DOI:10.1111/cgf.14083
Eurographics Symposium on Geometry Processing 2020
Q. Huang and A. Jacobson
(Guest Editors)

Volume 39 (2020), Number 5

Generating Adversarial Surfaces via Band-Limited Perturbations

G. Mariani[1], L. Cosmo[1,2], A. M. Bronstein[3] and E. Rodolà[1]

[1]Sapienza University of Rome
[2]University of Lugano
[3]Technion - Israel Institute of Technology

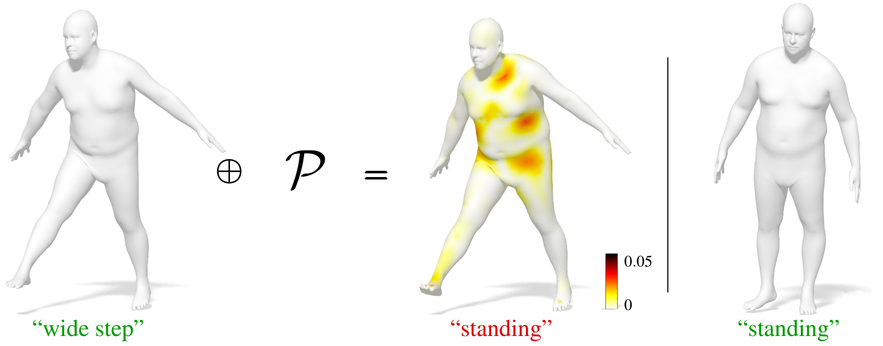

Figure 1: *An example of targeted adversarial attack to a 3D pose classifier. The input shape (leftmost), correctly classified in the "wide step" pose, is minimally perturbed so as to induce a mis-classification toward the target "standing" pose (rightmost, green label). Despite the natural-looking deformation, our approach does not use any parametric model for the input shape, but rather optimizes directly for a* smooth *perturbation of the 3D vertex coordinates. The heatmap encodes curvature distortion, growing from white to dark red.*

Abstract

Adversarial attacks have demonstrated remarkable efficacy in altering the output of a learning model by applying a minimal perturbation to the input data. While increasing attention has been placed on the image domain, however, the study of adversarial perturbations for geometric data has been notably lagging behind. In this paper, we show that effective adversarial attacks can be concocted for surfaces embedded in 3D, under weak smoothness assumptions on the perceptibility of the attack. We address the case of deformable 3D shapes in particular, and introduce a general model that is not tailored to any specific surface representation, nor does it assume access to a parametric description of the 3D object. In this context, we consider targeted and untargeted variants of the attack, demonstrating compelling results in either case. We further show how discovering adversarial examples, and then using them for adversarial training, leads to an increase in both robustness and accuracy. Our findings are confirmed empirically over multiple datasets spanning different semantic classes and deformations.

CCS Concepts

• ***Computing methodologies*** → ***Adversarial learning; Shape analysis;***

1. Introduction

In many applicative areas, accounting for the presence of malicious adversaries has become a prominent focus of research. In these contexts, the primary interest is to expose the inherent flaws of a given machine learning system, and therefore to design appropriate defense mechanisms that make the system robust to different types of attack. These attacks take the form of carefully perturbed data (*adversarial examples*) that are meant to induce an alteration of the output predicted by the machine learning model.

In computer vision, adversarial attacks are modeled as imperceptible pixel noise applied on the image domain, crafted in a way to fool image classifiers. Due to their high potential impact on critical vision-based systems (e.g. autonomous driving), their study has given rise to a thriving literature in recent years. However, much less attention has been devoted to the case of geometric data, where the variability in the representation and the non-Euclidean nature of the data itself pose additional hurdles that are not present in the flat and regular realm of images.

In this paper, we consider adversarial attacks on deep learning models operating with surfaces embedded in $\mathbb{R}^3$. Our method revolves around the question of what constitutes a *perceptible* perturbation in the case of surfaces. We show that naïvely mimicking image and point cloud-based attacks via additive noise on vertex coordinates leads to evident artifacts, and advocate the adoption of subspace parametrization to induce *smooth* perturbations on the 3D embeddings. Remarkably, the resulting adversarial examples exhibit semantically localized behavior without having access neither to local part information, nor to a parametric shape model at any stage of the generation process. Furthermore, by formulating the perturbation via the manipulation of a vector field on a geometric domain, our approach does not rely on a specific surface discretization (thus admitting both triangle meshes and point clouds), as long as the latter admits the construction of a Laplacian operator. Finally, similarly to the classical Euclidean setting, we demonstrate that injecting our adversarial examples into the training data can lead to strong improvements in terms of increased robustness of the targeted classifier across multiple datasets and settings.

1.1. Related work

Since the initial discovery of this phenomenon in [SZS*13], increasingly stronger defenses [GSS14, MMS*18, XWM*19, SGI19, KHM19, ZL19, HRF19, ZYJ*19] and counterattacks [GSS14, CW17, ACW18, MMS*18, RHO*19, PMG*17, CZS*17, LLW*19] were proposed in the literature. Adversarial attacks have also been shown to occur in tasks beyond image classification where they were first discovered: in real-life object recognition [BMR*17, XZL*19, AEIK18], object detection [WLW*19], natural language processing [GLSQ18, CKG19, JJZS19], reinforcement learning [GDK*19], speech-to-text [CW18], and point cloud classification [XQL19], just to mention a few. Moreover, the adversarial attacks can be used to improve the performance of the deep neural networks on unperturbed data [XTG*19, GRYL20, SWC*20]. Understanding the root cause of adversarial examples, how they are created and how we can detect and prevent such attacks, is at the center of many research works. [GMF*18] argued that adversarial examples are an inevitable property of high-dimensional data manifolds rather than a weakness of specific models. In view of this, the true goal of an adversarial defense is not to get rid of adversarial examples, but rather to make their search hard.

Current defense methods are based on either implicit or explicit regularization. Explicit regularization methods aim to increase the performance under adversarial attack by directly incorporating a suitable term into the loss of the network during training, usually by incorporating adversarial examples for the dataset used in the training process. In contrast, implicit regularization methods that do not change the objective, such as variational dropout [KSW15], seek to train the network to be robust against any perturbations without taking into account adversarial examples. In particular, adding randomness to the network can be especially successful [LCZH18, BMCM18, HRF19], since information acquired from previous runs cannot be directly applied to a current run. Another way to make use of randomness to improve classifier robustness is randomized smoothing [CRK19]: averaging of the outputs of the classifier over some random distribution centered in the input data point. The effects of these three approaches (explicit regularization, implicit regularization, and smoothing) do not necessarily line up with or contradict each other. Thus, one could use a combination of the three, when devising adversarial defenses.

In stark contrast to such a rich production, the literature on adversarial attacks for geometric (more in general, non-Euclidean) data is relatively scarce. Part of the reason lies in the fact that deep learning for structured data has only recently seen renovated interested from the community; we mention here seminal, although quite recent works on graphs [KW16], meshes [MBBV15], and point clouds [QSMG17]. Adversarial learning for such data is, in turn, much less developed and concentrated in the last 2-3 years. In the following, we cover the relevant literature.

Graphs. In the case of graph data, adversarial attacks are becoming more and more relevant due to their applicability in a range of tasks including community detection [CNK*17], fact plausibility prediction [ZZG*19], link prediction [STL*18], graph [DLT*18] and node classification [ZAG18] among others. These adversarial models typically attack the graph *topology* by adding, removing, or rewiring edge connections among the graph nodes. When per-node features are part of the data, the attack can also be phrased as a perturbation of these features [ZAG18]. We refer to the recent surveys [SWYL18, XML*19] for a more in-depth look at this family of techniques.

In this work, we focus on a different setting. Instead of considering topological modifications to the discrete structure representing the 3D shape (e.g., a triangle mesh), we attack the *embedding* itself independently of its specific representation.

