting of m triplets of vertices. We calculate the discrete LBO using the prevailing classic cotangent scheme [21]. In this case, the LBO is an $n \times n$ matrix and the eigenvectors are approximated samples of the continuous eigenfunctions on the vertices of the mesh graph [13]. Let us arrange the eigenvectors as the columns of $\Phi \in \mathbb{R}^{n\times n}$ and the n spectral coefficients of each Euclidean axis as the columns of $A \in \mathbb{R}^{n\times 3}$. Then, the spectral representation of the mesh vertices is given by:

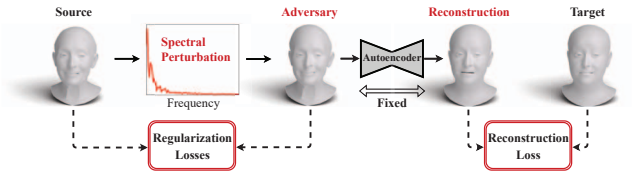

Figure 2. **The proposed attack framework.** Attack parameters perturb the spectral coefficients of the source shape to craft an adversarial example. The malicious input (Adversary) misleads the AE to reconstruct the geometry of the target mesh. The perturbation is optimized using a loss function that compares the AE's output with the target shape, and regularizes the adversarial shape to preserve the source's geometric properties.

$$V = \Phi A. \tag{2}$$

3.2. Shared Spectral Representation

The spectral decomposition of a mesh is computationally demanding, and it is restraining the efficiency of our attack. Thus, we propose a novel approach in which the attack is performed in a shared spectral domain. The idea is to represent all the attacked shapes in a shared coordinate system defined by a single set of spectral eigenvectors. This shared basis accelerates the attack by omitting the heavy calculations of a per-shape spectral decomposition.

Shared spectral basis. The spectral decomposition varies between different shapes since the surface of each shape is a unique manifold and its spectral eigenfunctions are defined over its specific geometric domain. However, the geometric resemblance of the shapes in the dataset can be utilized to construct a shared basis of eigenvectors. The idea of a shared set of eigenvectors assures that, practically, the Euclidean coordinates of the vertices of any shape can be spanned by the shared basis with a negligible error.

The shared basis was built as a linear combination of the bases of multiple shapes, which were sampled from different classes. The coefficients of the linear combination were optimized using gradient descent. The loss function was the sum, across all sampled shapes, of the mean-vertex Euclidean distance between the original coordinates and their representation in the shared spectral domain. More details can be found in the supplementary.

Basis transformation. We denote the shared basis by $\Phi_{shared} \in \mathbb{R}^{n\times n}$, where its columns are the set of n shared eigenvectors. In the new coordinate system, the vertex matrix V of a mesh M can be replaced by the spectral coefficients matrix $A' \in \mathbb{R}^{n\times 3}$ according to:

$$V = \Phi_{shared} A'. \tag{3}$$

Given Φ_{shared} and V, the spectral coefficients are found using least squares. In the following sections, we refer to A'

simply as A for ease of notation and assume it was calculated using Φ_{shared}.

3.3. Attack

We pose the attack as an optimization problem in a white-box framework, where the AE is fixed. We denote the source mesh taken from class $\mathcal{S}$ by $M_{\mathcal{S}} = (V_{\mathcal{S}}, F_{\mathcal{S}})$, and the target mesh taken from class $\mathcal{T}$ by $M_{\mathcal{T}} = (V_{\mathcal{T}}, F_{\mathcal{T}})$. The spectral representations of $V_{\mathcal{S}}$ and $V_{\mathcal{T}}$ are given by the spectral coefficients matrices $A_{\mathcal{S}}$ and $A_{\mathcal{T}}$, as defined in Equation 3. Let us denote by k the number of frequencies we aim to perturb. We add perturbation parameters from $B \in \mathbb{R}^{k\times 3}$ to obtain the adversarial input A_{adv}, according to:

$$A_{adv}(i) = \begin{cases} A_{\mathcal{S}}(i) + B(i), & \text{if } i < k \\ A_{\mathcal{S}}(i), & \text{otherwise,} \end{cases} \tag{4}$$

where $A_{\mathcal{S}}(i) = [\alpha_{i,x}, \alpha_{i,y}, \alpha_{i,z}] \in \mathbb{R}^3$ and $B(i) = [\beta_{i,x}, \beta_{i,y}, \beta_{i,z}] \in \mathbb{R}^3$ are the spectral coefficients of frequency i and their perturbation parameters, respectively. Note that the optimized parameters of the attack are the elements of B. The resulting adversarial mesh is $M_{adv} = (V_{adv}, F_{\mathcal{S}})$, where $V_{adv} = \Phi_{shared} A_{adv}$. Also, we propose an attack with a multiplicative perturbation, defined as:

$$A_{adv}(i) = \begin{cases} A_{\mathcal{S}}(i)(1 + B(i)), & \text{if } i < k \\ A_{\mathcal{S}}(i), & \text{otherwise.} \end{cases} \tag{5}$$

The advantages of operating in the spectral domain are realized by confining the attack to a limited range of low frequencies. By attacking only the low frequencies, we inherently enforce smooth surface perturbations and reduce sharp local changes of the curvature. Consequently, significantly fewer parameters are used compared to a Euclidean space attack where all vertices are shifted. It also offers the flexibility to control the number of optimized parameters.

