Semantic Adversarial Robustness with Differentiable Ray-Tracing

Rahul Venkatesh
Carnegie Mellon University
`rmvenkat@andrew.cmu.edu`

Eric Wong
Massachusetts Institute of Technology
`wongeric@mit.edu`

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI
`zkolter@cs.cmu.edu`

Abstract

Several works have shown that deep learning models are vulnerable to adversarial examples crafted by adding visually imperceptible noise to an image. However, these perturbations lack semantic meaning and have limited real world significance. Recently, a number of methods have been proposed for crafting semantic adversarial examples by leveraging techniques from computer graphics such as differentiable ray-tracing and image based rendering to simulate real world conditions. However, it is unclear whether algorithms can be designed to create models that are robust to such attacks. This poses a serious threat to the security guarantee of deep learning systems. In this work, we investigate whether techniques such as adversarial training that have been successful in creating models that are robust to image space attacks, can be used in conjunction with differentiable ray-tracing to defend against semantic adversarial attacks. Additionally, to the best of our knowledge there is no benchmark established for this task, so we create a simple 3D traffic-sign dataset to simulate attacks in the 3D domain. This dataset is meant to serve as a benchmark for future work. Finally, our results confirm that there is only a marginal benefit to using gradient based methods to defend against these attacks, demonstrating the importance of further study on this problem setting.

1 Introduction

The existence of adversarial examples poses serious security concerns for deep learning based systems [1]. A number of techniques have been proposed for both adversarial attacks [2, 3, 4] and defenses [5, 6, 7] in the 2D pixel space. However, such perturbations do not have semantic meaning, and scarcely occur in the real world [8, 9]. Therefore, recently there has been some work on producing semantic adversarial examples by using image based rendering [10] (e.g. by corrupting the albedo of the image), perturbing 3D scene parameters like lighting, camera location, mesh vertices [11, 12, 13, 14] or by introducing spatial changes such as translations and rotations to 2D images [15]. However, little has been done to learn models which are robust to these attacks. Therefore, in this work, our main goal is to learn models which are robust to meaningful semantic changes. Towards this goal, we parameterize semantic changes in the 3D scene space and employ a differentiable ray-tracer [11] to simulate real world conditions. This allows us to generate adversarial examples in the 3D space, which can then be used in standard algorithms to learn robust models.

Note that crafting an attack by perturbing 3D scene parameters, requires access to the 3D model/scene [11]. To the best of our knowledge, there is no well-defined benchmark for evaluating robust models in the 3D scene parameter space. Therefore, we create a simple dataset of 3D traffic signs (Fig. 1). Note that since synthetic traffic signs resemble the real world, this dataset has direct applicability for real world tasks such as autonomous driving. Moreover, we make use of a differentiable ray-tracer to simulate real world conditions. Therefore, this is a well motivated setting for adversarial robustness.

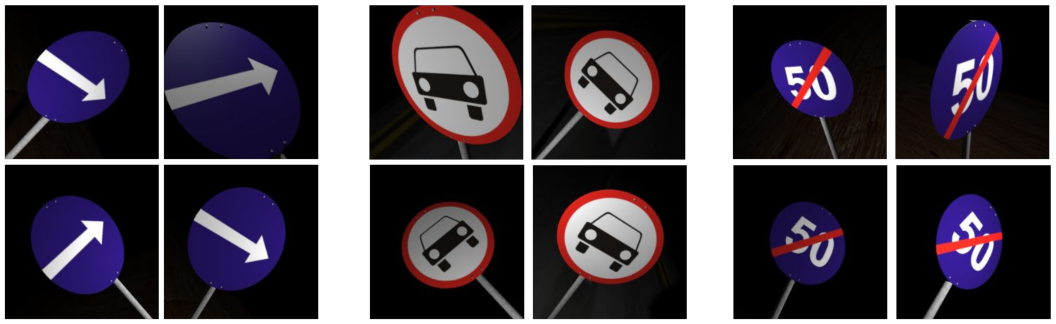


Figure 1: Exemplar ray-traced images from our dataset.

