基于生成对抗网络的三维模型识别攻击算法

刘 佳[1], 金志刚[2], 金诗博[1]

[1]天津中德应用技术大学软件与通信学院 天津 中国 300350
[2]天津大学电气自动化与信息工程学院 天津 中国 300072

摘要 现有三维模型识别网络对特征分布和扰动特性的关注不到位, 导致识别稳定性和灵活性差。因此, 提出一种新的对抗样本生成算法, 以探究深度网络模型容易受到攻击的原因。算法以点云为对象, 首先利用生成网络有效地学习点云关键点的特征, 兼顾原始点云分布及其对抗特性, 以生成对抗点的特征表示。此外, 生成器能够根据不同的输入点云调整对抗点的生成, 以达到欺骗原始三维模型识别网络的目的, 进而实现对三维模型深度识别网络稳定性的探究。不同于传统攻击模型的损失函数, 算法引入误分类损失扩大攻击力学习的可见范围。同时, 还在原有对抗损失函数的基础上提出了感知损失函数, 通过对比原始输入与生成样本的相似度来提高对抗样本的质量, 从而更加逼真地模拟可能出现的对抗样本。基于该设计, 算法所生成的对抗样本不仅可以欺骗三维识别网络, 甚至可以在视觉上欺骗人类, 从而实现对三维模型识别网络对抗鲁棒性的测试, 完成对深度网络模型脆弱性原因的探索。在 ModelNet10 和 ModelNet40 数据集上的对比实验及消融实验证明, 生成式对抗网络和感知损失的有机结合使算法可以有效地生成高质量的对抗样本。

关键词 对抗样本; 生成式对抗网络; 信息安全; 三维模型识别
中图法分类号 TP391 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2025.01.09

The 3D Model Recognition Attack Algorithm based on Generative Adversarial Networks

LIU Jia[1], JIN Zhigang[2], JIN Shibo[1]

[1] School of Software and Communication, Tianjin Sino-German University of Applied Sciences, Tianjin 300350, China
[2] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

Abstract The existing 3D model recognition network does not pay enough attention to the feature distribution and perturbation characteristics, which results in poor stability and flexibility of the network. In response to this problem, a novel adversarial sample generation algorithm is proposed to explore the reasons why the deep network model is vulnerable to attacks. The algorithm is based on point clouds and first uses the generation network to effectively learn the feature of the key points in the point cloud, taking into account the original point cloud distribution and the adversarial characteristics, in order to generate the feature representation of the adversarial points. Besides, the generator adjusts the generation of adversarial points according to different point cloud inputs. Therefore, the algorithm is able to achieve the purpose of deceiving the original 3D model recognition network and realize the investigation of the stability of the 3D model depth recognition network. Unlike the loss function of traditional attack models, the algorithm introduces misclassification loss to expand the visible range of attack learning. Meanwhile, the algorithm also proposes a perceptual loss function on the basis of the original adversarial loss function to improve the quality of the adversarial samples by comparing the similarity between the original input and the generated samples, so as to simulate the possible adversarial samples more realistically. Based on this design, the adversarial samples generated by the algorithm can not only deceive the 3D recognition network but can even visually deceive humans. Thus, the test of the adversarial robustness of the 3D model recognition network is realized and the exploration of the reasons for the vulnerability of the deep network model is completed. The comparison experiments and ablation experiments on ModelNet10 and ModelNet40 datasets demonstrate that the organic combination of generative adversarial networks and perceptual loss allows the algorithm to efficiently generate high-quality adversarial samples.

Key words adversarial samples; generative adversarial networks; information security; three-dimensional model identification

通讯作者: 刘佳, 硕士, 讲师, Email: liujia@tsguas.edu.cn。
收稿日期: 2023-02-13; 修改日期: 2023-04-05; 定稿日期: 2024-11-19

1 引言

近年来, 随着深度学习在三维模型识别任务中的广泛应用, 三维模型识别技术取得了突破性的进展[1]。三维模型识别技术也逐渐被应用到三维重建[2]、自动驾驶[3]、虚拟现实[4]等各个领域。然而, 神经网络的安全性缺陷也随着应用的推广逐渐暴露——样本被人为添加噪声后会使神经网络做出错误的预测。这种缺陷很容易被别有用心的攻击者利用, 尤其是在自动驾驶等需要极高准确性和安全性的领域中, 如不能及时防范, 会给使用者带来巨大的生命和财产安全损失[5-6]。因此, 提升识别模型的鲁棒性逐渐成为该领域的研究热点之一, 而研究此问题的挑战在于人为添加的噪声具有隐匿性和不确定性, 难以循迹是何种形式的干扰影响了网络性能。为此, 一部分三维视觉领域的研究者提出了生成对抗样本(即噪声干扰样本)来模拟攻击网络[7], 进而实现对模型识别网络稳定性的探究, 这种方法将三维点云划分为关键点与对抗点两个部分, 关键点代表点云易受攻击的点, 而对抗点代表受到噪声干扰的点。然而, 这些方法只关注干扰噪声给三维目标关键点带来的位移, 忽略了样本中对抗点本身的特征, 导致其生成的对抗样本可迁移性差, 不能对深度学习模型进行有效的攻击。

为了提高对抗样本的可迁移性和灵活性, 挖掘对抗样本的内在特性, 本文提出了一种新颖的对抗生成模型来生成高质量的对抗样本, 分别在攻击成功率和生成点云质量两个方面提高对抗样本的攻击能力。在所提出的网络中, 生成器可以有效地学习对抗点的表示, 并根据不同的点云输入来调整对抗样本的噪音输入, 从而达到欺骗原始三维模型识别网络的目的。此外, 本文提出了一种感知损失, 根据原始输入提高生成样本的质量, 在视觉上提高原始点云和生成样本的相似度, 更加逼真地模拟对抗样本, 进而提升三维模型识别网络的对抗鲁棒性。最后, 本文还利用常用的点云模型在 ModelNet10 和 ModelNet40 数据集上进行了大量攻击实验及消融实验。实验结果证明, 生成对抗网络和感知损失的有机结合大幅提升了对抗样本质量和对抗攻击成功率, 为神经网络的攻击鲁棒性研究提供了可靠的研究工具。