Problem statement. The problem statement is depicted in Figure 2. The parameters of the perturbation B are optimized according to the following objective:

$$\begin{aligned} \underset{B}{\text{argmin}} \quad & \mathcal{L}_{recon}(\widehat{M}_{adv}, M_{\mathcal{T}}) + \mathcal{L}_{reg}(M_{adv}, M_{\mathcal{S}}) \\ \text{s.t.} \quad & \widehat{M}_{adv} = f_{AE}(M_{adv}), \end{aligned} \tag{6}$$

where f_{AE} is the AE model and $\widehat{M}_{adv}$ is the reconstruction of M_{adv} by f_{AE}. $\mathcal{L}_{recon}$ and $\mathcal{L}_{reg}$ are the loss terms for the target reconstruction and the perturbation regularization, correspondingly. Both terms are further discussed next.

Reconstruction and regularization losses. The reconstruction of a target shape is achieved by explicitly minimizing the Euclidean distance between the vertices of the AE's output and the vertices of the clean target mesh. Specifically, $\mathcal{L}_{recon}$ is defined as:

$$\mathcal{L}_{recon} = \frac{1}{n} \sum_{i=1}^{n} \left\| \widehat{V}_{adv}(i) - V_{\mathcal{T}}(i) \right\|_2^2. \tag{7}$$

where $\widehat{V}_{adv}(i), V_{\mathcal{T}}(i) \in \mathbb{R}^3$ are the 3D coordinates of vertex i in meshes $\widehat{M}_{adv}, M_{\mathcal{T}}$, respectively. The sign $\|\cdot\|_2$ refers to the l_2-norm.

To alleviate the distortion of the source shape, we combine the inherent smoothness provided by the spectral perturbations with the $\mathcal{L}_{reg}$ loss. This loss consists of additional mesh-oriented regularizations that are meant to prevent abnormal geometric distortions.

We consider four kinds of regularization measures in $\mathcal{L}_{reg}$, each with a different weight assigned to it. Inspired by Sorkine [25], the first term, denoted by $\mathcal{L}_{lap}$, compares the shapes in a non-weighted-Laplacian representation. In this representation, a vertex $V(i)$ is represented by the difference between $V(i)$ and the average of its neighbors. This loss promotes smooth perturbations since it considers the relative location of a vertex compared to its neighbors. Let I be an identity matrix of size $n \times n$, J be the mesh adjacency matrix, and $D = diag(d_1, ..., d_n)$ be the degree matrix. Then, the non-weighted Laplacian operator, L_{non}, is defined as $L_{non} = I - D^{-1}J$, and the vertices matrix is transformed into $\tilde{V} = L_{non}V$. The loss $\mathcal{L}_{lap}$ is defined as:

$$\mathcal{L}_{lap} = \frac{1}{n} \sum_{i=1}^{n} \left\| \tilde{V}_{adv}(i) - \tilde{V}_{\mathcal{S}}(i) \right\|_2^2. \tag{8}$$

The second regularization term, $\mathcal{L}_{area}$, reduces the Euclidean distance between matching vertices, normalized by the total surface area of all the triangles containing the vertex in the clean source shape. The loss $\mathcal{L}_{area}$ retains changes in heavily sampled regions of high curvature, a vital requirement for geometric details preservation. It is defined as:

$$\mathcal{L}_{area} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{area(i)} \left\| V_{adv}(i) - V_{\mathcal{S}}(i) \right\|_2^2, \tag{9}$$

where $area(i)$ is a weight defined by the sum of the surface area of all the faces containing vertex i in $M_{\mathcal{S}}$.

Let us denote by $N(M) \in \mathbb{R}^{m\times 3}$ the normal vectors of all the faces of mesh M and by $E(M) \in \mathbb{R}^d$ the length of all the edges of mesh M, where d is the number of edges. The third and fourth regularization terms in $\mathcal{L}_{reg}$ are denoted by $\mathcal{L}_{norm}$ and $\mathcal{L}_{edge}$, and are defined as follows:

$$\mathcal{L}_{norm} = \frac{1}{m} \sum_{i=1}^{m} \left\| N(M_{adv})(i) - N(M_{\mathcal{S}})(i) \right\|_2^2, \tag{10}$$

$$\mathcal{L}_{edge} = \frac{1}{d}\sum_{i=1}^{d} |E(M_{adv})(i) - E(M_{\mathcal{S}})(i)|^2. \quad (11)$$