Thus, the contributions of this paper are as follows,

- We create a dataset of 3D traffic signs for benchmarking robustness to simple scene changes.
- We perform robust training in the 3D scene parameter space using PGD adversaries [16].
- Our results confirm that PGD adversaries marginally improve robustness to scene changes. However, our robust model obtains low accuracy on test-set adversaries generated using a local grid search[1], verifying that PGD cannot reliably find the worst case adversary.
- We introduce a novel strategy to generate stronger adversaries in the 3D scene parameter space by combining the local grid search with PGD.

2 Background

Pixel Space Adversarial Attacks. A standard method for computing adversarial examples for an image, x with class label, y is by projected gradient descent on the negative loss function [16],

$$x^{(k+1)} = \Pi_{x+\mathcal{S}}(x^{(k)} + \alpha(\nabla_x \mathcal{L}(\mathcal{H}, x^{(k)}, y)) \tag{1}$$

Here α is the learning rate, $\mathcal{S} \subseteq \mathbb{R}^d$ is the set of allowable perturbations to x and $\mathcal{L}$ refers to a loss function applied to predictions from classifier, $\mathcal{H}$. $\Pi_{x+\mathcal{S}}$ is a projection onto an allowable set.

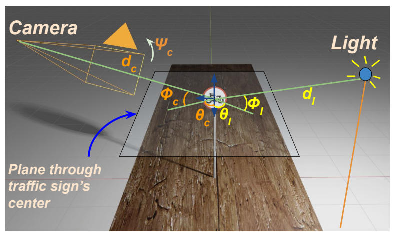
2D Spatial Attacks. These refer to a more natural/semantic class of perturbations such as 2D spatial translations and rotations. Works such as [15] show that standard first order techniques such as PGD [16] cannot be used to reliably find worst case perturbations in this setting. Given the low dimensional parameter space, adversaries are found by randomly sampling 10 transformations from the space of allowable perturbations. However, such methods are not guaranteed to work with a complex high dimensional search space such as 3D scene parameters. Moreover, these 2D transformations still do not accurately resemble real world conditions.

Adversarial Attacks on 3D scene parameters. Recently, a number of methods have been proposed to craft semantic adversarial attacks by perturbing 3D scene parameters such as camera position [17, 14], mesh geometry and light position [12]. However, most of these works consider a large range for each degree of freedom, and use complex geometrical objects which results in unnatural adversarial examples [17] that do not necessarily reflect what happens in the real world. For these reasons, the experimental setup makes it hard to quantify robustness with respect to such attacks. Therefore, in this work, we consider a simplified scenario of 3D traffic signs, constrain the scene parameters to match natural conditions, and employ a differentiable ray-tracer [11] to simulate the real world image generation process. Since traffic sign detection has clear applications in tasks like autonomous driving, this is a well motivated setting for evaluating adversarial robustness.

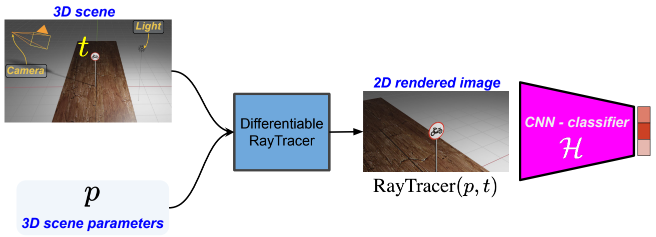
3 Adversarial Attack and Defense on 3D Scene Parameters

In this section, we first describe our dataset creation pipeline, followed by adversarial attacks on 3D scene parameters and a discussion on adversarial training pipelines to defend against these attacks. We end with a description of evaluation strategies to quantify robustness.

[1]For each sample, we create a uniform grid over the range of allowable perturbations



(a) 3D scene with scene parameters.



(b) Model Architecture.

Figure 2: ***Left:*** An illustration of 3D scene parameters viz. Camera and Light. The classifier takes in as input a ray-traced image of the 3D traffic sign captured by a camera at an initial distance d_c from the traffic sign. Additionally, the scene is illuminated by a light at distance d_l. The relative rotations of the camera and light can be uniquely described using a combination of three angles: azimuth (θ), elevation (ϕ) and tilt/in-plane rotation (ψ). ***Right:*** Overall Pipeline.