Point clouds and meshes. More recently, adversarial attacks have been demonstrated for irregular point cloud data on tasks of rigid 3D object classification. These attacks either move individual points by small shifts in 3D space [XQL19, LYS19, ZLSJ19, HRTG19, WLCJ19], or add outlier points to the cloud so as to confuse the classifier [XQL19]. As we show in the sequel, shifting points by small amounts is not a viable option when dealing with generic surfaces, even more so when these represent deformable objects; while hidden to the human eye in a point cloud representation, vertex-wise perturbation becomes immediately evident when the object is rendered as a surface, thus defeating the inherent idea of the attack being imperceptible. In [WLCJ19] this problem is partly addressed via a regularization term on the mean curvature difference between the adversarial point cloud and the original one. In [ZWCL20] it is proposed to generate perturbations by applying a global rigid isometry to the 3D point cloud, but this only affects systems that are not orientation-invariant. The work of [XYL*19] seems to be the only one, to date, to consider mesh data. It employs a differentiable renderer, together with a perceptual loss in the image domain, to generate attacks on photorealistic renderings by minimally perturbing the shape texture and geometry.

Similarly to the graph-based setting, adversarial learning on point clouds can be seen as an attack to the *representation* rather than to the underlying surface, since shifting, adding or removing points are operations that modify the local neighborhood relations. This consideration also begs the question as to what makes an attack "legitimate". While fake users or fake product reviews can be seen as realistic counterparts of graph-based adversarial models [SWT*20], in the case of 3D data the per-vertex perturbation of a point cloud might be provoked by tampering with the depth sensor, e.g., by malicious miscalibration or by shooting lasers [CXC*19], which is more difficult to realize in practice.

We position our method in a setting that is closer to what is done with images, where the discretization of the domain is given and is left untouched by the perturbation. In our scenario, the adversarial example is a minimally deformed version of the original shape, such as a slight change in pose or style. But unlike images, we do not assume any signal to be given on the surface.

1.2. Contribution

With this paper we introduce and analyze a new family of adversarial attacks for 3D surfaces. Our motivations are rooted in the fact that deep learning is gaining a growing presence as a major instrument in graphics and geometry processing. This requires studying the susceptibility and robustness of such learning models from the point of view of adversarial learning, and in turn, leverage on these results for the design of better models.

Our main contributions can be summarized as follows:

- To the best of our knowledge, ours is the first attempt at addressing adversarial learning for deformable 3D shapes.
- We introduce the notion of *band-limited perturbations*, which can be applied across different shape discretizations and are not tailored to a specific choice.
- We address both the targeted and untargeted cases, and further employ our attacks for the purpose of adversarial training.

We test our methodology on a selection of different datasets encompassing multiple semantic classes of organic shapes, demonstrating consistent behavior.

2. Background

We model our 3D shapes as 2-Riemannian manifolds $\mathcal{X}$ embedded in $\mathbb{R}^3$, possibly with boundary $\partial\mathcal{X}$. We denote by $\mathcal{F}(\mathcal{X})$ a Sobolev space of real-valued functions on $\mathcal{X}$, and use the inner product $\langle f,g\rangle = \int_{\mathcal{X}} f(x)g(x)\mathrm{d}x$, where $\mathrm{d}x$ is the standard volume form. To each shape $\mathcal{X}$ we attach the positive semi-definite Laplace-Beltrami operator $\Delta : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{X})$, which admits the spectral decomposition:

$$\Delta\phi_i(x) = \lambda_i\phi_i(x) \qquad x \in \mathrm{int}(\mathcal{X}) \qquad (1)$$

$$\langle \nabla\phi_i(x), \vec{n}(x)\rangle = 0 \qquad x \in \partial\mathcal{X} \qquad (2)$$

into eigenvalues $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots$, assumed to be ordered non-decreasingly, and the associated eigenfunctions $\phi_1, \phi_2, \phi_3, \ldots,$ which form an orthogonal basis for $\mathcal{F}(\mathcal{X})$. We adopt homogeneous Neumann boundary conditions (2), where $\vec{n}$ denotes the unit vector normal to the boundary.

2.1. Smoothness

The canonical ordering of the eigenvalues makes it so that truncating the Fourier-like series expansion of any scalar function $f \in \mathcal{F}(\mathcal{X})$ to the first k terms:

$$f(x) \approx \sum_{i=1}^{k} \langle \phi_i, f\rangle \phi_i(x)\,, \qquad (3)$$

yields a *band-limited* approximation of f with bandwidth k. In fact, the orthogonal basis $\{\phi_i\}$ is optimal for approximating functions with bounded gradient magnitude in the L^2 sense, as described in the following:

Theorem 1 [ABK15] For any given choice of $k \geq 1$ and any function $f \in \mathcal{F}(\mathcal{X})$, the inequality:

$$\|f - \sum_{i=1}^{k} \langle \psi_i, f\rangle \psi_i\|^2 \leq \alpha \frac{\|\nabla f\|^2}{\lambda_{k+1}} \qquad (4)$$

holds for $\alpha = 1$ whenever one chooses ψ_i to be the Laplacian eigenfunctions, while tightening the bound with $0 \leq \alpha < 1$ is not possible for *any* sequence of orthogonal functions $\{\psi_i \in \mathcal{F}(\mathcal{X})\}$.

In the inequality (4), the term $\|\nabla f\|^2 = \int_{\mathcal{X}} \|\nabla f(x)\|^2 \mathrm{d}x$ corresponds to the Dirichlet energy of f, which provides a measure of smoothness for the function f. Thus, according to the theorem, the approximation error of f is bounded by its smoothness. Smooth functions (for which $\|\nabla f\|^2$ is small) are well represented by the band-limited approximation. Further, by increasing the bandwidth k, and thus the denominator in $\frac{\|\nabla f\|^2}{\lambda_{k+1}}$, the error decreases.

2.2. Discretization

In the discrete setting, 3D shapes are sampled at n points $x_1 \ldots x_n$ and approximated by a triangle mesh with vertex positions $\mathbf{X} \in \mathbb{R}^{n\times 3}$, and where each edge $e_{ij} \in E$ belongs to at most two triangle faces T_{ijk} and T_{jih}. Scalar functions f are discretized as vectors $\mathbf{f} \in \mathbb{R}^n$ with the values $f(x_i)$ for $i = 1 \ldots n$, and linearly interpolated within each triangle. Inner products $\langle f,g\rangle$ are discretized as $\mathbf{f}^\top \mathbf{A}\mathbf{g}$, where $\mathbf{A}$ is a $n \times n$ diagonal matrix of local area elements $a_i = \frac{1}{3}\sum_{jk:ijk\in T} A_{ijk}$ (A_{ijk} is the area of triangle T_{ijk}). Vector fields $V : \mathcal{X} \to \mathbb{R}^3$ are discretized as matrices $\mathbf{V} \in \mathbb{R}^{n\times 3}$, and their integration $(\int_{\mathcal{X}} \|V(x)\|_2^2 \mathrm{d}x)^{1/2}$ is discretized as $\|\mathbf{V}\| = \sqrt{\mathrm{tr}(\mathbf{A}\mathbf{V}\mathbf{V}^\top)}$.

Following linear FEM discretization, the Laplacian Δ is defined in terms of $\mathbf{A}$ and of a symmetric matrix $\mathbf{W}$ of edge weights:

$$w_{ij} = \begin{cases} -(\cot\alpha_{ij} + \cot\beta_{ij})/2 & e_{ij} \in E; \\ \sum_{k\neq i} w_{ik} & i = j \end{cases} \qquad (5)$$

where α_{ij}, β_{ij} are the opposite angles to edge e_{ij}. A generalized eigenproblem $\mathbf{W\Phi} = \mathbf{A\Phi}\mathrm{diag}(\boldsymbol{\lambda})$ is solved for computing the Laplacian eigenvalues (stored in vector $\boldsymbol{\lambda} \in \mathbb{R}^k$) and eigenvectors (stored column by column in the matrix $\mathbf{\Phi} \in \mathbb{R}^{n\times k}$). In case the given shapes have low resolution, a higher-order FEM discretization of the Laplacian (see, e.g., [Reu10, Sec. 4.1]) can be computed while leaving the rest of our pipeline intact.

For polygon meshes, a discretization can also be easily computed, e.g., by following [BHKB20]. Similarly, for point clouds one can adopt the simple approach of [CRT04], where connectivity is established on the fly at each point according to a local Delaunay triangulation, and weights are locally computed as in Eq. (5) or by a higher-order counterpart.