2 相关工作

2.1 点云深度学习模型

点云因其在分类、分割和关键点采样等许多应用中的成功而备受关注。现有的基于点云的深度学习模型在三维模型分类任务上表现出优异的性能。Qi 等人[8]提出直接从点上获得特征描述的 PointNet。然而, 它在以点为种子进行采样时忽略了局部细节。在文献[9]中, Qi 等人提出了 PointNet++, 以弥补输入多尺度点云数据的缺陷, 在全局特征和局部细节中找到了平衡状态。DGCNN[10]提出了 EdgeConv, 它可以嵌入到现有的多个学习框架中, 以考虑局部细节信息并保持包络不变性。对于点云数据, 最大的挑战在于三维模型的点是非结构化的, 比有序点的可用信息少。KD-networks[11]考虑使用 Kd-tree[12]对点进行标准化处理, 并学习每个点的权重来生成描述。当利用包络不变性时, 这些模型出现了一个致命的限制——缺少局部特征。RS-CNN[13]将结构网格卷积神经网络扩展到不规则点分析, 其主要贡献是从采样点和其他点的关系中学习描述符。由于这些三维卷积神经网络的计算成本很高, LP3DCNN[14]提出利用三维局部邻域中的相位来生成特征图并减少可训练参数。

2.2 对抗攻击方法

由于机器学习模型很容易被对抗性样本所欺骗, 关于模型对抗性攻击的研究受到了广泛关注[5]。大多数基于神经网络的图像分类方法很容易被对抗性样本所欺骗, 这种样本通常添加了精心设计的扰动, 使其与原始图像在视觉上保持相似, 但在特征空间中产生可分辨的差异, 进而使识别模型发生误判。这一问题在基于点云数据的目标识别中同样存在, 通过对点云数据进行一定范围的扰动, 进而达到欺骗识别模型的目的, 因此有必要设计三维模型的对抗性攻击模型以增强三维物体识别模型的鲁棒性。Xiang 等人[7]填补了这一研究空白, 提出一种对抗性样本的生成方法, 它通过移动现有的关键点和生成小规模的新增点来施加扰动。Zheng 等人[15]设计了点云显著性图, 用于转移表示攻击重要性的高分点。Wicker 等人[16]提出了迭代样本闭塞, 以选择可以被移除的点并优化对抗性样本。Tsai 等人[17]提出了一个新的模型来生成点云和对抗性物体。在这里, 对抗性物体是在真实世界中构建的, 它们可以绕过现有的防御机制。从对抗物体中提取的对抗点也显示出很好的攻击性能。Zhao 等人[18]提出采用辛普森采样和混合目标约束来实现点云的对抗性设置。以上方法都利用了关键点和单个点的局部信息来构建扰动。LG-GAN[19]解决了这个问题, 并利用对抗生成模型来重建点云, 旨在学习扰动来攻击点云模型。

2.3 三维生成方法

深度学习的最新进展促进了深度生成模型的发

展。由于点云很容易进行几何运算，并且具有表面几何的表现力，因此，在点云上应用传统的深度生成模型来重建三维模型的问题最近引起了广泛的关注。Achlioptas 等人[20]提出了第一个关于点云自编码器(AutoEncoder)的深度生成模型，该模型学习了一个具有紧凑瓶颈层的表示。Gadelha 等人[21]提出了一个树状结构的编码器-解码器，对形状进行分类并直接生成点云。Li 等人[22]提出了 PC-GAN，在传统 GAN 的基础上学习一个分层的、可解释的采样过程，也可以学习点云的多功能潜在表示。Sun 等人[23]利用自回归模型设计了 PointGrow，该模型基于语义背景和点间关联性生成点云。Zamorski 等人[24]设计了 AAE，它可以接受三维输入，以端到端的方式，利用平滑插值的形状生成点云。Yang 等人[25]提出了 PointFlow，它可以学习形状的分布来生成三维点云，还可以从任意数量的点中生成点云形状。

3 算法基本原理

3.1 方法概述

本文算法的总体流程如图 1 所示：本方法设计了扰动学习生成器(DisGAN)来学习能对原始网络造成扰动的点的特征，从而生成具有对抗特性的攻击样本。具体来说，首先提取点云的关键点 C，然后根据关键点的特征生成新的对抗点 $\hat{C}$，它与关键点之外的点 U 相结合，得到对抗样本 $\hat{P}$。图中红色点云是由生成式对抗网络（Generative adversarial network, GAN）生成的，蓝色点云是原始点云。这里使用关键点作为输入，一方面是为了挖掘对抗点的特征分布，另一方面是为了利用其余点保留原始点云的结构，保证点云质量。具体方法如下：

首先，提取对三维模型特征贡献最大的点作为三维模型的关键点，并利用 K 均值聚类算法(K-means clustering algorithm, K-means)[12] 生成关键点集群。这里，原始点云被分为关键点 C 和非关键点 U 两部分。然后，生成模型探索输入的关键点 C 的特性，由此生成对抗点 $\hat{C}$。其次，非关键点 U 和对抗点 $\hat{C}$ 相结合，生成用于攻击点云模型 T 的对抗样本。在这里，生成的点云由判别器进行判别，对生成器进行监督。最后，本文算法设计了一个感知损失来监督原始点与生成点的相似度，从而控制生成点的质量。特别地，本文算法还引入了误分类损失以增强对抗样本的攻击性。由于本文算法 DisGAN 采用了生成对抗网络(GAN)的结构。因此，整个算法框架需要对生成器和判别器进行逐步训练。

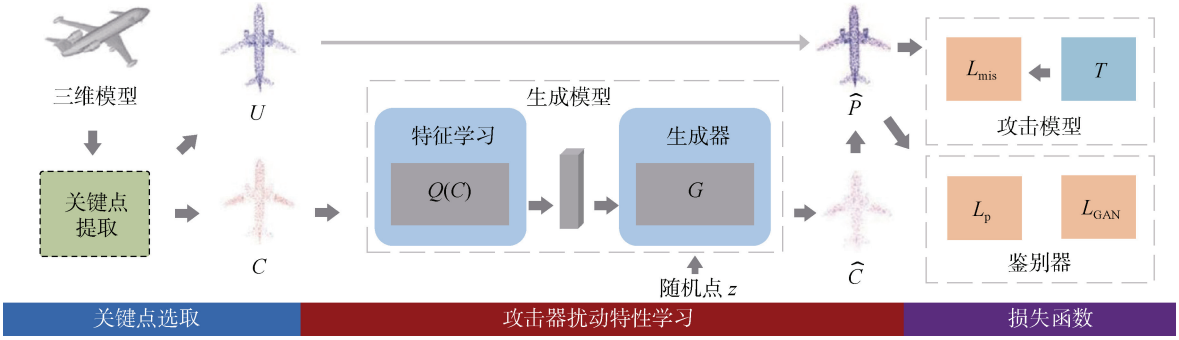

图 1 对抗样本生成的网络架构：生成器可以有效地学习对抗点的表示，并根据不同的点云输入来调整对抗样本的噪音输入，L_{mis} 通过生成模型和原有模型的比对，来提升生成模型的质量，使其更符合真实环境下的样本分布。鉴别器则保证了生成样本本身的迷惑性，使对识别模型的干扰更加有效。