The loss $\mathcal{L}_{norm}$ prevents the formation of sharp curves in the adversarial mesh by limiting the deviation of the surface's normal vectors. It is particularly beneficial when the geometric differences between the source and target shapes are coarse. The loss $\mathcal{L}_{edge}$, on the other hand, alleviates local stretches and volumetric changes by keeping the edges' length from changing. Referring to the problem statement in Equation 6, we define $\mathcal{L}_{reg}$ as:

$$\mathcal{L}_{reg} = \lambda_l \mathcal{L}_{lap} + \lambda_e \mathcal{L}_{edge} + \lambda_a \mathcal{L}_{area} + \lambda_n \mathcal{L}_{norm}, \quad (12)$$

where λ_l, λ_e, λ_a, and λ_n are the loss terms' weights.

3.4. Evaluation Metrics

A geometric attack on a mesh AE copes with a built-in trade-off between the need to confine the deformation of the source shape and the requirement to reconstruct the geometry of the different target shape using the AE. We present geometric and semantic quantitative metrics to evaluate these contradicting necessities.

To geometrically quantify the difference between shapes, we consider a *curvature distortion* measurement, defined as the absolute difference between the mean curvature of matching vertices in the compared shapes. This metric is typically used in semantic mesh adversarial attacks [14, 23]. We use the per-vertex curvature distortion to present heatmaps on the adversarial examples in our visualizations. A complete evaluation of the curvature distortion caused by our attack is reported in the supplementary.

We introduce a semantic evaluation of the adversarial reconstructions and a semantic interpretation of the extent to which the source shape was corrupted. To identify the AE's output, we use a classifier and report the accuracy of labeling the adversarial reconstructions with the target's label. We consider two settings, a targeted and an untargeted classification. In the targeted case, we check whether $\widehat{M}_{adv}$ is labeled as a shape from the target class $\mathcal{T}$. In the untargeted case, we only check if $\widehat{M}_{adv}$ is *not* labeled as a shape from the source class $\mathcal{S}$, which means the semantic identity of the malicious input was altered by the AE.

To appreciate the challenge of detecting adversarial geometric shapes, take the challenge quiz in Figure 3. Can you detect which shapes are clean and which ones are not? We estimate the noticeability of the perturbation by training a detector network in a binary classification task. The goal is to determine if a certain shape is an adversarial example or not. The detector's accuracy is used as a metric, where a lower score means a better attack.

[2](1) original. (2) adversary. (3) adversary. (4) adversary. (5) original. (6) adversary.

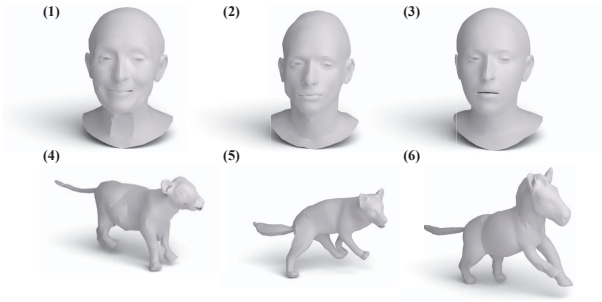

Figure 3. **Attack detection quiz.** Which shape is an original mesh from the dataset, and which is an adversarial example of SAGA? The answers can be found in the footnote[2].

A dataset of clean source shapes and their perturbed counterparts was constructed for the detection task, where all shapes were originally selected from the AE's test set. The detector was validated and tested using a leave-one-out method, in which shapes from all classes but one were used as the train set. Shapes from the remaining class were split into validation and test sets. For an unbiased comparison, we repeated the experiment multiple times, and each time a different class was excluded for validation and testing. The reported results are an average of all the experiments. A full description of the architecture and the training process appears in the supplementary.

4. Results

4.1. Experimental Setup

The attack was evaluated on the CoMA dataset of human faces [24] and on the SMAL animals dataset [32]. Both datasets are commonly used in the literature [14, 15, 23, 9, 2, 1]. We attacked the mesh AE proposed by Marin *et al*. [15]. The AE was trained using the same settings as in the original paper for both datasets. During the attack, the AE's weights were frozen, and we used only source and target shapes from the test set.

CoMA. We used 8325 examples to train the AE, 926 for validation, and 1398 for the test set, where all the sets included instances from 11 different semantic identities. Shapes from the 12^{th} identity were used for an out-of-distribution experiment. During the attack, only the first 500 frequencies were perturbed with an additive perturbation, as shown in Equation 4. The attack parameters were optimized over 500 gradient steps using Adam optimizer with a learning rate of 0.0001. We regularized the perturbation using three loss terms, $\mathcal{L}_{lap}$, $\mathcal{L}_{edges}$, and $\mathcal{L}_{area}$, with the corresponding weights $\lambda_l = 100$, $\lambda_e = 2$, and $\lambda_a = 500$.