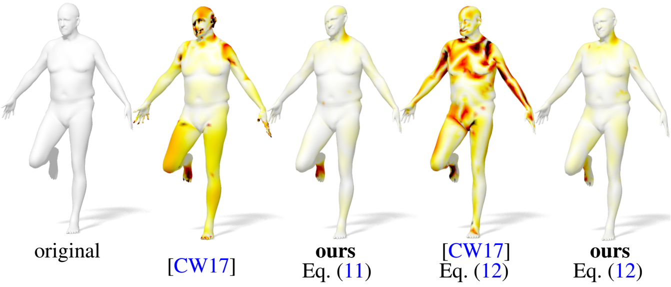

Figure 2: *Left to right: original shape; adversarial example obtained by [CW17], which is equivalent to Eq. (11) with $k = n$; our band-limited attack with point-wise distortion and $k = 40$; the method of [CW17] with the pair-wise energy of Eq. (12); and our band-limited attack with the pair-wise energy and $k = 40$.*

3. Adversarial surfaces

Our attacks are based on assuming partial knowledge of the learning model (*white-box* attack). Specifically, we require access to the model's loss and parameters. This is a widely adopted assumption; it has been noted [CW17] that moving to a *black-box* attack could be done by training a substitute model with black-box access to the target model, and then attacking the substitute model [PMG16].

We use pose and style classification of deformable 3D shapes as our primary evaluation domain. Throughout the following sections, in our qualitative plots we visualize the per-point absolute distortion of mean curvature between the original shape and the adversarial shape, encoded as a heatmap growing from white to dark red.

3.1. Setting & objective

Our target model is a deep m-class classifier $F_\theta : \mathbb{R}^{n\times 3} \to [0,1]^m$ which, given a 3D shape with n points as input, outputs a discrete probability distribution over m classes. The classifier is a deep neural network parametrized by θ. The network parameters are *fixed* and given, since they are the result of training the classifier; for this reason and to avoid confusion, we will omit them in what follows.

While the specific network architecture is not relevant for the attack, we assume the output layer to be a softmax, ensuring $\sum_i F(\mathbf{X})_i = 1$ and $F(\mathbf{X})_i \geq 0$. The inputs to the softmax layer are called *logits* and are denoted by $\mathbf{Z}_i$, where i ranges over the m classes. The classifier assigns the label $C(\mathbf{X}) = \arg\max_i F(\mathbf{X})_i$ to the input shape $\mathbf{X}$.

Objective. Denoting by $C^*(\mathbf{X})$ the ground-truth label of $\mathbf{X}$, our aim is to generate a new *adversarial* shape $\mathbf{X}'$ such that:

$$C(\mathbf{X}') \neq C^*(\mathbf{X}) \quad \text{and} \quad \mathbf{X}' \sim \mathbf{X}\,, \tag{6}$$

where $\sim$ signifies that $\mathbf{X}'$ is imperceptibly close to $\mathbf{X}$ according to some metric, which we discuss below.

For both the classifier and the adversarial generator, we only assume access to the raw geometric data represented as a set of (x,y,z) coordinates, possibly with connectivity information. During the attack we allow no editing operations on the discrete structure, i.e., vertices and edges can not be added, switched, or removed.

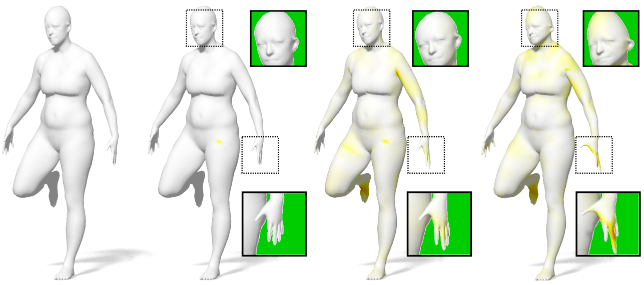

Figure 3: *Increasing the spectral bandwidth k of the perturbation in a* targeted *attack. For $k = 10$ the original shape is* not *misclassified. As k increases, the shape gets misclassified but the perturbation becomes more noticeable, manifesting a distorted head and elongated fingers; the green background in the insets is for better contrast.*

Choice of a metric. The success of the attack depends on how one measures the similarity $\mathbf{X}' \sim \mathbf{X}$ between the original shape and the adversarial shape. A typical choice consists in minimizing the L^p distance between points (pixel values in the case of images), in our case, between the original vertex positions and the perturbed positions. The choice of p is often driven empirically. To better account for the continuous nature of the underlying surface, as we will show in Section 3.3, we propose to compute similarity by comparing local *neighborhoods* instead of individual vertices.

3.2. Band-limited perturbations

We model the adversarial shape $\mathbf{X}'$ as a perturbation of $\mathbf{X}$ along a deformation field $\mathbf{V} \in \mathbb{R}^{n\times 3}$:

$$\mathbf{X}' = \mathbf{X} + \mathbf{V}\,. \tag{7}$$

In this paper, we advocate that the vector field $\mathbf{V}$ should be *smooth* in addition to having small norm. Smooth deformations preserve local neighborhoods, and prevent the formation of adversarial jittering that is observed with point cloud attacks; see Figures 2, 4 and 10 for examples.

Smoothness on $\mathbf{V}$ is enforced by appealing to Theorem 1, namely by passing to a subspace parametrization:

$$\mathbf{V} = \mathbf{\Phi}\mathbf{v}\,, \tag{8}$$

where $\mathbf{\Phi}$ contains the first k Laplacian eigenvectors of $\mathbf{X}$, and $\mathbf{v} \in \mathbb{R}^{k\times 3}$ is a set of expansion coefficients representing $\mathbf{V}$ in the reduced basis. With this parametrization, smoothness is easily controlled by varying the spectral bandwidth k, as illustrated in Figure 3. For large k, one admits high-frequency oscillations in the deformation field, while for small k we only retain the smoother, low-frequency behavior.

Remark. With this subspace parametrization, the high-frequency jittering that can be seen in Figure 2 can not even be *represented*, unless a very large bandwidth k is chosen.

We emphasize that we require smoothness for the deformation field $\mathbf{V}$ only, and *not* for the entire embedding $\mathbf{X}'$, which would instead lead to an undesirable loss of geometric detail on the surface.

We inject band-limited perturbations in two different settings, differing by the specificity of the attack.

3.3. Targeted attack

In the *targeted* scenario, the attacker prescribes a target class t towards which to steer the classifier. The adversarial shape $\mathbf{X}'$ is then generated so as to satisfy:

$$C(\mathbf{X}') = t\,. \tag{9}$$

Due to the difficulty of imposing this constraint, we follow the general approach of Carlini and Wagner [CW17], which requires $\mathbf{X}'$ to minimize the penalty function:

$$h_t(\mathbf{X}') = \max\{\mathbf{Z}'_i : i \neq t\} - \mathbf{Z}'_t\,, \tag{10}$$

In particular, $h_t(\mathbf{X}') < 0$ if and only if the constraint of Eq. (9) holds exactly. The intuition behind Eq. (10) is that minimizing $h_t(\mathbf{X}')$ with respect to $\mathbf{X}'$ induces a concentration of mass around the target class t, making t the most likely label.

Problem 1. Minimizing Eq. (10) alone would lead to $\mathbf{X}'$ deforming arbitrarily. Therefore, we pass to the unconstrained minimization problem:

$$\min_{\mathbf{v}\in\mathbb{R}^k} \|\mathbf{\Phi v}\| + c\,h_t(\mathbf{X}+\mathbf{\Phi v})^+\,, \tag{11}$$

where $a^+ = \max\{0, a\}$. The term $c \geq 0$ balances the perturbation strength (encoded as the 2-norm of the deformation field) with the misclassification penalty. If $c = 0$, no misclassification is obtained and the shape remains unperturbed, $\mathbf{v} = \mathbf{0}$. Otherwise, the solution will put the least possible amount of mass around t needed to cause misclassification. To ensure this, we select a value for c via exponential search as the smallest value for which the resulting solution $\mathbf{v}^*$ implies $h_t(\mathbf{X}+\mathbf{\Phi v}^*) < 0$.

