Supplementary Material for: Isometric 3D Adversarial Examples in the Physical World

Yibo Miao[1,3⋆], **Yinpeng Dong**[2,3†], **Jun Zhu**[2,3,4,5], **Xiao-Shan Gao**[1†]
[1] KLMM, UCAS, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China
[2] Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua-Bosch Joint ML Center,
THBI Lab, BNRist Center, Tsinghua University, Beijing 100084, China
[3] RealAI [4] Peng Cheng Laboratory [5] Pazhou Laboratory (Huangpu), Guangzhou, China
yibomiao21@163.com, {dongyinpeng, dcszj}@tsinghua.edu.cn, xgao@mmrc.iss.ac.cn

A Proof

Theorem 1. *Let S and $\tilde{S}$ denote two surfaces of $\mathbb{R}^3$; $\varphi : S \to \tilde{S}$ denote a diffeomorphism that takes a point v in S to point $v' = \varphi(v)$ in $\tilde{S}$; and $K(\cdot)$ be the Gaussian curvature of the points. If $|K(v) - K(v')| < \epsilon$ for any point v, then the surfaces S and $\tilde{S}$ are ϵ-isometric.*

Proof. We will prove Theorem 1 in geodesic polar coordinates [1], which are defined as follows.

Definition 1. *Choose in the tangent plane $T_v(S)$, a system of polar coordinates (ρ, θ) where ρ is the polar radius and θ,$0 < \theta < 2\pi$, is the polar angle, the pole of which is the origin 0 of $T_v(S)$. Set $\exp_v(l) = L$, where $\exp_v$ is the exponential map at v and l is the closed half-line which corresponds to $\theta = 0$. Since $\exp_p : T_v(S) - l \to S - L$ is a diffeomorphism, we may parametrize the points of $S - L$ by the coordinates (ρ, θ), which are called geodesic polar coordinates.*

Lemma 1 (Theorem 4.27 in [3])**.** *Let (ρ, θ) be geodesic polar coordinates. Then the coefficients $E = E(\rho, \theta)$ and $F = F(\rho, \theta)$ of the first fundamental form satisfy the conditions $E = 1, F = 0$.*

Remark 1. *The geometric meaning of the fact that $F = 0$ is that in a normal neighborhood the family of geodesic circles is orthogonal to the family of radial geodesics. This fact is known as the Gauss lemma.*

We can see that the first fundamental form $ds^2 = d\rho^2 + Gd\theta^2$ in a polar system by Lemma 1. By applying the Gauss formula [1],

$$-\frac{1}{\sqrt{EG}}\left\{\left(\frac{(\sqrt{E})_v}{\sqrt{G}}\right)_v + \left(\frac{(\sqrt{G})_u}{\sqrt{E}}\right)_u\right\} = \frac{LN - M^2}{EG} = K. \tag{A.1}$$

Remark 2. *The first fundamental form is the expression of how the surface S inherits the natural inner product of $\mathbb{R}^3$. Geometrically, the first fundamental form allows us to make measurements on the surface (lengths of curves, angles of tangent vectors, areas of regions) without referring back to the ambient space $\mathbb{R}^3$ where the surface lies. The second fundamental form describes the shape of the surface in the ambient space $\mathbb{R}^3$. The Gaussian curvature can be defined by the coefficients of the first fundamental form and the coefficients of the second fundamental form. The Gauss formula and the Mainardi-Codazzi equations reveal the relations between the first and second fundamental forms of a surface. Gauss formula expresses the Gaussian curvature as a function of the coefficients of the first fundamental form and its derivatives, which is also known as Gauss' Theorema Egregium.*

⋆This work was done when Yibo Miao was intern at RealAI, Inc; †Corresponding authors.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

the Gaussian curvature K can be written

$$K = -\frac{(\sqrt{G})_{\rho\rho}}{\sqrt{G}}. \tag{A.2}$$

We state the following lemma to calculate G.