SMAL. We used the SMAL parametric model to gen-

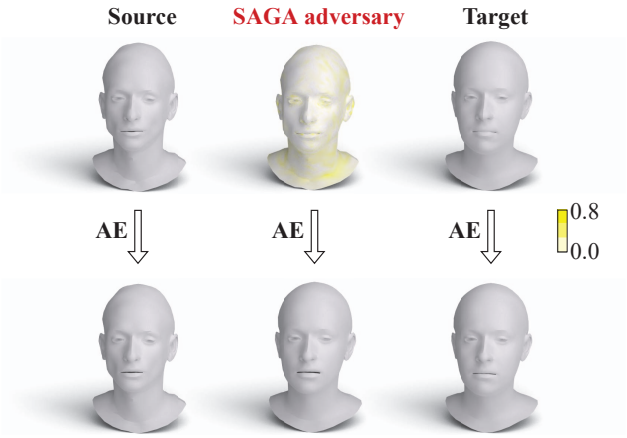

Figure 4. **Comparison to a clean target reconstruction.** Top row, left to right: the clean source mesh, SAGA's adversarial example, and the clean target mesh. Bottom row: the reconstructions of the shapes from the top row after passing through the AE. Note that the source has a different identity than the target, with sharper facial features. The heatmap encodes the per-vertex curvature distortion values between the adversarial example and the source mesh, growing from white to yellow. Our mild perturbation of the source human face leads to the reconstruction of a different identity, which is similar to the reconstruction of the clean target.

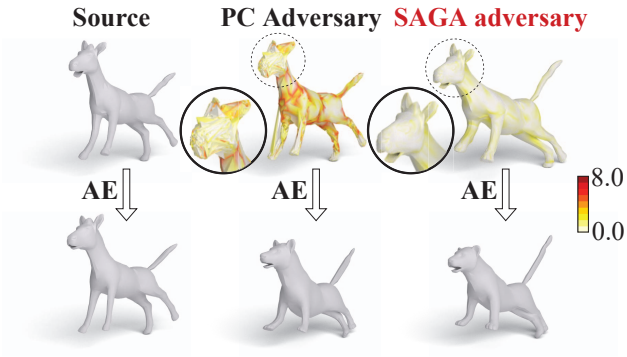

Figure 5. **Geometric attacks comparison.** Top row, left to right: the clean source mesh (*a horse*), the adversarial example produced by a geometric point cloud (PC) attack [12], and SAGA's adversarial example. Bottom row: the reconstructions of the shapes from the top row after passing through the AE. The heatmap encodes the per-vertex curvature distortion values between each adversarial example and the clean source shape, growing from white to red. SAGA's perturbation slightly changes the horse's pose while preserving its geometry. The adversarial horse misleads the AE to reconstruct the geometry of a target *leopard* shape. In contrast, the PC attack causes apparent surface distortions to the source mesh by switching vertices' locations (as seen in the inset), and its reconstruction lacks the fine-grained details of the target mesh.

erate 9918 shapes of the 5 animal species, divided into $85\%/10\%/5\%$ for train/validation/test. The variance between classes in the SMAL data required changes in the optimization process compared to the CoMA data. Following Equation 5, we performed a multiplicative attack to gain gradual perturbation refinements. We perturbed the eigenvectors of the first 2000 frequencies. The attack was optimized using the Adam optimizer with a learning rate of 0.01 over 3000 gradient steps. We used three regularization terms, $\mathcal{L}_{lap}, \mathcal{L}_{edges}$, and $\mathcal{L}_{norm}$, with the corresponding weights $\lambda_l = 50, \lambda_e = 5$, and $\lambda_n = 0.5$.

The attack setup included 50 source shapes from each class, paired with a single target shape from each of the other classes. This sums up to $50 \cdot 11 \cdot 10 = 5500$ attacked pairs for CoMA and $50 \cdot 5 \cdot 4 = 1000$ attacked pairs for SMAL. The average attack duration using an Nvidia Geforce GXT 1080Ti was $2.4/13.2$ seconds per pair in the CoMA/SMAL datasets, correspondingly.

We compared our results with the point cloud (PC) attack suggested by Lang *et al*. [12]. For a fair comparison, we used the same reconstruction loss as in Equation 7. The perturbations were applied as shifts of vertices in the Euclidean space, and we used the Chamfer Distance as the regularization loss, as explained in their paper.

4.2. Perceptual Evaluation

A visual demonstration of our attack appears in Figure 4. We optimize the changes to the clean source human face such that the AE reconstructs the desired target shape. Restricting the attack to a set of low mesh frequencies, combined with the explicit spatial regularization, maintains the similarity to the source and keeps the natural appearance of the adversarial example.