To summarize, problem (11) seeks for a band-limited perturbation with small norm, encoded in the expansion coefficients $\mathbf{v} \in \mathbb{R}^k$, that gives rise to a mislabeling towards a target class t. This attack does not directly use the output $F(\cdot)$ of the classifier, but instead operates one layer behind, at the logit level. Empirically, this choice was shown to lead to better results [CW17].

Similarity measure. In practice, we replace the point-wise measure $\|\mathbf{\Phi v}\|$ appearing in Eq. (11) with the pair-wise distortion:

$$\sum_{i=1}^{n} \sum_{j\in \mathrm{NN}(i)} \left(\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\| - \|\mathbf{X}'_{i:} - \mathbf{X}'_{j:}\|\right)^2\,, \tag{12}$$

where $\mathbf{X}' = \mathbf{X} + \mathbf{\Phi v}$ and $\mathbf{X}_{i:}$ denotes the i-th row of matrix $\mathbf{X}$; the band-limited term now appears in the definition of $\mathbf{X}'$. As mentioned in Section 3.1, this corresponds to comparing local neighborhoods; in particular, it promotes local Euclidean distances to be preserved in an as-rigid-as-possible fashion. The neighbors are computed as those lying in the 3-hop neighborhood in the case of meshes, and as the 1% of nearest points in the case of point clouds. In Figure 2 we compare between using the point-wise distortion of Eq. (11) and the pair-wise distortion of Eq. (12).

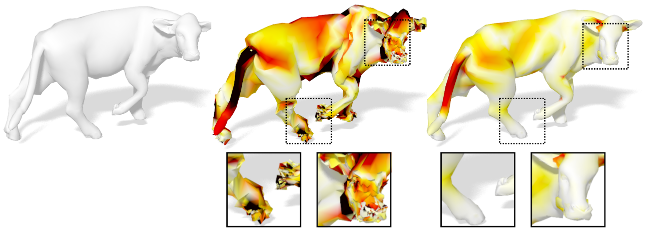

Figure 4: Untargeted *scenario. Comparison between the fast gradient sign method of [KGB16] (middle) and our band-limited attack (right) on the SMAL shape shown on the left. In both cases, the cow is misclassified as a lion. However, similarly to the targeted setting, the pointwise displacements of Eq. (13) give rise to noticeable perturbations, as highlighted in the insets.*

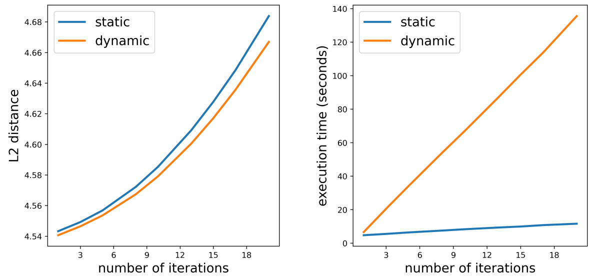

Figure 5: Untargeted *scenario. Comparison between our attack where* $\mathbf{A}$ *and* $\mathbf{\Phi}$ *are held fixed* (static) *or are re-computed at each iteration* (dynamic)*, in terms of execution time and* L_2 *distance* $\|\mathbf{X}-\mathbf{X}'\|$. *These results suggest that for time critical applications a static approach is to be preferred.*

3.4. Untargeted attack

In the *untargeted* scenario, instead of generating an adversarial shape $\mathbf{X}'$ for some given target class, we ensure that Eq. (6) is satisfied regardless of the specific label $C(\mathbf{X}')$.

Let $L(\mathbf{X}, y) = -\log p(y|\mathbf{X})$ be the cross-entropy loss between the probability output of the classifier and the ground-truth class y. An adversarial shape $\mathbf{X}'$ is generated by following the iterative fast gradient sign method [KGB16], consisting in applying the iterates:

$$\mathbf{X}'^{(i)} = \Pi_{\mathbf{X},\varepsilon}\left(\mathbf{X}'^{(i-1)} + \alpha\,\mathrm{sign}(\nabla L(\mathbf{X}'^{(i-1)}, y))\right)\,, \tag{13}$$

with $\mathbf{X}'^{(0)} = \mathbf{X}$. At each iteration, Eq. (13) takes a step of length α for each dimension of the gradient of the classification loss, and then projects back the perturbation to within an ε-radius from the original surface via the vertex-wise projection $\Pi_{\mathbf{X},\varepsilon}(\cdot)$. As the iterations proceed, the shape $\mathbf{X}'^{(i)}$ is updated so as to *increase* the loss, and thus induce a misclassification.

Problem 2. Applying this method to a given surface will not lead to a smooth deformation in general. Band-limited perturbations are

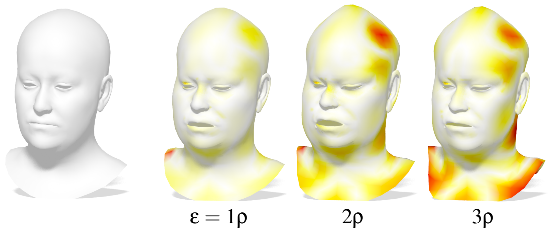

Figure 6: *Untargeted adversarial examples generated with increasing displacement threshold ε, where ρ is the median edge length. Smaller ε leads to less evident perturbations.*

injected by splitting the iterations as follows:

$$\mathbf{V}^{(i)} = \mathbf{V}^{(i-1)} + \alpha\,\mathrm{sign}(\nabla L(\mathbf{X}'^{(i-1)}, y)) \tag{14}$$

$$\mathbf{V}^{(i)} = \Pi_{\mathbf{0},\varepsilon}\left(\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A}\mathbf{V}^{(i)}\right) \tag{15}$$

$$\mathbf{X}'^{(i)} = \mathbf{X} + \mathbf{V}^{(i)} \tag{16}$$

where $\mathbf{V}^{(0)}$ is a zero vector field, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the matrix of area elements, and $\mathbf{\Phi} \in \mathbb{R}^{n\times k}$ contains the first k eigenfunctions of the initial shape $\mathbf{X}$. We do not update $\mathbf{A}$ and $\mathbf{\Phi}$ across the iterations; as shown in Figure 5, updating them would cause a large increase in runtime with no significant benefit. For $k = n$, it is easy to prove that Eqs. (14)–(16) are equivalent to Eq. (13), since $\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{A} = \mathbf{Id}$ and the rest follows by induction. However, for $k < n$ we obtain a band-limited representation of the perturbation $\mathbf{V}$ (computed at Eq. (15)), seen as a displacement field over the surface. In Figure 4 we show an example of adversarial surfaces obtained with and without the band-limited regularizer.

In our tests, we also experimented with a different perturbation model in which we take gradient steps with respect to the spectral coefficients of $\mathbf{V}$, rather than w.r.t. the vertices $\mathbf{X}'$ as done in Eq. (14). However, doing so did not yield good results. This is because the spectral coefficients do not contribute equally to the spatial deformation due to the canonical ordering of the frequencies, while the gradient sign method attributes equal weight to each dimension in the representation of the perturbation.

Parameters. In Eqs. (14)–(16) we choose $\alpha, \varepsilon > 0$ to be sufficiently small, specifically we set $\alpha = 0.3\rho$ and $\varepsilon = 3\rho$, where ρ is the median edge length of $\mathbf{X}$. Compared to the targeted scenario, this provides us with more fine-grained control on the desired amount of deformation for the adversarial shape, as illustrated in Figure 6. In fact, in the former case, one can control the amount of deformation only indirectly, by tuning the trade-off parameter c in Eq. (11).

3.5. Relation to existing methods

Our approach bears some similarity with other recent works operating on point clouds, although with some important differences. The work of [XQL19] generates adversarial examples by either shifting or adding individual points; the shifting operation is obtained by applying the plain method of [CW17] on the point cloud vertices, while the addition of points uses a variant tailored for the task. In both cases, no precaution is taken to preserve smoothness on the underlying surface, therefore producing adversarial examples with substantial high-frequency noise. In [LYS19, HRTG19] the authors adopt the iterative fast gradient sign method, leading again to high-frequency jittering. This is partially addressed in [LYS19] by projecting the perturbation on a given triangular mesh, which in turn limits the effectiveness of the attack. Concurrently to our work, [TYHJ20] introduced a method for generating adversarial examples for the PointNet++ classifier. The authors modify the loss of [CW17] by adding a smooth penalty term on the adversarial point cloud. This penalty needs to be properly tuned as it often leads to over-smoothing and does not preserve sharp edges on the original surface. We prevent this by enforcing smoothness on the *perturbation* via our band-limited prior. Finally, in [WLCJ19], preservation of geometric detail is achieved by adding a regularization term on the point cloud curvature distortion, leading to effective perturbations. We show a comparison with this latter method in Figure 11.