Figure 1 Architecture for the adversarial generation network: The generator can effectively learn the representation of confrontation points, and adjust the noise input of the confrontation samples according to different point cloud inputs. L_{mis} improves the quality of the generated model by comparing the generated model with the original model to make it more consistent with the sample distribution in the real environment. Discriminator ensures the confusion of the generated sample itself, which is more effective to interfere with the recognition model.

3.2 点云关键点提取

点云是对三维模型表面进行采样得到的，由于三维模型表面的复杂程度不同，点云中不同的点对模型特征的贡献度也有所差异。改变特征贡献度高的点，并利用剩余点维持点云结构是最合适的手段，即用最小的成本改变三维模型特征，同时维持点云结构。因此，本文算法利用关键点作为生成器的输入，生成对抗点替换关键点，进而生成对抗样本实现对点云模型的攻击，降低运算成本，控制替换点对点云形状的影响。为了在点云中找到对特征影响最多的关键点，本方法首先利用点云学习网络 PointNet 对每个点进行特征提取。PointNet 通过对每个点的特

征向量进行池化操作来生成三维模型点的特征, 与最终的三维模型特征向量相比, 点的特征与三维模型特征重复度最高即可认为点的贡献度最大。本方法选取重复度最高的八个点作为贡献度最高的点。随后, 利用 K-means[26]对点进行聚类, 将三维模型分为八个点簇。最后, 对每个集群进行下采样得到 $m+1$ 个点, 以此来保持原有点云的结构。最终一共有 $8*(m+1)$ 个点被选为关键点, 与随机抽样相比, 本方法可以避免采样不均匀问题。在此, 本方法选取 K-means 的原因是它在聚类方法中运算速度最快, 并且它只计算两点之间的距离, 计算成本也相对较低。此外, K-means 还可以根据指定的聚类数目对点进行聚类。

3.3 基于生成对抗网络的扰动学习框架

与传统对抗攻击模型中的扰动学习模式相比, 本方法利用对抗点来替换特征临界点更加合理有效。因为传统的扰动学习模式只是针对扰动量进行数学上的建模, 与样本本身的特征并无关联。因此, 扰动度量学习不能顾及到每个对抗样本。而对抗攻击应该关注每个类别的特征边界, 学习对抗样本更深层的特征[27]。因此, 本文认为必须找到一种特征学习方法来构造攻击模型。很明显, 生成对抗网络结构是最合适的, 它可以在鉴别器的监督下学习输入点的特征, 然后进行生成。在这里, 本方法使用点云对抗生成网络 PC-GAN[22]作为生成模型。具体来说, 将点云关键点 ϕ 表示成一个 n 维向量 $\boldsymbol{C}=\{c_1,...,c_n,c_i\in\mathbb{R}^3\}$ 输入到生成器 G 中, 得到学习点的分布并生成对抗点。在此, 关键点 ϕ 可以看作是 $p(\boldsymbol{C}|\phi)$ 中的样本。点分布可以被描述为联合似然分布:

$$\mathrm{P}(\boldsymbol{C},\phi)=p(\phi)\prod_{i=1}^{n}p(c_i\,|\,\phi), \tag{1}$$

生成对抗网络的原理是对原始点云和对抗点的分布进行建模。在这里, 为了探究原始点云与对抗点之间的关系, PC-GAN 首先对 ϕ 进行特征学习, 得到更丰富的特征表示 $Q(\boldsymbol{C})$, 再利用 $Q(\boldsymbol{C})$ 学习对抗点的分布。与直接学习对抗点分布相比, 这种方式更关注于原有点云的特征表示, 而不是单纯对点的分布进行学习, 这也是本方法选择 PC-GAN 作为生成模型的重要原因。在这里, 生成对抗网络的损失 L_{GAN} 为:

$$L_{GAN}=\mathbb{E}_{\phi\sim p(\phi)}\left[\begin{array}{l}\min\limits_{G,Q}\max\limits_{f\in\Omega_f}\mathbb{E}_{c\sim p(\boldsymbol{C}|\phi)}\left[f(c)\right]-\\ \mathbb{E}_{z\sim p(z),\boldsymbol{C}\sim p(\boldsymbol{C}|\phi)}\left[f\left(G\left(z,Q(\boldsymbol{C})\right)\right)\right]\end{array}\right], \tag{2}$$

其中, $f(\cdot)$ 是区分生成点和临界点的判别器, Ω_f 是不同概率距离的约束, z 是随机噪声点集 Z 中的点。

3.4 损失函数

3.4.1 误分类损失

上文中描述的生成对抗网络学习框架可以根据原始点云关键点生成对抗点, 但判别器只能区分点的来源, 因此, 还需要一个损失函数来加强对抗点的攻击能力。本文为此提出了误分类损失函数。该函数不指定单个攻击目标, 而是将训练攻击模型时输入的错误分类设置为除正确类别外的任何其他类别。这一设置简化了针对特定类别的训练方式, 实现了无目标攻击的训练方式, 有效地降低了优化难度。具体公式如下:

$$L_{mis}=-\sum_{k=1}^{K}\varsigma(T(\hat{P}))_k\,(1-\delta)\mathrm{l}_{\arg\min T(P)_k}+\delta v_k, \tag{3}$$

其中, ς 代表传统的 softmax 函数, K 是数据集类别的个数, $v=[\frac{1}{K-1},...,0,...,\frac{1}{K-1}]$ 是平滑正则化项。这里, 当 k 不是真实标签索引值时, $v_k=\frac{1}{K-1}$。$\arg\min(\cdot)$ 是为了寻找概率向量中的最小值, 从而明确攻击目标。