Lemma 2 (Theorem 4.6.3 in [1]). *Let (ρ, θ) be geodesic polar coordinates. Then the coefficients $G = G(\rho, \theta)$ of the first fundamental form satisfy the conditions $\lim_{\rho\to 0} G = 0, \lim_{\rho\to 0}(\sqrt{G})_\rho = 1$.*

If $K > 0$, $(\sqrt{G})_{\rho\rho} + K\sqrt{G} = 0$. The general solution is given by

$$\sqrt{G} = f(\theta)\cos(\sqrt{K}\rho) + g(\theta)\sin(\sqrt{K}\rho). \tag{A.3}$$

From the conclusion in Lemma 2, we know $f(\theta) = 0$ and $g(\theta) = \frac{1}{\sqrt{K}}$. We finally have

$$ds^2 = d\rho^2 + \frac{1}{K}\sin^2(\sqrt{K}\rho)\cdot d\theta^2. \tag{A.4}$$

If $K = 0$, $(\sqrt{G})_{\rho\rho} = 0$. Thus,

$$(\sqrt{G})_\rho = f(\theta). \tag{A.5}$$

From the conclusion in Lemma 2, we know $f(\theta) = 1$ and

$$\sqrt{G} = \rho + g(\theta). \tag{A.6}$$

By reusing Lemma 2, we know $g(\theta) = 0$ and $\sqrt{G} = \rho$. We finally have

$$ds^2 = d\rho^2 + \rho^2\cdot d\theta^2. \tag{A.7}$$

If $K < 0$, $(\sqrt{G})_{\rho\rho} + K\sqrt{G} = 0$. The general solution is given by

$$\sqrt{G} = f(\theta)\cosh(\sqrt{-K}\rho) + g(\theta)\sinh(\sqrt{-K}\rho). \tag{A.8}$$

From the conclusion in Lemma 2, we know $f(\theta) = 0$ and $g(\theta) = \frac{1}{\sqrt{-K}}$. We finally have

$$ds^2 = d\rho^2 + \frac{1}{-K}\sinh^2(\sqrt{-K}\rho)\cdot d\theta^2. \tag{A.9}$$

Thus we can derive

$$\begin{aligned}
&\left|\tfrac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho) - \tfrac{1}{K(v')}\sin^2(\sqrt{K(v')}\rho)\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[K(v')\sin^2(\sqrt{K(v)}\rho) - K(v)\sin^2(\sqrt{K(v')}\rho)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[K(v')\left(K(v)\rho^2 - \tfrac{K(v)^2\rho^4}{3}\right) - K(v)\left(K(v')\rho^2 - \tfrac{K(v')^2\rho 4}{3}\right)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\cdot K(v)K(v')\cdot\tfrac{\rho^4}{3}(K(v') - K(v))\right| < \tfrac{\rho^4}{3}\epsilon.
\end{aligned} \tag{A.10}$$

$$\left|\frac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho) - \rho^2\right| = \left|\frac{1}{K(v)}\left(K(v)\rho^2 - \frac{K(v)^2\rho^4}{3}\right) - \rho^2\right| = \left|-\frac{p4}{3}(K(v) - 0)\right| < \frac{p^4}{3}\epsilon. \tag{A.11}$$

$$\begin{aligned}
&\left|-\tfrac{1}{K(v)}\sinh^2(\sqrt{-K(v)}\rho) + \tfrac{1}{K(v')}\sinh^2(\sqrt{-K(v')}\rho)\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[-K(v')\sinh^2(\sqrt{-K(v)}\rho) + K(v)\sinh^2(\sqrt{-K(v')}\rho)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\left[-K(v')\left(-K(v)\rho^2 + \tfrac{K(v)^2\rho^4}{3}\right) + K(v)\left(-K(v')\rho^2 + \tfrac{K(v')^2\rho^4}{3}\right)\right]\right| \\
&= \left|\tfrac{1}{K(v)K(v')}\cdot K(v)K(v')\cdot\tfrac{\rho^4}{3}(K(v') - K(v))\right| < \tfrac{\rho^4}{3}\epsilon.
\end{aligned} \tag{A.12}$$

$$\left|-\frac{1}{K(v)}\sinh^2(\sqrt{-K(v)}\rho) - \rho^2\right| = \left|-\frac{1}{K(v)}\left(-K(v)\rho^2 + \frac{K(v)^2\rho^4}{3}\right) - \rho^2\right| < \frac{p^4}{3}\epsilon. \tag{A.13}$$