We compare our attack to the PC geometric attack proposed by Lang *et al*. [12]. Figure 5 exhibits a visual comparison. Lang *et al*.'s attack, being adjusted to point clouds, caused a distinctive surface corruption by replacing the order of the vertices. On the contrary, our SAGA reached better target reconstructions with perturbations that preserve the underlying surface.

We used a PointNet classifier [22] to semantically evaluate the adversarial reconstructions. The classifier was trained, validated and tested by the same sets as our victim AE. We trained the model over 1000 epochs using the same loss function and optimizer as in Rampini *et al*.'s work [23].

Table 1 shows the accuracy obtained from classifying the adversarial reconstructions as the target, in the targeted case, or differently from the source, in the untargeted case. The experiment included all the attacked pairs. We compare our attack with Lang *et al*.'s PC attack [12] and with the clean target reconstructions.

The results of Table 1 demonstrate that our attack is also effective on the semantic level. SAGA consistently reached a higher target classification accuracy compared to Lang *et al*.'s attack. On the CoMA dataset, SAGA reached over 99% accuracy in all cases. The results were lower on the

Input Type	Targeted ↑	Untargeted ↑
Clean target (CoMA)	100%	100%
PC attack [12] (CoMA)	96.22%	98.05%
SAGA - ours (CoMA)	**99.31%**	**99.82%**
Clean target (SMAL)	99.80%	100%
PC attack [12] (SMAL)	46.70%	74.90%
SAGA - ours (SMAL)	**67.00%**	**82.50%**

Table 1. **Semantic interpretation.** The table shows the classification accuracy of the AE's outputs given different inputs. We report the accuracy of labeling the reconstructions as the target class (targeted case) or as any class besides the source class (untargeted case). The adversarial reconstructions of SAGA are compared to those of the point cloud (PC) attack [12] and to the reconstructions of the clean targets. SAGA consistently outperforms the PC attack on both datasets. The lower accuracy rates on the SMAL [32] dataset stem from the large geometric differences between the source and target shapes.

Attack Type	Detection Accuracy ↓
PC attack [12] (CoMA)	98.56%
SAGA - ours (CoMA)	**53.69%**
PC attack [12] (SMAL)	90.90%
SAGA - ours (SMAL)	**49.80%**

Table 2. **Attack detection.** We report the accuracy of a detector trained to differentiate between adversarial examples and clean inputs. We compare the detection of SAGA to the point cloud (PC) attack [12]. Details about the training and test procedures appear in Sections 3.4 and 4.3. Low detection accuracies correspond with a better, unapparent attack. The results demonstrate the difficulty of distinguishing SAGA's adversarial shapes, in contrast to the distinct recognition of the PC attack.

SMAL dataset due to the disparity between the different classes. The classifier labeled 67% of SAGA's adversarial reconstructions of animals as the target class. In 82% of the cases, the adversarial reconstructions were classified differently from their source class. In contrast, the PC attack reached a lower accuracy, less than 50% and 75% in the targeted and untargeted settings, respectively.

4.3. Attack Detection

We semantically examined the malicious inputs using a detector network. The detector was separately trained to identify the adversarial shapes of SAGA and the PC attack. We used an MLP architecture to consider the connectivity of the vertices in each mesh. Since the objective of the attack is to create invisible perturbations, a *lower* accuracy rate corresponds to better adversarial examples.

The results of both ours and the PC attack [12] appear in Table 2. The detector failed to spot SAGA's perturbations, reaching less than 55% detection accuracy on both datasets. On the other hand, the PC attack was distinctive to the detector. Over 98% of the shapes from CoMA and over 90% of the shapes from SMAL were classified correctly. Therefore, we quantitatively demonstrate the efficiency of SAGA in constructing untraceable malicious inputs. We show that a trained network successfully detects another attack but still fails to identify SAGA's adversarial examples.

4.4. Comparison to Semantic Attacks

The literature on semantic adversarial attacks on 3D meshes is abundant [30, 14, 23, 1]. Semantic attacks are aimed against classifiers, where adversarial shapes induce misclassifications. An interesting experiment is to check whether a semantic attack is also effective as a geometric attack on an AE. To this end, we applied the semantic attacks of Rampini *et al*. [23] and Huang *et al*. [11] on our data to produce semantic adversarial examples, and we analyzed their impact on the AE.

Using Rampini *et al*.'s framework, we attacked the same animal shapes [32] that were used for SAGA. That is, the attacked set included 250 animal shapes, consisting of 50 source shapes from each of the 5 animal classes. We attacked the pre-trained PointNet classifier [22] that was presented in Section 4.2. This classifier was also used in Rampini *et al*.'s original paper [23], and it was trained, evaluated, and tested using the same sets as our AE. The classifier obtained 99.2% accuracy on the clean shapes. All shapes were originally selected from the classifier's test set.

Although Rampini *et al*. suggested a universal attack that may be applied to new unseen shapes, we optimized their attack on our specific meshes for a fair comparison. The semantic adversarial meshes were fed through our victim AE and we compare the attack's success rate before and after the AE. The success rate is defined as the accuracy of predicting a different label than the source's label.