3.4.2 用于优化点云质量的感知损失

对抗样本除了具有攻击性外, 还与原始点云具有外观相似性。因此, 本方法必须关注点云质量以确保其外观相似性。对抗样本的目标是生成不引人注意的点或点的集合来欺骗深度网络, 得到错误的结果。在评价点云质量的时候, 不可能评判每个位置的点是否相似, 但根据点云的相似性, 对抗点云在表面上应该和原始点云一样平滑, 也就是说原始点云和对抗点云的法向量在角度方面应该相似。因此, 本方法提出计算对抗样本 $\hat{C}$ 与原始点云 C 的角度相似度来评测生成点云的质量。具体而言, 对于每个点 $a_i\in C$, 都有一个关联的法向量 $\vec{\boldsymbol{n}}_i^a$。对于点 $b_j\in\hat{C}$, 有一个法向量 $\vec{\boldsymbol{n}}_j^b$。假设 a_i 和 b_j 处于相同的位置, 如图 2 所示。

法向量角度相似度可以被定义为 $1-\hat{\theta}/\pi$, 其中 $\hat{\theta}=\arccos(\cos(\theta))$ 表示向量角度。由此, 向量 $\vec{\boldsymbol{n}}_i^a$ 和 $\vec{\boldsymbol{n}}_j^b$ 的余弦相似度计算如下:

$$\mu=\cos(\theta)=\frac{\vec{\boldsymbol{n}}_i^a\cdot\vec{\boldsymbol{n}}_j^b}{\left\|\vec{\boldsymbol{n}}_i^a\right\|\left\|\vec{\boldsymbol{n}}_j^b\right\|}, \tag{4}$$

然后, 利用 μ 计算角度相似度的反余弦值。由于本方法关注切平面的角度相似性, 因此在定义

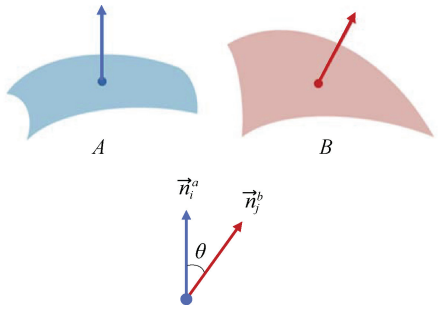

图 2　点云法向量相似度计算示意图

Figure 2　Point cloud normal vector similarity

$\hat{\theta}=\min(\theta,\pi-\theta),\theta\in[0,\pi/2]$ 时，只需要考虑两个角度中较小的一个。因此，两点的角度相似度 s 可以计算为:

$$\hat{\theta}=\arccos(|\mu|),\quad s=1-\frac{2\hat{\theta}}{\pi}. \tag{5}$$

最终的角度相似度 S 应该是每个点的角度相似度。所以需要计算 C 的每个点与 $\hat{C}$ 中最近邻点的角度相似度来生成最终的相似度。计算过程如算法 1 所示。

算法 1.点云相似度计算过程

输入: 原始点云 C 生成的对抗点云 $\hat{C}$。

1　初始化参考点云 $C=c_1,c_2,...,c_n$。

2　**for** $j=1$ to n **do**

　　取 c_k 作为 $\hat{c}_j$ 在 C 中最近邻的点。

　　根据公式(5)计算角度相似度 $s_{\hat{c},c}(j)$。

3　**end for**

4　计算相似度 $\hat{S}_{\hat{C},C}=\sum_{j=1}^{n}s_{\hat{c},c}(j)$。

5　初始化参考点云 $\hat{C}=\hat{c}_1,\hat{c}_2,...,\hat{c}_n$。

6　**for** $i=1$ to n **do**

　　取 $\hat{c}_m$ 作为 c_i 在 $\hat{C}$ 中最近邻的点。

　　根据公式(5)计算角度相似度 $s_{c,\hat{c}}(i)$。

7　**end for**

8　计算相似度 $\hat{S}_{C,\hat{C}}=\sum_{i=1}^{n}s_{c,\hat{c}}(i)$。

9　计算角度相似度 $S=\min(\hat{S}_{\hat{C},C},\hat{S}_{C,\hat{C}})$。

输出: 角度相似度 S。

本方法利用上述点云质量评价方法定义感知损失函数 L_p 如下:

$$L_p=S(C,\hat{C}), \tag{6}$$

其中，S 表示两个点云的角度相似度。最终的网络损失函数如下:

$$L=L_{GAN}+\zeta L_{mis}+\eta(1-L_p), \tag{7}$$

其中，ζ 和 η 用于平衡损失权重。

4　实验结果及分析

4.1　实验设置

4.1.1　数据集设置

本文在 ModelNet10 和 ModelNet40 数据集上开展实验，这两个数据集常用来评估算法分类准确度和检索性能，有很多知名的三维模型算法都在此数据库上进行评测，便于直接进行比较。两个数据集的模型示例如图 3 所示。

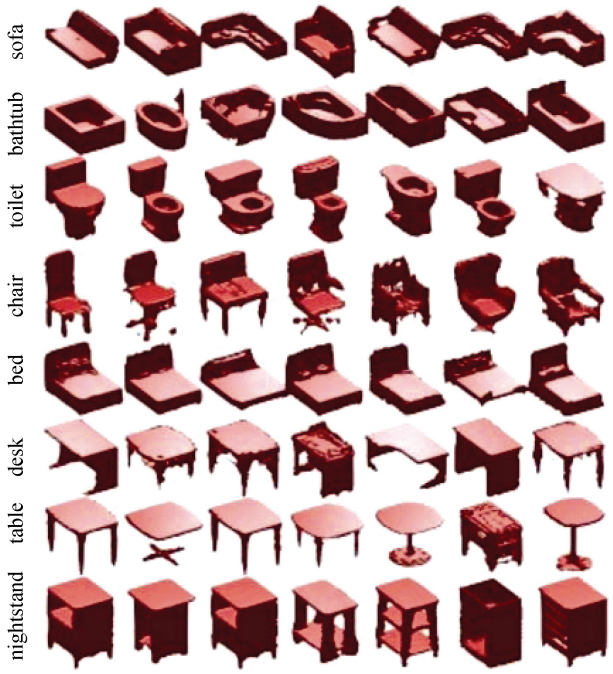

图 3　ModelNet10 和 ModelNet40 数据集模型展示

Figure 3　Examples on ModelNet10 and ModelNet40 datasets

其中, ModelNet10 包含 4899 个 CAD 模型，并分为 10 个类别，与其他数据集不同的是，ModelNet10 中官方给定了训练和测试集的划分，训练集和测试集分别有 3991 和 908 个模型。ModelNet40 包含 12311 个 CAD 模型，包含 40 个类别。ModelNet40 训练集和测试集分别包含 9843 和 2468 个模型。该数据集人工规定了模型分类，手动删除了不属于指定类别的模型。并且 ModelNet10 在模型平移和旋转方面进行了标准化，而 ModelNet40 的模型没有进行标准化。