$$
\begin{aligned}
&\left|\tfrac{1}{K(v)}\sin^2(\sqrt{K(v)}\rho)+\tfrac{1}{K(v')}\sinh^2(\sqrt{-K(v')}\rho)\right| \\
=&|\ \tfrac{1}{K(v)K(v')}\left[K(v')\sin^2(\sqrt{K(v)}\rho)+K(v)\sinh^2(\sqrt{-K(v')}\rho)\ \right| \\
=&\left|\tfrac{1}{K(v)K(v')}\left[K(v')\left(K(v)\rho^2-\tfrac{K(v)^2\rho^4}{3}\right)+K(v)\left(-K(v')\rho^2+\tfrac{K(v')^2\rho^4}{3}\right)\right]\right| \\
=&\left|\tfrac{1}{K(v)K(v')}\cdot K(v)K(v')\cdot\tfrac{\rho^4}{3}(K(v')-K(v))\right|<\tfrac{\rho^4}{3}\epsilon.
\end{aligned}
\tag{A.14}
$$

Thus from the condition $|K(v)-K(v')|<\epsilon$ we have

$$
|ds^2-d\tilde{s}^2|<\frac{\rho^4}{3}\cdot d\theta^2\cdot\epsilon, \tag{A.15}
$$

where ds is the element of arc length on a curved surface.

Let $s(C)=n\cdot ds$ and $s(\tilde{C})=n\cdot d\tilde{s}$, then we have

$$
|s(\tilde{C})-s(C)|=n|ds-d\tilde{s}|<s(C)\cdot\frac{d\theta^2}{ds^2+dsd\hat{s}}\cdot\frac{\rho^4}{3}\epsilon \tag{A.16}
$$

This equation consequently results in

$$
\left(1-\frac{d\theta^2}{ds^2+dsd\hat{s}}\cdot\frac{\rho^4}{3}\cdot\epsilon\right)s(C)<s(\tilde{C})<\left(1+\frac{d\theta^2}{ds^2+dsd\hat{s}}\cdot\frac{\rho^4}{3}\cdot\epsilon\right)s(C) \tag{A.17}
$$

Thus the surfaces S and $\tilde{S}$ are ϵ-isometric. □

B Algorithm

We provide the algorithm to solve problem (6) using Bayesian Optimization in Alg. 1. We choose GP as a surrogate model, which provides a Bayesian posterior distribution to describe the objective function $f(t;D_n)\sim GP(\mu(D_n),\Sigma(D_n,D_n))$, where $f(t;D_n)$ is the modeling of the unknown function $\mathcal{L}_{ce}(t(S(\mathcal{M}_{adv})),y^*)$ and $D_n=\{t_i,\mathcal{L}_{ce}(t_i(S(\mathcal{M}_{adv})),y^*)\}_{i=1}^n$ is the n samples observed so far. μ and Σ are the mean and covariance functions, respectively. We use the expected improvement (EI) [2, 4] acquisition function $\alpha_{EI}(t;D_n)=E_{f(t;D_n)\sim GP(\mu(D_n),\Sigma(D_n,D_n))}\left[\max\left(f(t)-f^+_{D_n},0\right)\right]$, which measures the expected improvement of each point with respect to the current best value, where $f^+_{D_n}=\max_{i\leq n}f(t_i)$ is the best value observed so far. Then, the target function f is sampled by $\arg\max_t\alpha_{EI}(t;D_n)$ to better explore the space of all transformations by selecting the next query point t_{n+1} in the region where the prediction is high and the model is very uncertain.

C Supplementary experimental results

We provide more experimental results in this section. All of the experiments are conducted on NVIDIA 3080 Ti GPUs. The source code is submitted as part of the supplementary material, and will be released after the review process.

C.1 The effects of the penalty parameter λ_1

We study the effects of the penalty parameter λ_1 of the Gaussian curvature consistency regularization term in Sec. 4.6. We adjust the value of λ_1 to perform quantitative and qualitative experiments. The qualitative visualization results in Fig. C.1 demonstrate the irregularity of the 3D objects as λ_1 is turned down. Our default value of $\lambda_1=1$ guarantees stealthiness.

C.2 The effectiveness of Bayesian optimization

To investigate the effectiveness of Bayesian optimization to find the initial transformations, we test the attack success rate for randomly selecting the initial transformation in the MaxOT algorithm. As shown in Table C.1, the improvement of using Bayesian optimization over random initialization in the MaxOT algorithm is significant. This confirms the importance of Bayesian optimization methods in our approach.