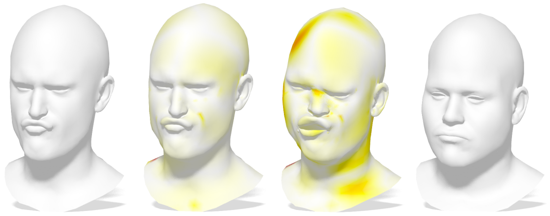

Figure 7: *Example of targeted attack before and after adversarial training. The band-limited attack yields a negligible deformation before the adversarial training (second column), and becomes much more noticeable after (third column).*

Dataset	success rate	L_2–norm	mean-curv. dist.
CoMA	94%	8.47e-3	3.30
SMAL	100%	3.6e-2	2.51
FAUST	100%	6e-2	3.05
CoMA (adv.)	**80%**	1.4e-2	4.36
SMAL (adv.)	**87.5%**	7.7e-2	6.21
FAUST (adv.)	**96.6%**	4.5e-2	3.18

Table 1: *Success rates of our* targeted *adversarial attacks, L^2-norm, and average mean-curvature distortion before (first three rows) and after (last three rows) adversarial training. The drop in success rate is desired and signifies an improvement in robustness for the attacked classifiers.*

4. Adversarial training

In this Section we show how our adversarial examples can be employed to improve the robustness of the learning model under attack. To this end, we follow the general approach of [MMS*18]. Specifically, let us be given a training pair $(\mathbf{X}, y)$, where y is the true label associated to shape $\mathbf{X}$. For the given pair, we generate

Dataset	Normal Training	Adversarial Training
CoMA	96.6%	**99.3%**
SMAL	98.2%	**99.5%**
FAUST	90.1%	**94.2%**

Table 2: *Classifier accuracy before (first column) and after (second column) training with our adversarial examples over three different datasets.*

an adversarial example $(\mathbf{X}', z)$ and construct a new training pair $(\mathbf{X}', y)$, that we use to enrich the training dataset.

In Figure 7 we show a qualitative example, in the targeted setting, of how the learning model increases its robustness after training on our band-limited adversarial examples. In Table 1 we report the drop in success rate of our adversarial attacks, *after* re-training the classifiers with our band-limited adversarial examples. Adversarial training leads to an increase in robustness, as previously suggested in Figure 7. Likewise, in Table 2 we report the accuracy improvement of the classifiers before and after adversarial training.

5. Implementation details

We implemented our method in PyTorch and executed our code on a Titan Xp GPU. In the targeted scenario, the optimization problem for the attack of Eq. (11) was solved with the ADAM solver [KB14] with learning rate 10^{-4}. While working with the similarity measure of Eq. (12), we found it occasionally tended to drift the shape rigidly in space, thus causing misclassification due to the lack of translation invariance in the classifier. To rule out this type of "accidental" attacks, we added a penalty term on the mesh centroid, forcing it to be close to the original position. For the untargeted scenario, the gradient appearing in Eq. (14) was computed via automatic differentiation in autograd.

Network architecture of the classifier. Our attack and defense mechanisms do not change depending on the classifier's architecture. However, to exclude possible misleading results owed to the design of a new classifier, we used well-established architectures from the geometry processing literature. In particular, we used two types of networks in our tests, depending on the kind of input. For **triangle meshes** we employed a deep classifier composed by three layers of ReLU-activated fast localized spectral filtering [DBV16] interleaved by mesh decimation via iterative edge collapse [GH97], and a final dense layer. This architecture is similar to the state-of-the-art encoder component used by the CoMA [RBSB18] autoencoder. For **point clouds**, we used the PointNet classifier [QSMG17], composed by two layers of point convolution followed by a max pooling operation and two final fully connected layers to obtain the classification.

Runtime. Our targeted and untargeted attacks exhibit different runtimes in practice. In particular, the optimization problem of Eq. (11) requires several hundred iterations to yield a good result; typically, more iterations are needed if the target class is very different from the initial shape. In the untargeted setting of Eqs. (14)-(16) we can find a good solution (within the prescribed bounds) in a few dozen iterations. For this reason, as also remarked elsewhere in the literature [CW17, KGB17], untargeted attacks are far more practical for adversarial training.

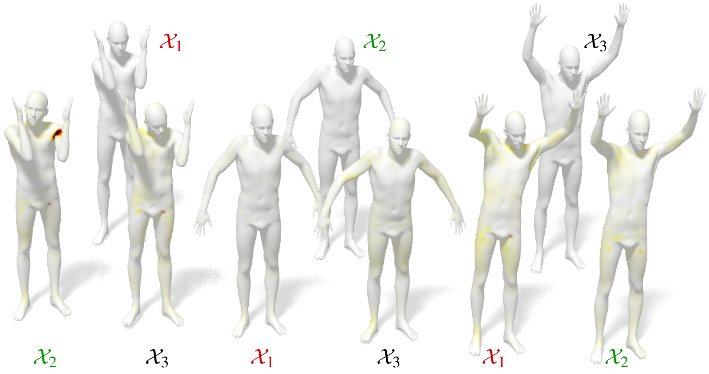

Figure 8: *For each shape $\mathcal{X}_i$ (top row), we conduct a targeted attack where we set the remaining two poses as targets, resulting in two adversarial examples per shape. In the bottom row, labels denote the target shapes toward which each adversarial example is misclassified. To a human observer, each of the three shape clusters clearly represents a specific pose; to the eyes of the attacked classifier, the shapes are clustered according to label color.*

6. Results

In this Section we present additional and more extensive experimental results of our band-limited attacks on different datasets.

Remark. Even though in our examples we often show the adversarial surfaces side-by-side with the original shape, we encourage the reader to evaluate each adversarial shape in isolation. Differently from the existing approaches, in our setting we deal with *deformable* surfaces. Therefore, even if a small deformation might be noticeable *when compared to a knowingly unperturbed reference*, it will hardly be recognized as a perturbation to a human observer who sees the shape in isolation, or who sees the two shapes without knowing which one is the reference. See Figure 8 for examples.

Data. We experimented with four datasets of organic shapes: CoMA [RBSB18] (composed of human faces taking different expressions), SMAL [ZKJB17] (four-legged animals in different poses), FAUST [BRLB14] and SHREC'14 [PSR*14] (full-body human subjects in different poses). On **FAUST**, we evaluated the task of pose classification; we used 8 of the available subjects for training, and then performed classification of the 10 poses over the remaining 2 subjects. **SHREC'14** was used for the task of pose classification on point clouds; it contains 400 scans of 40 different people in 10 different poses. We trained on 32 subjects and used the remaining 8 subjects (4 males, 4 females) for evaluation. **CoMA** is a dataset of human faces composed by sequences of 3D meshes of 13 subjects performing 13 different facial expressions. With this dataset our focus is on classifying the subject's identity; in this setting, the network was trained using a portion of the frames of each sequence, and tasked to recognize the identity of the subjects in the remaining frames. Following the train/test split

FAUST	untar	untar (ours)	tar	tar (ours)
success rate	**100%**	88.8%	63.8%	**100%**
L^2–norm	5.84e-2	**3.91e-2**	**4.31e-2**	6.20e-2
curv. dist.	28.03	**10.34**	21.6	**3.05**
CoMA	untar	untar (ours)	tar	tar (ours)
success rate	72%	**77%**	89%	**94%**
L^2–norm	8.72e-3	**8.58e-3**	**3.07e-3**	8.47e-3
curv. dist.	39.2	**4.82**	9.95	**3.30**
SMAL	untar	untar (ours)	tar	tar (ours)
success rate	**100%**	**100%**	**100%**	**100%**
L^2–norm	**5.84e-2**	7.73e-2	**1.52e-2**	3.59e-2
curv. dist.	23.36	**13.93**	5.05	**2.51**