4.1.2　实验细节设置

本文算法使用预训练的点云特征学习网络来提取点云特征，并使用前文所提出的损失函数来训练生成器 G 和鉴别器 D。用于实验的计算机主要配置

为 NVIDIA 1080Ti GPU 2 张、32 GB RAM 和 Intel Xeon(R) E5-2609 V4 @ 1.70 GHz × 8 CPU。所提出算法使用 Python 语言并基于 PyTorch 深度学习框架进行开发, 整个网络架构以端到端的方式进行了 200 轮训练, 学习率为 10^{-4}, 数据批次大小为 4, 使用 Adam[28]进行参数优化。特别地, 每个三维模型的点云大小为 1024 个点。在实验过程中使用的点云识别网络包括 PointNet[8], PointNet++[9], DGCNN[10]和 RS-CNN[13]。

4.1.3 评价指标

为了客观评价模型的有效性和正确性, 本文采用对抗攻击成功率作为评价指标。该指标描述将生成对抗样本输入被攻击模型后, 模型推理错误样本数占所有输入样本数的百分比。有时也用模型对对抗样本的分类准确率表示该指标, 模型对对抗样本的分类准确率越低说明攻击者的攻击成功率越高。

设最终的分类目标有两类, 分别为正例 (P) 和负例 (N)。真正例 (True positives, TP) 为被正确地划分为正例的个数, 即实际为正例且被分类器划分为正例的实例数; 假正例 (False positives, FP) 为被错误地划分为正例的个数, 即实际为负例但被分类器划分为正例的实例数; 假负例 (False negatives, FN) 为被错误地划分为负例的个数, 即实际为正例但被分类器划分为负例的实例数; 真负例 (True negatives, TN) 为被正确地划分为负例的个数, 即实际为负例且被分类器划分为负例的实例数。所以, 对抗攻击成功率可表示为:

$$\text{Success Rate} = (FP+FN) / (P+N). \tag{8}$$

4.2 对比实验

4.2.1 实验细节设置

生成器是生成攻击点的关键结构。为了证明生成模型的性能, 本文对不同的生成模型进行了测试, 包括 r-GAN[20]、l-GAN[20]、PointGrow[23]和 PC-GAN[22]。实验的结果如表 1 所示。

分析表 1 可知:

1) r-GAN 直接将原始点云作为输入来学习点的特征分布, 但由于点云是无序的, r-GAN 很难学习到共同分布, 因此, r-GAN 表现出最差的性能。

表 1 在 ModelNet10 和 ModelNet40 数据集上使用不同生成模型的攻击成功率对比

Table 1 Comparison of attack success rates using different generative models on ModelNet10 and ModelNet40

生成模型	ModelNet10 /攻击成功率(%)				ModelNet40 /攻击成功率(%)			
	PointNet	PointNet++	DGCNN	RS-CNN	PointNet	PointNet++	DGCNN	RS-CNN
r-GAN	89.39	87.26	86.51	84.93	85.67	83.54	82.92	80.43
l-GAN(CD)	91.47	89.28	88.94	87.67	87.39	82.37	82.14	81.36
l-GAN(EMD)	93.68	92.51	92.16	90.53	90.24	88.34	88.10	86.67
PointGrow	98.65	97.43	97.27	95.44	96.64	95.55	95.17	94.45
PC-GAN	**99.41**	**98.24**	**98.07**	**97.42**	**98.46**	**97.63**	**97.25**	**96.65**

2) l-GAN 通过预训练的自编码器学习点特征分布, 然后通过解码器进行点云生成。根据自编码器表示特征分布时使用的距离度量方法, 可以分为基于倒角距离 (Chamfer Distance, CD) 和基于动土距离 (Earth Mover's Distance, EMD) 两种。与 r-GAN 相比, l-GAN 的特征学习有一定的目标, 输入的减少也简化了学习过程。因此, 与 r-GAN 相比, l-GAN 表现出更好的性能。对于 l-GAN 中使用的不同距离度量方法, CD 优于 EMD, 因为 CD 测量点与点特征之间的距离, 而 EMD 侧重于计算从一种分布更改为另一种分布的最小成本。

3) PointGrow 的提出是为了打破距离度量的限制, 并增加可解释性。PointGrow 在生成点云时考虑了生成形状时点与点之间的相关性, 在生成过程中根据之前的点进行条件分布采样, 反复运行生成样本。因此, PointGrow 优于之前的方法。但是, 它需要一些原始点作为输入, 生成的点与输入点相关联, 且属于同一个分布。因此, 它不适合本文算法。

4) PC-GAN 以原始点云作为输入, 利用分层贝叶斯建模和隐式生成模型生成点云。分层贝叶斯建模在绘制点分布方面起着重要作用, 并有助于学习对抗点的表示。因此, 本方法采用性能最佳的 PC-GAN 作为生成器。

4.2.2 不同数量的关键点对比

关键点的数量是影响攻击成功率的重要参数。仅有几个点的移动导致微小的变化并不能欺骗深度神经网络。但许多点的移动可能会改变点云的形状, 与对抗样本本身的特点不符。本文对不同数量的点进行了相关实验, 实验结果见表 2。

由表 2 可以看出, 随着选择关键点的数量增加, 攻击成功率显着提高。特别地, 当 n 被设置为 384 时, 在 PointNet 上的攻击成功率可以达到 99.41%。这是

表 2 在 ModelNet10 和 ModelNet40 数据集上不同对抗点数量条件下的攻击成功率对比

Table 2 Comparison of attack success rates using different number of adversarial points on ModelNet10 and ModelNet40

n	ModelNet10 / 攻击成功率(%)				ModelNet40 / 攻击成功率(%)			
	PointNet	PointNet++	DGCNN	RS-CNN	PointNet	PointNet++	DGCNN	RS-CNN
128	87.24	86.53	86.22	85.12	86.69	85.54	85.09	83.96
256	94.33	93.87	93.61	91.58	93.36	92.28	91.56	90.84
384	**99.41**	**98.24**	**98.07**	**97.42**	**98.46**	**97.63**	**97.25**	**96.65**
512	98.75	97.83	97.52	97.05	98.16	97.17	96.92	96.33