Algorithm 1 Solve problem (6) using Bayesian Optimization

Require: Observation data D_K, surrogate model $GP\left(\mu\left(\cdot\right), \Sigma\left(\cdot,\cdot\right)\right)$, objective function $\mathcal{L}_{ce}\left(t(S(\mathcal{M}_{adv})), y^*\right)$, the number of transformations in the MaxOT algorithm S, the number of gradient descent iterations K, and learning rate η_t.

$D_K = \{t_i, \mathcal{L}_{ce}\left(t_i(S(\mathcal{M}_{adv})), y^*\right)\}_{i=1}^{K}$
for $s = 1$ **to** S **do**
    Update the surrogate model:

$$f\left(t; D_{sK}\right) \sim GP\left(\mu\left(D_{sK}\right), \Sigma\left(D_{sK}, D_{sK}\right)\right);$$

    Calculate the acquisition function:

$$\alpha_{EI}\left(t; D_{sK}\right) \leftarrow E_{f(t;D_{sK}) \sim GP(\mu(D_{sK}), \Sigma(D_{sK}, D_{sK}))}\left[\max\left(f(t) - f^{+}_{D_{sK}}, 0\right)\right];$$

    Select $t_{sK+1} \leftarrow \arg\max_t \alpha_{EI}\left(t; D_{sK}\right)$;
    for $k = 1$ **to** K **do**
        Update t_{sK+1+k} with gradient ascent:

$$t_{sK+1+k} \leftarrow t_{sK+k} - \eta_t \cdot \nabla_{t_{sK+k}} \mathcal{L}_{ce}\left(t_{sK+k}(S(\mathcal{M}_{adv})), y^*\right);$$

        if $k = K$ **then**
            $t_s^* \leftarrow t_{sK+1+k}$;
        end if
    end for
    $\overline{D}_s \leftarrow \{t_{sK+1+k}, \mathcal{L}_{ce}\left(t_{sK+1+k}(S(\mathcal{M}_{adv})), y^*\right)\}_{k=0}^{K-1}$;
    $D_{(s+1)K} \leftarrow D_{sK} \cup \overline{D}_s$;
end for
return $t_s^*, s = 1, \cdots, S$

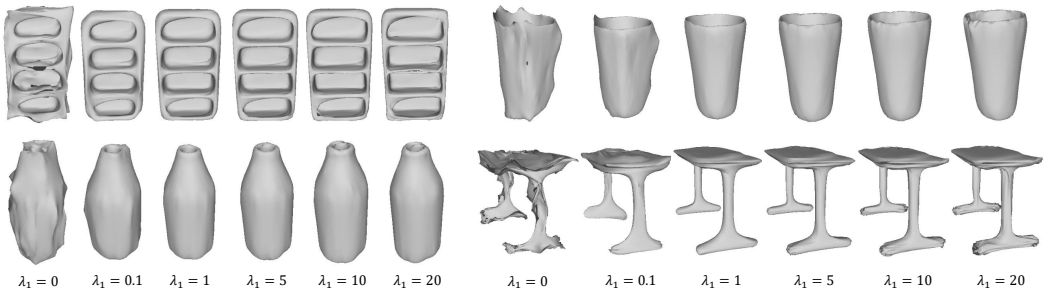

Figure C.1: Qualitative visualization results of the effects of the penalty parameter λ_1. When λ_1 is turned down, the naturalness of the adversarial objects is worse. Our default value of $\lambda_1 = 1$ guarantees stealthiness.

Table C.1: Quantitative results of attacking different models using EOT, MaxOT w/o BO and MaxOT. The improvement of using Bayesian optimization over random initialization in the MaxOT algorithm is significant.

Model	PointNet			PointNet++			DGCNN		
	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$	ASR	$\mathcal{D}_c$	$\mathcal{D}_g$
EOT	76.20%	0.0074	0.0009	74.28%	0.0094	0.0007	65.72%	0.0068	0.0041
MaxOT w/o BO	80.59%	0.0075	0.0009	81.09%	0.0094	0.0007	70.25%	0.0067	0.0039
MaxOT	82.50%	0.0074	0.0009	84.14%	0.0094	0.0006	72.40%	0.0067	0.0039

C.3 The effectiveness of Gaussian curvature

To investigate the effectiveness of Gaussian curvature, we test the mean curvature [1] as a substitute for Gaussian curvature. As shown in Fig. C.2, the mean curvature consistency regularization term does not guarantee naturalness and smoothness. This is because the Gaussian curvature consistency regularization term theoretically provides a sufficient condition (see Theorem **??**) to ensure that two surfaces are ϵ-isometric, while the mean curvature consistency regularization term is not.