Table 3: *Comparison between our band-limited method and the approaches of [KGB16] and [CW17] in the untargeted (**untar**) and targeted (**tar**) cases respectively. Bold numbers denote the best results. Curvature distortion gives a measure of noticeability of the attack.*

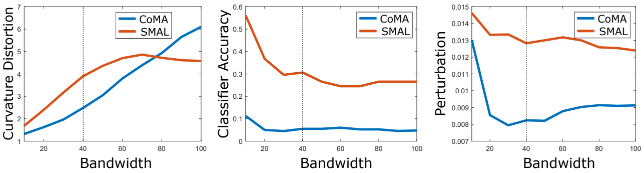

Figure 9: *Sensitivity analysis of our targeted adversarial attacks as a function of the bandwidth parameter k on the CoMA and SMAL datasets. The targets are randomly chosen at each run. The dashed line at* $k = 40$ *denotes the bandwidth chosen for all the other tests in this paper. See the main text for a detailed discussion.*

proposed in [RBSB18], the test set is composed by 10% of contiguous frames from each 3D sequence. Finally, the **SMAL** parametric model was used to generate 600 meshes of 5 types of animals: Felidae, Canidae, Equidae, Bovidae, and Hippopotamidae. The models were generated using the shape and pose parameters available at `http://smal.is.tue.mpg.de/downloads`. The classification task on this synthetic dataset is the categorization of the animal type. The training set is composed by 16 poses and 6 shapes for each animal family, summing up to 480 meshes. The test set was synthesized using 6 different shapes and 4 new poses for each family, amounting to 120 triangle meshes. All the datasets were further augmented by applying random rotations to each shape and normalized within each dataset to have unitary area.

Comparisons. We first report an extensive quantitative comparison between our band-limited attacks and the methods of [KGB16] (for the untargeted case) and [CW17] (for the targeted case), upon which many other approaches are based. We do this on the FAUST, CoMA and SMAL datasets. The comparisons are reported in Table 3 in terms of three error measures: *perturbation strength*, defined as the average L^2 distance between the adversarial shape and the corresponding vertices in the original shape; *curvature distortion*, defined as the average absolute difference between the mean curvature at those points; and the *success rate*, which counts how many adversarial attacks are successful on the test set.

Spectral bandwidth. Following the qualitative experiment of Figure 3, we carried out a more extensive evaluation of how the spectral bandwidth k affects the generated adversarial examples. The quantitative results on the entire CoMA and SMAL datasets are reported in Figure 9.

For these tests we analyze the *targeted* scenario, which is the most difficult setting for an adversarial attack. As expected, increasing the bandwidth k leads to an increase in curvature distortion (Figure 9, left column), since more high-frequency deformations are admitted by the representation. In turn, this leads to a decrease in accuracy for the classifier (middle column), meaning that the attacks are more successful thanks to the higher deformation budget, but also more noticeable. Finally, the perturbation strength decreases with k (right column). This is also expected; to induce misclassification, high-frequency (i.e. jittered) perturbations tend to be *sparse* as they move fewer points than a low-frequency (i.e. smooth) one, despite the latter being less noticeable.

Based on these results, we chose a value of $k = 40$ in all our tests.

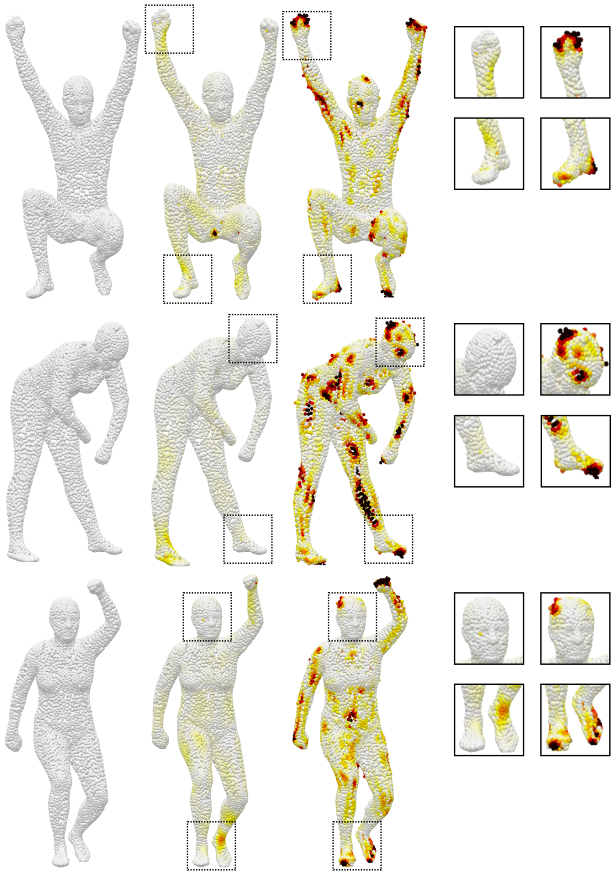

Figure 10: *Comparison between our band-limited adversarial examples (middle column) and the point-wise adversarial examples of [CW17] (right column) in the* targeted *scenario on SHREC'14 data. Both attacks lead to a successful misclassification, but in the latter case the perturbation is sharper and more evident, since points are shifted in a sparse manner.*

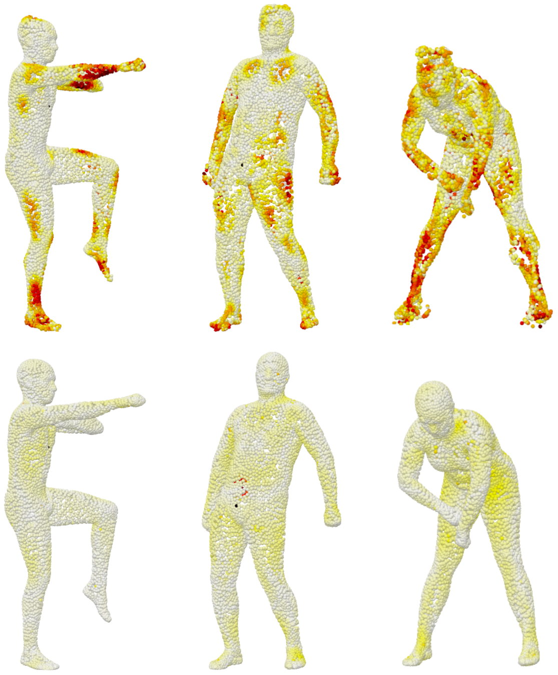

Figure 11: *Comparison between our approach and [WLCJ19]. On the bottom are shown the adversarial examples obtained using our approach, on the top the one obtained using [WLCJ19]. While their approach resulted in improved adversarial examples with respect of [CW17], they still struggle to preserve the structure of finer details: hands, nose, toes, etc. On the other hand, our approach, by constraining the perturbation in the truncated spectral domain (thus removing high frequency noise), can more easily preserve the structure of the surface.*

Point clouds. We ran our adversarial attacks on point cloud data from the SHREC'14 dataset, which consists of real-world shapes. When dealing with point clouds, our method changes in how neighbors are computed, but the overall approach remains the same. We show some qualitative results in Figure 10, in comparison with the pure point-based approach of [CW17]. A jittering effect is observed for the latter method, while our adversarial perturbations remain smooth. As already remarked, our band-limited attacks inherently avoid jittering due to the low-pass effect of the reduced spectral representation.

We also qualitatively compare our method with [WLCJ19] in Figure 11; this is a point-based approach that enforces smoothness of adversarial examples by employing a regularization term on the mean curvature distortion. These adversarial point clouds are generated using the default parameters for [WLCJ19] ($\lambda_1 = 0.1$, $\lambda_2 = 1$, $k = 16$), with 15 exponential search iterations to tune the adversarial loss coefficient.