因为关键点的选择标准是三维模型的全局描述特征, 这会直接影响点云深度模型的性能。但是, 当继续增加关键点的数量时, 攻击性能就会变差。这是因为误分类损失依然存在的情况下, 关注点云质量的感知损失增大, 影响了模型的进一步优化, 导致性能变差。

4.2.3 不同损失函数权重对比

本实验在 ModelNet40 数据集上使用 PointNet 来研究损失权重 ζ 和 η 的影响, 实验结果如表 3 所示。

表 3 在 ModelNet40 数据集上使用不同损失函数权重的攻击成功率对比

Table 3 Comparison of attack success rates using different loss function weights on the ModelNet40 dataset

权重	$\zeta=0.01$	$\zeta=0.1$	$\zeta=1$	$\zeta=10$
$\eta=2$	91.67	94.56	95.87	95.29
$\eta=4$	93.48	96.69	**98.46**	97.87
$\eta=6$	92.62	95.41	96.87	96.03

其中, 两个损失函数作用是相斥的, ζ 控制误分类能力和与错误类别点云的相似度, η 控制对抗样本与原始点云之间的分布相似度。所以一定存在一个平衡状态使算法发挥最大的性能。实验结果发现, 当设置 $\zeta=1$, η=4 时, 这些损失函数可以获得最佳性能。

4.3 消融实验

为了验证本文算法提出的损失函数的有效性, 对不同损失函数的组合进行了实验, 结果如表 4 所示。本实验考虑了三种损失函数的组合。首先, 第一种组合移除了 GAN 损失之外的其他损失函数, 这种方法可以被认为是本方法提出的损失函数的对比基准。可以看到, 这种组合攻击成功率很小, 因为生成点的 GAN 仅仅是为了区分原始点云和对抗样本而设计的, 并没有学习到攻击能力。第二种组合是在 GAN 框架中加入误分类损失, 成功率比原来的 GAN 好很多。这都归功于误分类损失, 因为它与传统攻击损失相比, 放宽了用于优化交叉熵的目标原则。误分类损失选择不正确的类, 而不仅仅是最不可能的类。除此之外, 由于点云的质量对于模拟自然攻击很重要, 所以第三种组合使用感知损失进行了实验。从实验结果中可知, 由于感知损失的影响, 攻击成功率略有下降, 但相对于点云质量而言, 可以忽略。为了证明这一结论, 本文对对抗点云进行了可视化, 如图 4 所示。

其中, 图 4 (a)是原始点云, 图 4 (b)是使用感知损失训练后的对抗样本, 图 4 (c)是未使用感知损失训练后的对抗样本。由此观察到, 在感知损失的监督下生成的对抗样本比没有任何监督生成的要好得多。这一比较结果证明了感知损失对于保留原始外观的重要性。本方法在感知损失中考虑点云和对抗点云法向量的角度相似度。如果每个点的法向量具有相似的角度, 那么这些点共同组成的点云也会相似。由于由原始点组成的网格是完美的, 感知损失使生成的点云与原始点云尽量接近, 所以生成的点云可以获得更好的质量。

表 4 在 ModelNet10 和 ModelNet40 数据集上使用不同损失函数攻击成功率对比

Table 4 Comparison of attack success rates using different loss functions on ModelNet10 and ModelNet40 datasets

损失函数		ModelNet10 / 攻击成功率(%)				ModelNet40 / 攻击成功率(%)			
L_{mis}	L_p	PointNet	PointNet++	DGCNN	RS-CNN	PointNet	PointNet++	DGCNN	RS-CNN
—	—	96.53	95.28	95.04	94.23	93.60	92.88	92.47	91.28
√	—	**99.53**	**98.37**	**98.14**	**97.58**	95.52	**97.85**	**97.58**	**96.89**
√	√	99.41	98.24	98.07	97.42	**98.46**	97.63	97.25	96.65

图 4　对抗点云可视化

Figure 4　Adversarial point cloud visualization

4.4　攻击性能可迁移性实验

可迁移性是对抗样本中常见的评估指标[27]。它指的是基于一种模型生成的对抗样本可以攻击误导另一种不同模型的能力, 是一种常见的黑盒攻击形式。本实验在不同被攻击模型上进行了交叉验证, 实验结果如表 5 所示。本方法考虑了关键点的特性, 并学习了扰动特征, 而不是仅仅参考错误分类目标来生成对抗点。另一方面, 本方法关键点的选择取决于全局特征, 而不是深度模型的缺陷。因此, 本文算法可以获得较强的迁移性能。

此外, 本文与 C&W[6]、IFGM[29]和 LG-GAN[19]的攻击能力可迁移性进行了对比, 实验使用 PointNet 生成的对抗点云作为被攻击网络, 实验结果见表 6。由表可知, 本方法在 PointNet++和 DGCNN 上分别达到了 16.9%和 15.3%的攻击成功率, 与其他方法相比, 在攻击的可迁移性上具备一定的优势。针对各个方法的具体分析将在下一小节中展开。

4.5　与先进方法的对比实验

为了证明本文所提出算法的先进性, 在 ModelNet40 数据集上进行了一系列实验来评估本文算法。本文选择最先进的对抗攻击方法进行比较, 包括 C&W、3D-Adv[6]、IFGM、AdvPC[30]、KNN[17]和 LG-GAN。

此处, 本实验利用从 PointNet 生成的对抗样本来攻击原始的模型、经过 SRS 防御的模型和经过 DUP-Net 防御[31]的模型。实验结果显示在表 7 中, 实验结果表明本文算法(DisGAN)优于其他方法。此外, 相关方法的可迁移性已记录在表 6 的实验结果中。由以上多组数据表中的实验结果分析可知:

1) 攻击能力和可迁移性

C&W 表现最差, 因为它只考虑原始点云中噪点与原始点的距离, 忽略了对抗点的显著特征, 给点云质量带来了影响。因此, 在 DUP-Net 的防御下, 对抗点很容易被去除, 从而导致攻击失败。同时, 它的可迁移性也是最差的。IFGM 参考了 Goodfellow 等人[6]提出的快速梯度下降法(Fast Gradient Sign Method, FGSM), 并进行了改进, 通过增加交叉熵损失来生成对抗样本。