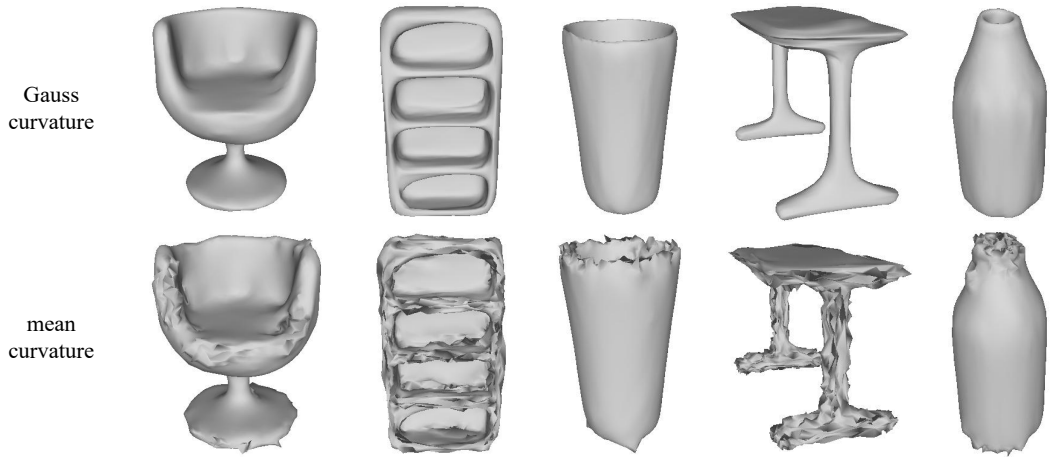

Figure C.2: Qualitative visualization results of Gaussian curvature and mean curvature. The mean curvature consistency regularization term does not guarantee naturalness and smoothness, while Gaussian curvature does.

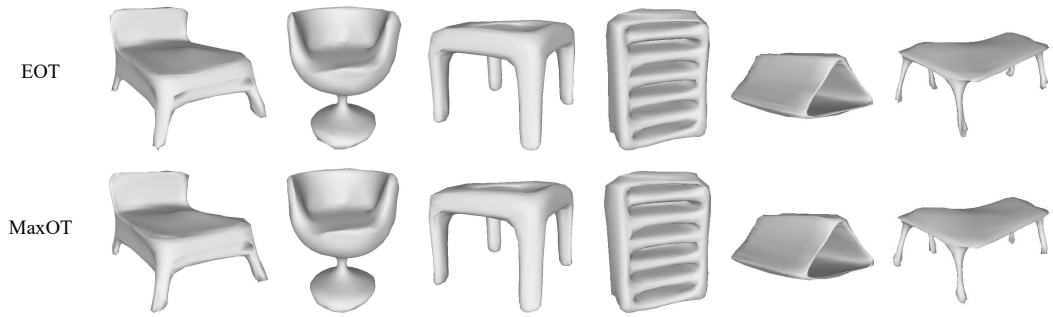

Figure C.3: Qualitative visualization results of MaxOT and EOT. MaxOT algorithm and EOT algorithm generate adversarial objects with the same degree of naturalness.

Table C.2: The effect of the penalty parameter λ_2. When λ_2 is large, the attack success rate decreases rapidly. Our default value of $\lambda_2 = 0.2$ give the best result for balancing the attack success rate and stealthiness.

Model	$\lambda_2 = 0$	$\lambda_2 = 0.02$	$\lambda_2 = 0.2$	$\lambda_2 = 1$	$\lambda_2 = 2$	$\lambda_2 = 4$
PointNet	98.72%	98.67%	98.45%	94.22%	78.03%	69.45%
PointNet++	99.69%	99.66%	99.58%	89.25%	79.82%	72.42%
DGCNN	85.13%	84.82%	84.16%	69.76%	60.48%	52.46%

C.4 Qualitative visualization results of MaxOT and EOT

We compare MaxOT to EOT in Sec. 4.3. The qualitative visualization results of MaxOT and EOT in Fig. C.3 demonstrate the same degree of naturalness, while our proposed MaxOT algorithm outperforms the EOT algorithm in terms of attack success rate and efficiency.