Mesh resolution. In Figure 12 we show the impact of the mesh resolution to the adversarial surface produced by our method. We give as input to the PointNet classifier (trained on the full resolution SHREC'14 shapes) the same shape sampled with different number

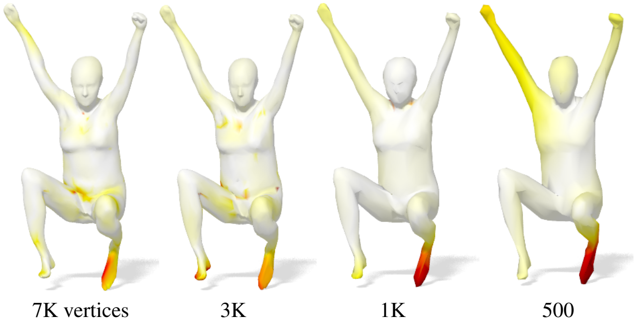

Figure 12: *An example of a successful targeted attack carried out with our method with different input mesh resolution. The deformation applied to the input mesh is consistent despite the mesh resolution.*

of points. The deformation of the adversarial surface is consistent despite the different sampling.

Localization. We conclude by reporting a surprising phenomenon that we observed in many of our tests, namely the localization of the perturbation on *semantic* parts of the shape under attack. This can be seen in some of the examples throughout the manuscript, most notably in Figure 8 where the first triplet of shapes exhibit variations in the head orientation. We consider this remarkable, in that neither the classifier, nor the attack itself are informed about the shape parts and semantics in any step of the entire process. Another example of this is shown in Figure 13 at increasing attack bandwidth. Band-limited attacks seem to concentrate on high-level features in a similar way, although in a completely different context, as compressed manifold modes [NVT*14] localize around prominent shape features.

7. Discussion and conclusions

We introduced a new approach for generating adversarial attacks against learning models fed on deformable 3D objects. Our model revolves around the idea that, when dealing with organic shapes, the "noticeability" of an attack is closely related to its smoothness. This principle might not hold for images, where smooth changes in pixel values (i.e. a color gradient) may be easier to spot for a human observer. In contrast, smooth surface deformations such as a slight, global change in volume or a local increase in bending can be hard to perceive, even more so when the shapes are expected

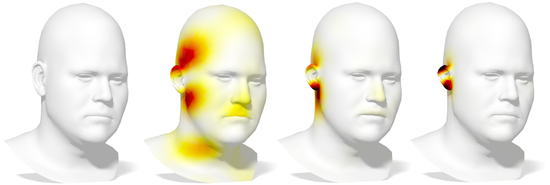

Figure 13: *Attack localization at increasing bandwidth. Left to right: original shape and the adversarial shapes obtained with* $k = 20, 40, 60$.

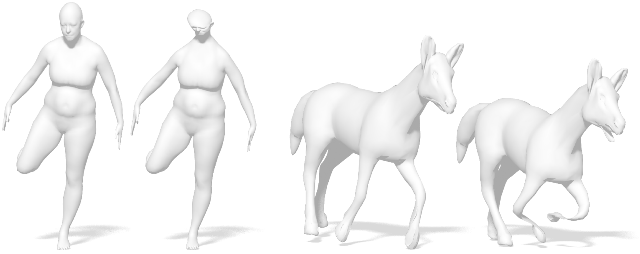

Figure 14: *Two challenging cases where the perturbation, despite being smooth and leading to misclassification, results in unnatural looking shapes.*

to deform. This is especially true when the adversarial surface is observed without an unperturbed reference to compare against – the typical case in a realistic scenario. We showed examples of such attacks on classification problems, over different datasets and with different surface representations. The results we obtained are promising, and suggest that adversarial learning for surface data holds the potential for further discoveries.

Limitations. Similarly to existing techniques in the image and point cloud domains, one limitation of our current approach lies in the difficulty to *control* the localization of the adversarial perturbations, resulting in failure cases where the input shape is deformed in semantically implausible ways; see Figure 14 for examples. One possible way to introduce a form of localization is to point-wise multiply the deformation field by the (inverse of) scalar curvature, or to steer the field anisotropically based on the local curvature directions [ARAC14]. Further, since our current approach is currently designed for deformable surfaces, it can not be applied as-is to *rigid* objects such as ShapeNet [CFG*15], where free-form deformations are not expected. Introducing part or symmetry-awareness in our framework may be a viable solution. We keep these as potential directions of follow-up work.

Future directions. We believe there are many possible ways to pursue this direction further. For example, while here we focused on classification problems in analogy to the classical settings dealing with image and point cloud data, *regression* problems may also be considered. In the context of graphics and geometry processing, regression problems arise in shape modeling and reconstruction among other sub-areas. Furthermore, the adoption of existing attack and defense techniques from the more generic literature might be replaced by ad-hoc methods for geometric data; for example, by devising attacks based on the processing of tangent vector fields. Finally, an important question that remains open is the *transferability* [PMG16] of our attacks across multiple inputs. Differently from the Euclidean setting, transferring an attack from a surface domain to another requires invoking the notion of a *map* between surfaces, which in turn involves solving a correspondence problem. We leave the question as to whether such maps are strictly necessary for adversarial purposes, as an exciting open problem for future research.

Acknowledgments

This work was supported by the ERC Starting Grant No. 802554 (SPECGEO), and the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University. LC was partially supported by the ERC Consolidator Grant No. 724228 (LEMAN).

References

[ABK15] Aflalo Y., Brezis H., Kimmel R.: On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences 8*, 2 (2015), 1141–1160. 3

[ACW18] Athalye A., Carlini N., Wagner D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. ICML* (2018), vol. 80, pp. 274–283. 2

[AEIK18] Athalye A., Engstrom L., Ilyas A., Kwok K.: Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning* (2018), vol. 80, pp. 284–293. 2

[ARAC14] Andreux M., Rodolà E., Aubry M., Cremers D.: Anisotropic laplace-beltrami operators for shape analysis. In *European Conference on Computer Vision (Workshops)* (2014), Springer, pp. 299–312. 10

[BHKB20] Bunge A., Herholz P., Kazhdan M., Botsch M. U.: Polygon laplacian made simple. *Computer Graphics Forum 39*, 2 (2020). 3

[BMCM18] Bietti A., Mialon G., Chen D., Mairal J.: A kernel perspective for regularizing deep neural networks. *arXiv preprint arXiv:1810.00363* (2018). 2

[BMR*17] Brown T. B., Mané D., Roy A., Abadi M., Gilmer J.: Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017). 2

[BRLB14] Bogo F., Romero J., Loper M., Black M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE. 7

[CFG*15] Chang A. X., Funkhouser T., Guibas L., Hanrahan P., Huang Q., Li Z., Savarese S., Savva M., Song S., Su H., et al.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 10

[CKG19] Chaturvedi A., KP A., Garain U.: Exploring the robustness of nmt systems to nonsensical inputs. *arXiv preprint arXiv:1908.01165* (2019). 2

[CNK*17] Chen Y., Nadji Y., Kountouras A., Monrose F., Perdisci R., Antonakakis M., Vasiloglou N.: Practical attacks against graph-based clustering. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), pp. 1125–1142. 2

[CRK19] Cohen J., Rosenfeld E., Kolter Z.: Certified adversarial robustness via randomized smoothing. In *Proc. ICML* (2019), Chaudhuri K., Salakhutdinov R., (Eds.), vol. 97, pp. 1310–1320. 2

[CRT04] Clarenz U., Rumpf M., Telea A.: Finite elements on point based surfaces. In *Proceedings of the First Eurographics conference on Point-Based Graphics* (2004), Eurographics Association, pp. 201–211. 3

[CW17] Carlini N., Wagner D. A.: Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017* (2017), IEEE Computer Society, pp. 39–57. 2, 4, 5, 6, 7, 8, 9

[CW18] Carlini N., Wagner D.: Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)* (2018), IEEE, pp. 1–7. 2

[CXC*19] Cao Y., Xiao C., Cyr B., Zhou Y., Park W., Rampazzi S., Chen Q. A., Fu K., Mao Z. M.: Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), pp. 2267–2281. 3

[CZS*17] Chen P.-Y., Zhang H., Sharma Y., Yi J., Hsieh C.-J.: ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (New York, NY, USA, 2017), AISec '17, ACM, pp. 15–26. 2

[DBV16] Defferrard M., Bresson X., Vandergheynst P.: Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2016), NIPS'16, Curran Associates Inc., p. 3844–3852. 7

[DLT*18] Dai H., Li H., Tian T., Huang X., Wang L., Zhu J., Song L.: Adversarial attack on graph structured data. *arXiv preprint arXiv:1806.02371* (2018). 2