与 C&W 相比, IFGM 侧重于攻击难以被防御方法检测的表面, 并进行调整, 减少了时间成本, 提高了防御下的攻击性能, 因此降低了攻击成功率。然而, 它依然没有考虑扰动特性, 从而影响迁移性能。

AdvPC 利用自编码器结构来学习点云的分布, 然后利用该分布生成对抗样本, 与本文算法的思路

表 5　在 ModelNet10 和 ModelNet40 数据集上基于不同被攻击网络的迁移攻击成功率

Table 5　Success rates of transferred attack based on different victim networks on ModelNet10 and ModelNet40 datasets

方法	ModelNet10(攻击成功率/%)				ModelNet40(攻击成功率/%)			
	PointNet	PointNet++	DGCNN	RS-CNN	PointNet	PointNet++	DGCNN	RS-CNN
PointNet	—	17.6	15.8	13.3	—	16.9	15.3	12.9
PointNet++	20.3	—	16.9	17.6	19.2	—	16.1	16.7
DGCNN	19.4	18.9	—	18.2	18.9	17.5	—	17.3
RS-CNN	19.2	17.3	16.2	—	18.7	16.4	15.4	—

表 6　在 ModelNet40 数据集上以 PointNet 为训练网络的不同方法的迁移攻击成功率对比

Table 6　Comparison of transferred attack success rates for different methods trained on PointNet on ModelNet40 dataset

方法	在 ModelNet40 数据集上以 PointNet 为训练被攻击网络(攻击成功率/%)				
	C&W+l_2	C&W+Chamfer	IFGM	LG-GAN	DisGAN
PointNet (训练网络)	**100**	**100**	73	98.3	98.46
PointNet++(迁移网络)	0	0	3	11.6	**16.9**
DGCNN(迁移网络)	0	0	2.6	14.5	**15.3**

表 7 与先进方法在 ModelNet40 数据集上的攻击成功率、l_2 距离和样本生成时间对比

Table 7 Comparison of attack success rate, l_2 distance, and sample generation time with state-of-the-art methods on the ModelNet40 dataset

方法	无防御/% ↑	SRS 防御/% ↑	DUP-Net 防御/% ↑	l_2 距离/m ↓	时间/s ↓
C&W+l_2	**100**	0	0	**0.01**	40.8
C&W+Chamfer	**100**	0	0	—	43.73
3D-Adv	**100**	70.7	8.5	0.18	21.34
IFGM	73	14.5	3.3	0.31	0.275
LG-GAN	98.3	88.8	84.8	0.35	0.04
AdvPC	97.4	81.4	7.8	0.18	0.137
KNN	99.6	29.2	28.8	0.18	0.249
DisGAN	98.46	**89.6**	**86.3**	0.32	**0.03**

类似。但是自编码器专注于对原始点云分布的学习, 并通过改变分布的方法生成对抗样本。这样直接对分布进行操作会影响部分样本。因此, 它在攻击经过防御的模型方面表现不佳。

KNN 提出了一种新的点云生成模式, 即先生成对抗性三维模型, 再从三维模型中提取点云。这种方法增加了计算复杂度, 但性能并没有大幅提升。

LG-GAN 利用编码器-解码器结构重建点云来生成对抗样本, 扰动被直接添加到解码器当中, 与之前的方法相比获得了显著的改进。但 LG-GAN 的扰动是由标签指定的类别产生的, 而 DisGAN 在生成器中学习扰动, 因此, 本文算法在灵活性、攻击能力和可迁移性方面优于 LG-GAN 以及其他方法。

2) 点云质量

由于传统方法改变点云所有的点以生成对抗点, 破坏了原有的点云结构, 使生成点云质量较低。本方法利用非关键点保留了点云原始结构, 并通过感知损失来监督对抗样本的质量, 对抗点云可视化如图 5 所示。由图可知, 与有目标攻击方法 LG-GAN 相比, 本方法作为无目标攻击方法也能够保持较好的对抗点云质量。

图 5 与先进方法的对抗样本可视化对比

Figure 5 Comparisons of visualized adversarial examples

3) 速度

在计算成本方面, 本方法的优势在于只考虑重构关键点而非对所有点进行运算。表 7 统计了各方法生成对抗样本的平均用时, 其中, 本方法的用时仅为 C&W 的 1/1200、是 IFGM 的 1/8, 对比 LG-GAN, 本方法也有一定的速度提升。由此可见, 本方法在计算成本和成功率方面都优于其他方法。

5 结论

本文针对点云深度神经网络中存在的安全性问题进行了研究, 设计了一种新颖的对抗样本生成方法来研究深度神经网络容易受到攻击的原因。在本文算法中, 所提出的对抗样本生成方法可以在训练过程中学习对抗点的特征, 并生成对抗点来欺骗原始点云深度网络。此外, 本文提出了感知损失来测试对抗样本和原始样本之间的相似性, 监督对抗样本生成过程, 改善点云生成质量, 使其保持原有的形状。最终通过在 ModelNet10 和 ModelNet40 数据集上进行的大量攻击实验证明所生成的对抗样本可以有效地用于攻击点云模型。

本文通过研究基于扰动特征生成对抗样本的方法, 探索深度神经网络容易受到攻击的原因。在本文算法的研究中发现, 产生对抗攻击的重要原因是神经网络学习到的不同类别样本的特征分布边界不清晰, 而攻击算法利用这种模糊的边界生成易混淆的对抗样本, 从而增加了识别模型的识别难度, 降低了识别准确性, 完成了对模型的有效攻击。这为提升神经网络模型的安全性奠定了基础, 在后续的研究中, 我们将探索如何使用对抗样本的扰动特性来对现有的三维模型识别网络进行防御, 从而提升三维模型神经网络的对抗鲁棒性和安全性。

参考文献

[1] Lin A A, Li T B, Wang X W, et al. Review of 3D Model Retrieval Algorithms Based on Deep Learning[J]. Journal of Data Acquisition and Processing, 2021, 36(1): 1-21.

(刘安安, 李天宝, 王晓雯, 等. 基于深度学习的三维模型检索算法综述[J]. 数据采集与处理, 2021, 36(1): 1-21.)