A visual demonstration of using Rampini *et al*. [23]'s semantic adversarial shapes against the AE is depicted in Figure 6. The semantic attack altered the labels of its adversarial shapes in 86% of the cases. However, after passing through the AE, the success rate dropped to only 1.6%, as opposed to 82.5% of SAGA's reconstructions. Figure 6 demonstrates that the semantic attack fails at the geometric level, as the AE's output remains similar to the source shape. These results show that the semantic attack is ineffective geometrically since it fails to alter the AE's output. In contrast, SAGA is successful in both the geometric and semantic aspects. A comparison of our attack to Huang *et al*. [11]'s semantic attack shows similar results, and it appears in the supplementary.

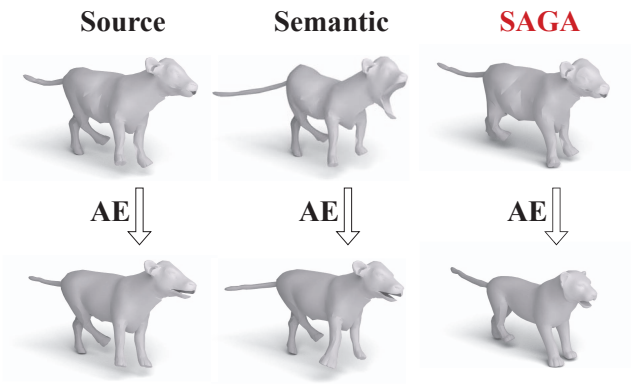

Figure 6. **A comparison to a semantic attack.** Top row, left to right: the clean source mesh (*a cow*), the adversarial example produced by a semantic mesh attack [23], and SAGA's adversarial example. Bottom row: the reconstructions of the shapes from the top row after passing through the AE. SAGA's adversarial cow successfully misleads the AE to reconstruct the geometry of a target *leopard shape*. However, in contrast to our attack, the reconstructed *shape* of the semantic adversarial mesh remains similar to the source.

4.5. Transferability

A common test for an adversarial attack is to check its efficiency on an unseen model. In the following experiment, we explored a black-box framework, where the adversarial shapes are used against a different AE than the one they were designed for. We used two unseen AEs. The first has the same architecture as our victim AE but was trained with another random weight initialization. The second is the popular AE proposed by Ranjan *et al*. [24].

A visual example is presented in Figure 7. It demonstrates that malicious shapes that were crafted to deceive one AE may change the output of other AEs to the target's geometry. Therefore, SAGA can be transferred to other AEs and still be effective in a black-box setting. More details on the transferred attack can be found in the supplementary.

4.6. Attacking a Defended AE

To present the robustness of our attack, we tested its efficiency against a defense method. We employed SAGA on an AE defended by the method of Naderi *et al*. [18]. According to their approach, we applied a Gaussian low pass filter on our training set and trained the AE with the low pass filtered shapes. Figure 8 shows an example of this experiment.

An underlying assumption of the proposed defense [18] is that an adversarial attack perturbs the high frequencies of the shape. However, our attack does the exact opposite: it is applied *only* to the low mesh frequencies. Thus, SAGA remains highly effective against the defended AE.

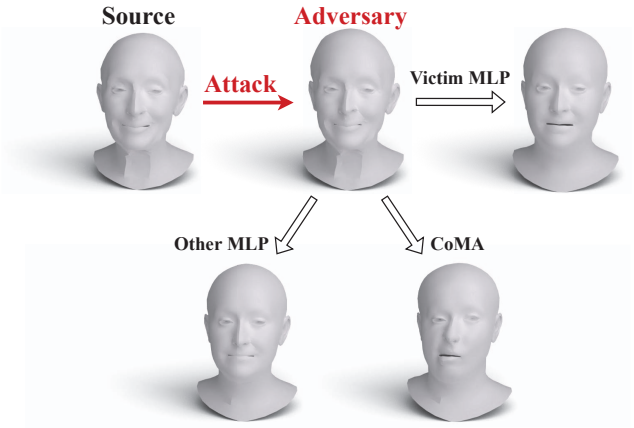

Figure 7. **Attack transferability.** A source shape (top left) is perturbed by SAGA into an adversarial example (top middle). The adversarial shape passes through three different AEs. The first (top right) is the victim AE used in the attack, with a multilayer perceptron (MLP) architecture (denoted as Victim MLP). The second AE (bottom left) has the same MLP architecture but was trained with a different random weight initialization (denoted as Other MLP). The third (bottom right) is a convolutional AE [24] (denoted as CoMA). The three AEs reconstruct the same target identity, and CoMA changes the facial expression of the shape.