C.5 The effects of the penalty parameter λ_2 and λ_3

To investigate the effect of the penalty parameter λ_2 and λ_3, we adjust the value of λ_2 and λ_3 to perform quantitative and qualitative experiments.

As shown in Table C.2 and Table C.3, when λ_2 or λ_3 is tuned high, the attack success rate decreases rapidly. As shown in Fig. C.4, when λ_2 is adjusted lower, 3D objects show local unevenness and minor self-intersection. When λ_3 is adjusted lower, 3D objects show large areas of self-intersection. Our default value of $\lambda_2 = 0.2$ and $\lambda_3 = 0.8$ give the best result for balancing the attack success rate and stealthiness.

Table C.3: The effect of the penalty parameter λ_3. When λ_3 is large, the attack success rate decreases rapidly. Our default value of $\lambda_3 = 0.8$ give the best result for balancing the attack success rate and stealthiness.

Model	$\lambda_3 = 0$	$\lambda_3 = 0.08$	$\lambda_3 = 0.8$	$\lambda_3 = 4$	$\lambda_3 = 8$	$\lambda_3 = 16$
PointNet	99.02%	98.86%	98.45%	89.29%	78.42%	63.01%
PointNet++	99.73%	99.70%	99.58%	86.56%	80.47%	71.96%
DGCNN	87.28%	86.89%	84.16%	65.92%	56.92%	42.35%

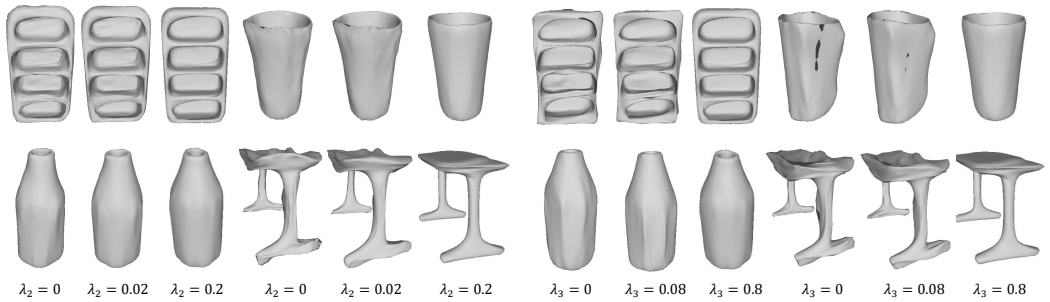

Figure C.4: Qualitative visualization results of the effects of the penalty parameter λ_2 and λ_3. When λ_2 is adjusted lower, 3D objects show local unevenness and minor self-intersection. When λ_3 is adjusted lower, 3D objects show large areas of self-intersection. Our default value of $\lambda_2 = 0.2$ and $\lambda_3 = 0.8$ guarantees stealthiness.

Table C.4: The transfer-based attack success rates on three models by various attacks. Our ϵ-ISO attack has much higher success rates than the baselines.

White-box Target Model	Attacks	Black-box Victim Model		
		PointNet	PointNet++	DGCNN
PointNet	KNN	-	11.1%	10.7%
	$GeoA^3$	-	11.5%	2.5%
	ϵ-ISO	-	47.6%	35.8%
PointNet++	KNN	6.4%	-	7.9%
	$GeoA^3$	9.4%	-	19.7%
	ϵ-ISO	32.9%	-	51.2%
DGCNN	KNN	7.2%	32.2%	-
	$GeoA^3$	12.4%	24.2%	-
	ϵ-ISO	55.4%	62.7%	-

C.6 Evaluation on the Transferability

We conduct experiments on the transfer-based attacks. We generate 3D adversarial examples against one white-box model and evaluate the black-box attack success rates on the other black-box victim models. As shown in Table C.4, our ϵ-ISO attack has much higher success rates than the baselines. This is because our ϵ-ISO retains the geometric properties of the 3D objects well, without local outliers or anomalous deformations. Thus the crafted adversarial examples more transferable across different models.