Dataset Creation. We downloaded a traffic sign 3D mesh model pack (containing 56 different traffic signs) from an open source website[2] and used *Blender* to create the 3D scene. This allows us to simulate various scene changes such as lighting effects, camera viewpoint, etc. (See Fig. 1) Note that this pipeline can be enhanced by adding in a sky, environment lighting, and other objects to the scene like occluders, etc, but we chose to start with a simple dataset. Our dataset is formally defined as,

$$\mathcal{X} = \{(\text{RayTracer}(p^j, t), y_t) : p^j \sim \mathcal{P}, t \in \mathcal{T}, 1 \leq j \leq N\} \tag{2}$$

Here $\mathcal{P} \subseteq \mathbb{R}^7$ is the allowed camera and light configurations in the dataset (Refer Table. 1), $\mathcal{T}$ is the set of traffic signs, and N is the dataset size. A subset, $\mathcal{X}_{train} \subset \mathcal{X}$ is set aside for training.

Algorithm 1: Robust Model Training

```
require: A training set $\mathcal{X} = \{(\text{RayTracer}(p^j, t), y_t) : p^j \sim \mathcal{P}, t \in \mathcal{T}\}$

/*Pretraining the classifier on $\mathcal{X}$*/
for iter in $n_{pret}$ do
    Sample a minibatch $\mathcal{X}_s \in \mathcal{X}$
    Train $\mathcal{H}$ on minibatch $\mathcal{X}_s$
end

/*PGD-Adversarial Training*/
for iter in $n_{iters}$ do
    Sample a minibatch $\mathcal{X}_s \in \mathcal{X}$
    /*Modify $\mathcal{X}_s$*/
    for $RayTracer(p^j, t), y_t$ in $\mathcal{X}_s$ do
        Randomly perturb $p^j$
        $n_{pgd}^{train}$ step PGD on $p^j$ (Eqn. 3).
    end
    /*Train*/
    Train $\mathcal{H}$ on minibatch $\mathcal{X}_s$
end
```

3D Threat model. The standard notion of a pixel space adversarial attack can be extended to the domain of 3D scene parameters such as camera and light position. Note that these perturbations which we aim to find are in the 3D domain, while our classifier takes in a ray-traced 2D image as input. Therefore, we use a differentiable ray-tracer to backpropagate gradients from the classifier to 3D scene parameters. We consider four degrees of freedom for a perturbation to the camera position, and three for the light position[3] (See Fig. 2a for an illustration of these parameters). This gives us a set of 7 scene parameters which we collectively refer to using the vector p. More concretely, we aim to perturb p to craft an adversarial attack. We perform projected gradient descent on the negative loss function to discover the adversary,

$$p^{(k+1)} = \Pi_{p+\mathcal{S}}(p^{(k)} + \alpha(\nabla_p \mathcal{L}(\mathcal{H}, \text{RayTracer}(p^{(k)}, t), y_t)) \tag{3}$$

Here, $\mathcal{H}$ refers to the classifier parameters, $t \in \mathcal{T}$ is the 3D traffic sign scene with scene parameters specified by p, and y_t signifies the class label associated with traffic sign t. α is the learning rate, which can be set per-parameter and $\mathcal{S} \subseteq \mathbb{R}^7$ is the set of allowed perturbations.

Adversarial Training using 3D Threat model. We craft adversarial examples using this definition of a 3D threat model, and use these samples to perform adversarial training (See Fig. 2b). Note that we initialize each adversarial attack to uniformly within $\mathcal{S}$ (See Algorithm. 1).

Evaluation Strategy. We evaluate the test set accuracy using four methods. In the first method (PGD), for a given traffic sign t starting at scene parameter p^j, we run the PGD without the random

[2]turbosquid.com

[3]Note that, we omit the tilt parameter for the light, as it doesn't significantly alter the ray-traced image.

p_i	min_{p_i}	max_{p_i}
Cam. Azimuth	-60	60
Cam. Elevation	-30	30
Cam. Tilt	-45	45
Cam. Distance	0.3	0.6
Lgt. Azimuth	-60	60
Lgt. Elevation	-70	70
Lgt. Distance	0.3	0.7

Test Accuracy				
Model	PGD	PGD + RS	GS	GS+PGD
Data Aug.	0.78	0.72	**0.42**	0.27
Robust	**0.89**	**0.81**	0.40	**0.33**
Hyperparameters				
n_{starts}^{test}	1	8	-	-
n_{pgd}^{test}	50	50	0	5