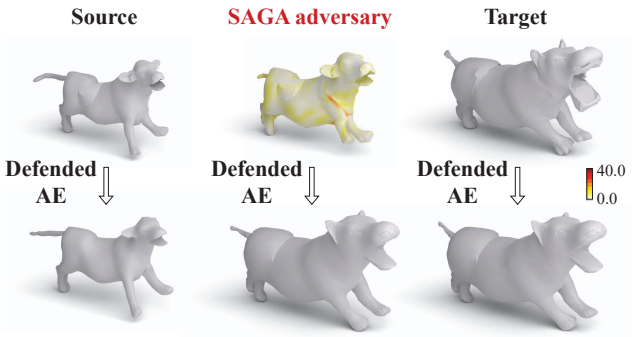

Figure 8. **Attack against a defense.** The defended AE was trained on shapes with low frequencies [18] and outputs a smoother version of clean inputs (left and right). Our attack is resilient to this defense and successfully alters the reconstructed geometry (center).

4.7. Spectral Analysis

We analyze the behavior of the spectral perturbation by measuring its magnitude in each frequency. Recall the notation of the spectral coefficients and their perturbation parameters from Equation 4. We define their magnitudes in frequency i as:

$$\alpha(i) = \sqrt{\alpha_{i,x}^2 + \alpha_{i,y}^2 + \alpha_{i,z}^2}, \tag{13}$$

$$\beta(i) = \sqrt{\beta_{i,x}^2 + \beta_{i,y}^2 + \beta_{i,z}^2}. \tag{14}$$

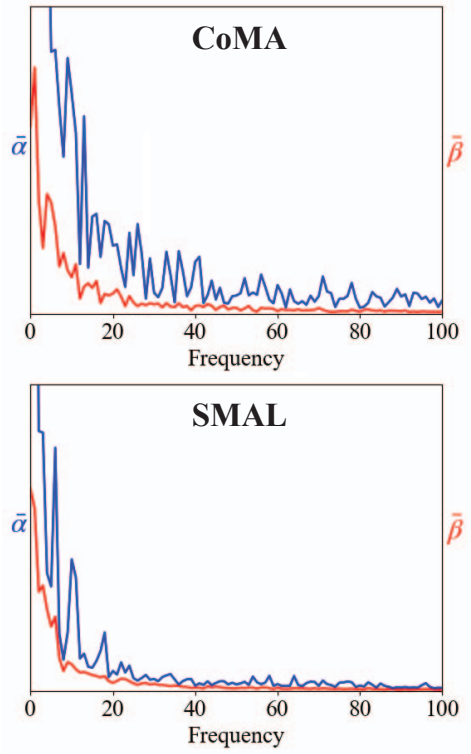

Figure 9. **Spectral analysis**. The graphs show the magnitudes of the spectral coefficients ($\bar{\alpha}$) and their perturbation factors ($\bar{\beta}$) for each frequency, as defined in Equations 13 and 14. The upper graph relates to the results on the CoMA [24] dataset, and the lower graph relates to the results on the SMAL [32] dataset. The values are averaged over all the attacked shapes from each dataset. For visual purposes, we truncate the graphs at the frequency 100. The perturbation's magnitude follows the natural spectral behavior of the data. SAGA preserves the higher mesh frequencies, keeping its fine geometric details almost intact.

Figure 9 shows the average values of $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^n$ over all the attacked pairs, denoted as $\bar{\alpha}$ and $\bar{\beta}$. The perturbation's magnitude follows the natural spectral behavior of the data in both datasets. The graphs demonstrate the attack's emphasis on lower frequencies. By preserving the higher mesh frequencies, SAGA keeps the fine geometric details of the source shape.

4.8. Additional Experiments

In the supplemental material, we analyze the AE's latent space and show the adversarial latent representations. Also, we conduct an out-of-distribution experiment, where we use a new semantic class that was not part of the AE's training set. We expose the difficulty of reconstructing its unfamiliar figure but the simplicity of altering the geometry of such an unseen identity. As part of a thorough ablation study, we change the regularizations, the number of eigenvectors, and the attacked space. We also present the speed and performance of our attack compared to a spectral attack without a shared basis.

5. Ethical Considerations

Deep Learning for mesh processing has made great progress in recent years. The attack we propose is designed to highlight vulnerabilities in existing methods in hopes of better understanding these models. We acknowledge that such methods can be used negatively in the wrong hands. We hope that shedding light on these vulnerabilities will encourage research on ways to address them.

6. Conclusions

We introduced a novel geometric attack on a 3D mesh autoencoder (AE). While previous research mostly focused on semantic attacks on classifiers, our method produced malicious inputs that aim to modify the geometry of an AE's output. A previous geometric attack on point clouds utilized the lack of connectivity between points to form adversarial examples. In contrast, a mesh attack is constrained to preserve the delicate structure of the surface to avoid noticeable perturbations. Our method yielded smooth low-frequency perturbations, and leveraged different mesh attributes to regularize apparent malformations.