[GDK*19] Gleave A., Dennis M., Kant N., Wild C., Levine S., Russell S.: Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615* (2019). 2

[GH97] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., p. 209–216. 7

[GLSQ18] Gao J., Lanchantin J., Soffa M. L., Qi Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)* (2018), IEEE, pp. 50–56. 2

[GMF*18] Gilmer J., Metz L., Faghri F., Schoenholz S. S., Raghu M., Wattenberg M., Goodfellow I.: Adversarial spheres. *arXiv preprint arXiv:1801.02774* (2018). 2

[GRYL20] Gong C., Ren T., Ye M., Liu Q.: Maxup: A simple way to improve generalization of neural network training. *arXiv preprint arXiv:2002.09024* (2020). 2

[GSS14] Goodfellow I. J., Shlens J., Szegedy C.: Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014). 2

[HRF19] He Z., Rakin A. S., Fan D.: Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2

[HRTG19] Hamdi A., Rojas S., Thabet A., Ghanem B.: Advpc: Transferable adversarial perturbations on 3d point clouds. *arXiv preprint arXiv:1912.00461* (2019). 2, 6

[JJZS19] Jin D., Jin Z., Zhou J. T., Szolovits P.: Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932* (2019). 2

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 7

[KGB16] Kurakin A., Goodfellow I., Bengio S.: Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016). 5, 8

[KGB17] Kurakin A., Goodfellow I. J., Bengio S.: Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net. 7

[KHM19] Khoury M., Hadfield-Menell D.: Adversarial training with voronoi constraints. *arXiv preprint arXiv:1905.01019* (2019). 2

[KSW15] Kingma D. P., Salimans T., Welling M.: Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems 28*, Cortes C., Lawrence N. D., Lee D. D., Sugiyama M., Garnett R., (Eds.). Curran Associates, Inc., 2015, pp. 2575–2583. 2

[KW16] Kipf T. N., Welling M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016). 2

[LCZH18] Liu X., Cheng M., Zhang H., Hsieh C.-J.: Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 369–385. 2

[LLW*19] Li Y., Li L., Wang L., Zhang T., Gong B.: NATTACK: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (2019), vol. 97, pp. 3866–3876. 2

[LYS19] Liu D., Yu R., Su H.: Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), IEEE, pp. 2279–2283. 2, 6

[MBBV15] Masci J., Boscaini D., Bronstein M., Vandergheynst P.: Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops* (2015), pp. 37–45. 2

[MMS*18] Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A.: Towards deep learning models resistant to adversarial attacks. In *Proc. ICLR* (2018). 2, 6

[NVT*14] Neumann T., Varanasi K., Theobalt C., Magnor M., Wacker M.: Compressed manifold modes for mesh processing. *Computer Graphics Forum 33*, 5 (2014), 35–44. 9

[PMG16] Papernot N., McDaniel P., Goodfellow I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016). 4, 10

[PMG*17] Papernot N., McDaniel P., Goodfellow I., Jha S., Celik Z. B., Swami A.: Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (New York, NY, USA, 2017), ASIA CCS '17, ACM, pp. 506–519. 2

[PSR*14] Pickup D., Sun X., Rosin P. L., et al.: SHREC'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval* (2014), EG 3DOR'14, Eurographics Association. 7

[QSMG17] Qi C. R., Su H., Mo K., Guibas L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 2, 7

[RBSB18] Ranjan A., Bolkart T., Sanyal S., Black M. J.: Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)* (Sept. 2018), vol. Lecture Notes in Computer Science, vol 11207, Springer, Cham, pp. 725–741. 7, 8

[Reu10] Reuter M.: Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision 89*, 2-3 (2010), 287–308. 3

[RHO*19] Rony J., Hafemann L. G., Oliveira L. S., Ayed I. B., Sabourin R., Granger E.: Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2

[SGI19] Sarkar A., Gupta N. K., Iyengar R.: Enforcing linearity in dnn succours robustness and adversarial image generation. *arXiv preprint arXiv:1910.08108* (2019). 2

[STL*18] Sun M., Tang J., Li H., Li B., Xiao C., Chen Y., Song D.: Data poisoning attack against unsupervised node embedding methods. *arXiv preprint arXiv:1810.12881* (2018). 2

[SWC*20] SUN H., WANG R., CHEN K., UTIYAMA M., SUMITA E., ZHAO T.: Robust unsupervised neural machine translation with adversarial training. *arXiv preprint arXiv:2002.12549* (2020). 2

[SWT*20] SUN Y., WANG S., TANG X., HSIEH T.-Y., HONAVAR V.: Non-target-specific node injection attacks on graph neural networks: A hierarchical reinforcement learning approach. In *Proc. WWW* (2020). 3

[SWYL18] SUN L., WANG J., YU P. S., LI B.: Adversarial attack and defense on graph data: A survey. *arXiv preprint arXiv:1812.10528* (2018). 2

[SZS*13] SZEGEDY C., ZAREMBA W., SUTSKEVER I., BRUNA J., ERHAN D., GOODFELLOW I., FERGUS R.: Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013). 2

[TYHJ20] TSAI T., YANG K., HO T.-Y., JIN Y.: Robust adversarial objects against deep learning models. In *Proc. AAAI* (2020). 6

[WLCJ19] WEN Y., LIN J., CHEN K., JIA K.: Geometry-aware generation of adversarial and cooperative point clouds. *arXiv preprint arXiv:1912.11171* (2019). 2, 6, 9

[WLW*19] WANG D., LI C., WEN S., NEPAL S., XIANG Y.: Daedalus: Breaking non-maximum suppression in object detection via adversarial examples. *arXiv preprint arXiv:1902.02067* (2019). 2

[XML*19] XU H., MA Y., LIU H., DEB D., LIU H., TANG J., JAIN A.: Adversarial attacks and defenses in images, graphs and text: A review. *arXiv preprint arXiv:1909.08072* (2019). 2

[XQL19] XIANG C., QI C. R., LI B.: Generating 3d adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 9136–9144. 2, 6

[XTG*19] XIE C., TAN M., GONG B., WANG J., YUILLE A., LE Q. V.: Adversarial examples improve image recognition. *arXiv preprint arXiv:1911.09665* (2019). 2

[XWM*19] XIE C., WU Y., MAATEN L. V. D., YUILLE A. L., HE K.: Feature denoising for improving adversarial robustness. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), pp. 501–509. 2

[XYL*19] XIAO C., YANG D., LI B., DENG J., LIU M.: Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 6898–6907. 2

[XZL*19] XU K., ZHANG G., LIU S., FAN Q., SUN M., CHEN H., CHEN P.-Y., WANG Y., LIN X.: Evading real-time person detectors by adversarial t-shirt. *arXiv preprint arXiv:1910.11099* (2019). 2

[ZAG18] ZÜGNER D., AKBARNEJAD A., GÜNNEMANN S.: Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 2847–2856. 2

[ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D., BLACK M. J.: 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 7

[ZL19] ZHANG Y., LIANG P.: Defending against whitebox adversarial attacks via randomized discretization. In *Proceedings of Machine Learning Research* (2019), Chaudhuri K., Sugiyama M., (Eds.), vol. 89, pp. 684–693. 2

[ZLSJ19] ZHANG Y., LIANG G., SALEM T., JACOBS N.: Defense-pointnet: Protecting pointnet against adversarial attacks. In *2019 IEEE International Conference on Big Data (Big Data)* (2019), IEEE, pp. 5654–5660. 2

[ZWCL20] ZHAO Y., WU Y., CHEN C., LIM A.: On isometry robustness of deep 3d point cloud models under adversarial attacks. *arXiv preprint arXiv:2002.12222* (2020). 2

[ZYJ*19] ZHANG H., YU Y., JIAO J., XING E., GHAOUI L. E., JORDAN M.: Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning* (Long Beach, California, USA, 09–15 Jun 2019), Chaudhuri K., Salakhutdinov R., (Eds.), vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 7472–7482. 2

[ZZG*19] ZHANG H., ZHENG T., GAO J., MIAO C., SU L., LI Y., REN K.: Data poisoning attack against knowledge graph embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (2019), AAAI Press, pp. 4853–4859. 2