Table 1: ***Left:*** Sampling limits of different scene parameters (angles: degrees, distance: meters). Refer Fig. 2a for a visualization of these parameters. ***Right:*** Comparison between *Data Augmentation* and the *Robust* model for different number of random starts, using PGD and Local Grid Search (GS).

initialization, whereas the second method (PGD+RS) runs the PGD iterations after a random start in the allowable perturbation range (same as Algorithm 1). For each test sample, we perform n_{starts}^{test} restarts and select the worst case accuracy. The third method is grid search (GS), where we sample a uniform grid of size n_{grid}^7 in the range $[p^j - \epsilon, p^j + \epsilon]$, and select the worst case sample[4]. The fourth method (GS+PGD) is grid search with an additional step of PGD iterations at each grid location. As the range is now quantized, a lower value of $\alpha = \epsilon/(n_{grid} * n_{pgd}^{test})$ is sufficient to explore the entire parameter space. Note that the perturbation strength ϵ can be set per scene parameter.

4 Experiments

We start with a dataset that has 200 ray-traced images per traffic sign. For creating the test set, we set aside one scene configuration for each traffic sign, which gives us 56 test samples. The remaining data is split into a train set (90%), and a validation set (10%). As a baseline, we simply skip the iterative update to p^j in Algorithm 1, and retain the random start. We call this the *"Data-Augmentation"* model, and the model trained with the PGD iterations is called *"Robust"* model. The set of allowed scene configurations, $\mathcal{P}$ are reported in Table 1, and we set the perturbation strength, ϵ to 20% of the total range. This means that the allowable perturbations are in the range $[-\epsilon, +\epsilon]$ where $\epsilon = 0.2 * (max_p - min_p)$. Note we set $\alpha = \epsilon/n_{pgd}^{train}$ where $n_{pgd}^{train} = 10$. The architecture of the classifier $\mathcal{H}$, comprises of the first 6 layers of a pretrained VGG16 [18], followed by a 3x3 conv and two FC layers of dimensions 256 and 128. We use ReLU activations between the layers.

We draw some interesting observations from this experiment. First, we find that the *Robust* model performs better than *Data Augmentation* when the test strategy is PGD or PGD+RS. Note that the test accuracies are reasonably high for these two evaluation strategies. However, when we evaluate the models using grid search (GS), we immediately find a significant drop in accuracy, confirming that PGD adversaries are insufficient for measuring robustness in this setting. Additionally, it is interesting to note that when we use grid search augmented with PGD iterations as the test strategy, the accuracy of the *Robust* model is slightly higher than *Data Augmentation*. This is primarily due to the fact that GS has a limited sampling capacity and doesn't approximate the parameter space as well as (GS+PGD). Since *Data Augmentation* is bound to perform better on average case (non-adversarial) examples, it is not surprising that when evaluated just with GS, it has a higher accuracy when compared to the *Robust* model. We immediately find that, when the grid search is augmented with PGD iterations (GS+PGD), the *Robust* model outperforms *Data Augmentation* as it better models the worst case adversary.

5 Conclusion

Our results indicate that even on a simple traffic signs dataset, both the *Robust* model trained with PGD and the *Data Augmentation* model yield low test set-accuracy using grid search. Thus, it is clear that although standard first order methods such as PGD have some marginal benefit, they fail to reliably find the worst case adversary. Therefore, new techniques are required to work in the highly discontinuous and non-convex space of 3D scene parameters. This inspires a new direction of study in the area of adversarial robustness.

[4]For each parameter, we consider 4 points in it's range, which gives us $4^7 = 16384$ scene parameter vectors

References

[1] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[2] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[3] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018.

[4] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018.

[5] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*, 2018.

[6] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[7] BS Vivek, Konda Reddy Mopuri, and R Venkatesh Babu. Gray-box adversarial training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 203–218, 2018.

[8] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[10] Lakshya Jain, Wilson Wu, Steven Chen, Uyeong Jang, Varun Chandrasekaran, Sanjit Seshia, and Somesh Jha. Generating semantic adversarial examples with differentiable rendering. *arXiv preprint arXiv:1910.00727*, 2019.

[11] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

[12] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019.

[13] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4302–4311, 2019.

[14] Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1201–1210, 2020.

[15] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[17] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4845–4854, 2019.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.