[2] Zhang Y W, Hu K, Wang P S. Review of 3D Reconstruction Algorithms[J]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition), 2020, 12(5): 591-602.
(张彦雯, 胡凯, 王鹏盛. 三维重建算法研究综述[J]. 南京信息工程大学学报(自然科学版), 2020, 12(5): 591-602.)

[3] Jing Y Q, Wang Y H, Han W, et al. Overview of SLAM Methods for Unmanned Vehicles and Mobile Robots[J]. Electronics World, 2021(13): 4-5.
(景元泉, 王跃辉, 韩伟, 等. 无人驾驶车辆与移动机器人 SLAM 方法综述[J]. 电子世界, 2021(13): 4-5.)

[4] Zhang F J, Dai G Z, Peng X L. A Survey on Human-Computer Interaction in Virtual Reality[J]. Scientia Sinica (Informationis), 2016, 46(12): 1711-1736.
(张凤军, 戴国忠, 彭晓兰. 虚拟现实的人机交互综述[J]. 中国科学: 信息科学, 2016, 46(12): 1711-1736.)

[5] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing Properties of Neural Networks[EB/OL]. 2013: 1312.6199. https://arxiv.org/abs/1312.6199v4.

[6] Goodfellow I J, Shlens J, Szegedy C, et al. Explaining and Harnessing Adversarial Examples[EB/OL]. 2014: 1412.6572. https://arxiv.org/abs/1412.6572v3.

[7] Xiang C, Qi C R, Li B. Generating 3D Adversarial Point Clouds[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 9128-9136.

[8] Charles R Q, Hao S, Mo K C, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017: 77-85.

[9] Qi C R, Yi L, Su H, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space[EB/OL]. 2017: 1706.02413. https://arxiv.org/abs/1706.02413v1.

[10] Wang Y, Sun Y B, Liu Z W, et al. Dynamic Graph CNN for Learning on Point Clouds[J]. ACM Transactions on Graphics, 2019, 38(5): 1-12.

[11] Klokov R, Lempitsky V. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models[C]. 2017 IEEE International Conference on Computer Vision, 2017: 863-872.

[12] Wang R, Hu P, Jiang J H. Research on Point Cloud Index Technology Based on KD Tree[J]. Science & Technology Vision, 2015(30): 111.
(王儒, 胡萍, 蒋俊豪. 基于 KD 树的点云索引技术研究[J]. 科技视界, 2015(30): 111.)

[13] Liu Y C, Fan B, Xiang S M, et al. Relation-Shape Convolutional Neural Network for Point Cloud Analysis[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 8887-8896.

[14] Kumawat S, Raman S. LP-3DCNN: Unveiling Local Phase in 3D Convolutional Neural Networks[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 4898-4907.

[15] Zheng T H, Chen C Y, Yuan J S, et al. PointCloud Saliency Maps[C]. 2019 IEEE/CVF International Conference on Computer Vision, 2019: 1598-1606.

[16] Wicker M, Kwiatkowska M. Robustness of 3D Deep Learning in an Adversarial Setting[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 11759-11767.

[17] Tsai T, Yang K C, Ho T Y, et al. Robust Adversarial Objects Against Deep Learning Models[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1): 954-962.

[18] Zhao Y, Wu Y W, Chen C H, et al. On Isometry Robustness of Deep 3D Point Cloud Models under Adversarial Attacks[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 1198-1207.

[19] Zhou H, Chen D D, Liao J, et al. LG-GAN: Label Guided Adversarial Network for Flexible Targeted Attack of Point Cloud Based Deep Networks[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 10353-10362.

[20] Achlioptas P, Diamanti O, Mitliagkas I, et al. Learning Representations and Generative Models for 3D Point Clouds[EB/OL]. 2017: 1707.02392. https://arxiv.org/abs/1707.02392v3.

[21] Gadelha M, Wang R, Maji S. Multiresolution Tree Networks for 3D Point Cloud Processing[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 105-122.

[22] Li C L, Zaheer M, Zhang Y, et al. Point Cloud GAN[EB/OL]. 2018: 1810.05795. https://arxiv.org/abs/1810.05795v1.

[23] Sun Y B, Wang Y, Liu Z W, et al. PointGrow: Autoregressively Learned Point Cloud Generation with Self-Attention[C]. 2020 IEEE Winter Conference on Applications of Computer Vision, 2020: 61-70.

[24] Zamorski M, Zięba M, Klukowski P, et al. Adversarial Autoencoders for Compact Representations of 3D Point Clouds[J]. Computer Vision and Image Understanding, 2020, 193: 102921.

[25] Yang G D, Huang X, Hao Z K, et al. PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows[C]. 2019 IEEE/CVF International Conference on Computer Vision, 2019: 4540-4549.

[26] Hartigan J A, Wong M A. Algorithm AS 136: A K-Means Clustering Algorithm[J]. Applied Statistics, 1979, 28(1): 100.

[27] Ilyas A, Santurkar S, Tsipras D, et al. Adversarial Examples Are Not Bugs, they Are Features[EB/OL]. 2019: 1905.02175. https://arxiv.org/abs/1905.02175v4.

[28] Kingma D P, Ba J, Hammad M M. Adam: A Method for Stochastic Optimization[EB/OL]. 2014: 1412.6980. https://arxiv.org/abs/1412.6980v9.

[29] Liu D, Yu R, Su H. Extending Adversarial Attacks and Defenses to Deep 3D Point Cloud Classifiers[C]. 2019 IEEE International Conference on Image Processing, 2019: 2279-2283.

[30] Hamdi A, Rojas S, Thabet A, et al. AdvPC: Transferable Adversarial Perturbations on 3D Point Clouds[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 241-257.

[31] Zhou H, Chen K J, Zhang W M, et al. DUP-Net: Denoiser and Upsampler Network for 3D Adversarial Point Clouds Defense[C]. 2019 IEEE/CVF International Conference on Computer Vision, 2019: 1961-1970.

刘佳 硕士。现任天津中德应用技术大学软件与通信学院讲师。研究领域为人工智能、物联网。Email: liujia@tsguas.edu.cn

金志刚 博士。现任天津大学电气自动化与信息工程学院教授。研究领域为无线网络与网络安全、水下通信与网络。 Email: zgjin@tju.edu.cn

金诗博 硕士。现任天津中德应用技术大学软件与通信学院讲师。研究领域为人工智能。Email: 15122148909@126.com