We showed that our attack is highly effective in a white-box setting by testing it on datasets of human faces and animals. Semantic and geometric evaluation metrics demonstrated that SAGA's perturbations are hard to detect, while effectively changing the geometry of the AE's output. Our attack outperformed the point cloud attack in all the experiments. Further analysis explored our attack in a black-box scenario, where we demonstrated that SAGA's adversarial shapes are effective against other unseen AEs.

Acknowledgment. This work was partly funded by ISF grant number 1549/19.

References

[1] Amir Belder, Gal Yefet, Ran Ben-Itzhak, and Ayellet Tal. Random Walks for Adversarial Meshes. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1, 2, 5, 7

[2] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7213–7222, 2019. 2, 5

[3] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1

[4] Chen, Zhiqin and Tagliasacchi, Andrea and Zhang, Hao. Learning Mesh Representations via Binary Space Partition-

ing Tree Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[5] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976. 2

[6] Gao, Lin and Yang, Jie and Wu, Tong and Yuan, Yu-Jie and Fu, Hongbo and Lai, Yu-Kun and Zhang, Hao. SDM-NET: Deep Generative Network for Structured Deformable Mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 1

[7] David George, Xianghua Xie, and Gary KL Tam. 3D Mesh Segmentation via Multi-Branch 1D Convolutional Neural Networks. *Graphical Models*, 96:1–10, 2018. 1

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. 1

[9] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C. Russell, and Mathieu Aubry. 3D-CODED: 3D Correspondences by Deep Deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 5

[10] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. AdvPC: Transferable Adversarial Perturbations on 3D Point Clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 241–257, 2020. 1, 2

[11] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3D Adversarial Point Clouds. In *CVPR*, pages 15335–15344, 2022. 7, 14

[12] Itai Lang, Uriel Kotlicki, and Shai Avidan. Geometric Adversarial Attacks and Defenses on 3D Point Clouds. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 1196–1205, 2021. 2, 6, 7, 12, 18, 24, 26

[13] Bruno Lévy. Laplace-Beltrami Eigenfunctions Towards an Algorithm That Understands Geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, 2006. 2, 3

[14] Giorgio Mariani, Luca Cosmo, Alexander M Bronstein, and Emanuele Rodola. Generating Adversarial Surfaces via Band-Limited Perturbations. In *Computer Graphics Forum*, volume 39, pages 253–264. Wiley Online Library, 2020. 1, 2, 5, 7

[15] Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodola, Maks Ovsjanikov, and Simone Melzi. Instant Recovery of Shape From Spectrum Via Latent Space Connections. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 120–129, 2020. 2, 5, 14, 17

[16] Mehr, Eloi and Jourdan, Ariane and Thome, Nicolas and Cord, Matthieu and Guitteny, Vincent. DiscoNet: Shapes Learning on Disconnected Manifolds for 3D Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3474–3483, 2019. 1

[17] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003. 2, 3

[18] Hanieh Naderi, Kimia Noorbakhsh, Arian Etemadi, and Shohreh Kasaei. LPF-Defense: 3D Adversarial Defense Based on Frequency Analysis. *Plos one*, 18(2):e0271388, 2023. 8

[19] Nousias, Stavros and Arvanitis, Gerasimos and Lalos, Aris S and Moustakas, Konstantinos. Fast Mesh Denoising with Data Driven Normal Filtering using Deep Variational Autoencoders. *IEEE Transactions on Industrial Informatics*, 17(2):980–990, 2020. 1

[20] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387, 2016. 1

[21] Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 2, 3

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 652–660, 2017. 6, 7, 18

[23] Arianna Rampini, Franco Pestarini, Luca Cosmo, Simone Melzi, and Emanuele Rodola. Universal Spectral Adversarial Attacks for Deformable Shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3216–3226, 2021. 1, 2, 5, 6, 7, 8

[24] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D Faces Using Convolutional Mesh Autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 725–741, 2018. 2, 5, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24

[25] Olga Sorkine. Laplacian Mesh Processing. *Eurographics (State of the Art Reports)*, 4, 2005. 4

[26] Gabriel Taubin. A signal Processing Approach to Fair Surface Design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995. 2

[27] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008. 12

[28] Yuxin Wen, Jiehong Lin, Ke Chen, C. L. Philip Chen, and Kui Jia. Geometry-Aware Generation of Adversarial Point Clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[29] Chong Xiang, Charles R Qi, and Bo Li. Generating 3D Adversarial Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern)Recognition (CVPR)*, pages 9136–9144, 2019. 1, 2

[30] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial Meshes for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019. 1, 7

[31] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully Convolutional

Mesh Autoencoder Using Efficient Spatially Varying Kernels. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9251–9262, 2020. 2

[32] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6365–6373, 2017. 2, 5, 7, 9, 15, 16, 18, 19, 21, 22, 23, 26