D Proof of $\boldsymbol{n}_u \wedge \boldsymbol{n}_v = K\boldsymbol{r}_u \wedge \boldsymbol{r}_v$

The Weingarten map $\mathcal{W}$ is defined by

$$\mathcal{W}: \begin{array}{ll} T_P S & \to T_P S \\ \boldsymbol{v} = \lambda \boldsymbol{r}_u + \mu \boldsymbol{r}_v & \to \mathcal{W}(\boldsymbol{v}) = -\left(\lambda \boldsymbol{n}_u + \mu \boldsymbol{n}_v\right) \end{array} \tag{D.1}$$

Thus, $\mathcal{W}(\boldsymbol{r}_u) = -\boldsymbol{n}_u, \mathcal{W}(\boldsymbol{r}_v) = -\boldsymbol{n}_v$. The coefficient matrix of the Weingarten map $\mathcal{W}$ is $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, where $\mathcal{W}(\boldsymbol{r}_u) = -\boldsymbol{n}_u = a\boldsymbol{r}_u + b\boldsymbol{r}_v, \mathcal{W}(\boldsymbol{r}_v) = -\boldsymbol{n}_v = c\boldsymbol{r}_u + d\boldsymbol{r}_v$. Take the dot product

of each of these equations with $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$. This gives

$$\begin{aligned}
\langle -\boldsymbol{n}_u, \boldsymbol{r}_u \rangle &= a \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle + b \langle \boldsymbol{r}_v, \boldsymbol{r}_u \rangle \\
\langle -\boldsymbol{n}_u, \boldsymbol{r}_v \rangle &= a \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle + b \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle \\
\langle -\boldsymbol{n}_v, \boldsymbol{r}_u \rangle &= c \langle \boldsymbol{r}_u, \boldsymbol{r}_u \rangle + d \langle \boldsymbol{r}_v, \boldsymbol{r}_u \rangle \\
\langle -\boldsymbol{n}_v, \boldsymbol{r}_v \rangle &= c \langle \boldsymbol{r}_u, \boldsymbol{r}_v \rangle + d \langle \boldsymbol{r}_v, \boldsymbol{r}_v \rangle
\end{aligned} \tag{D.2}$$

Since $\boldsymbol{r}_u$ and $\boldsymbol{r}_v$ are tangent vectors to the surface, $\langle \boldsymbol{r}_u, \boldsymbol{n} \rangle = 0, \langle \boldsymbol{r}_v, \boldsymbol{n} \rangle = 0$. Differentiating these equations with respect to u and v gives

$$\begin{aligned}
\langle \boldsymbol{r}_{uu}, \boldsymbol{n} \rangle &= \langle \boldsymbol{r}_u, -\boldsymbol{n}_u \rangle \\
\langle \boldsymbol{r}_{uv}, \boldsymbol{n} \rangle &= \langle \boldsymbol{r}_u, -\boldsymbol{n}_v \rangle = \langle \boldsymbol{r}_v, -\boldsymbol{n}_u \rangle \\
\langle \boldsymbol{r}_{vv}, \boldsymbol{n} \rangle &= \langle \boldsymbol{r}_v, -\boldsymbol{n}_v \rangle
\end{aligned} \tag{D.3}$$

This gives

$$\begin{aligned}
L &= aE + bF, \quad M = cE + dF, \\
M &= aF + bG, \quad N = cF + dG.
\end{aligned} \tag{D.4}$$

These four scalar equations are equivalent to the single matrix equation

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \tag{D.5}$$

Thus we can derive

$$\begin{aligned}
\boldsymbol{n}_u \wedge \boldsymbol{n}_v &= (a\boldsymbol{r}_u + b\boldsymbol{r}_v) \wedge (c\boldsymbol{r}_u + d\boldsymbol{r}_v) \\
&= (ad - bc)\boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \det\left(\begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \right) \boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \frac{\det\left(\begin{bmatrix} L & M \\ M & N \end{bmatrix} \right)}{\det\left(\begin{bmatrix} E & F \\ F & G \end{bmatrix} \right)} \boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= \frac{LN - M^2}{EG - F^2} \boldsymbol{r}_u \wedge \boldsymbol{r}_v \\
&= K\boldsymbol{r}_u \wedge \boldsymbol{r}_v
\end{aligned} \tag{D.6}$$

References

[1] Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[2] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[3] Wolfgang Kühnel. *Differential geometry*, volume 77. American Mathematical Soc., 2015.

[4] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2, 1978.