Project Report
LSP-236

Learning Robust Representations for Automatic Target Recognition: FY18 Line-Supported Information, Computation & Exploitation Program

J.A. Goodwin
O.M. Brown
T.W. Killian
S.-H. Son

13 November 2018

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001.

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering

Massachusetts Institute of Technology
Lincoln Laboratory

Learning Robust Representations for Automatic Target Recognition: FY18 Line-Supported Information, Computation & Exploitation Program

J.A. Goodwin
O.M. Brown
T.W. Killian
S.-H. Son

Group 36

Project Report LSP-236

13 November 2018

Lexington Massachusetts

ABSTRACT

Machine Learning (ML) has an increasing role within many mission areas across the Laboratory. Yet, it remains to be seen how robust and secure these algorithms are to inputs that are intentionally designed to cause a ML model to make a mistake, i.e., adversarial examples. Many potential issues arise from the existence of adversarial examples. Examples include an adversary biasing the training data via data poisoning techniques for Automatic Target Recognition (ATR) systems or attacking cybersecurity systems by inserting malicious content that appears legitimate. Building a framework to evaluate and build robustness into ML algorithms will become increasingly important as the USG invests in new capabilities.

The unique set of mission areas across the Laboratory offers a set of challenging problems for generating attacks; namely, the adversary is often limited in its ability to fully understand the defenses capability. Yet, the adversary may have a good understanding of the training data used for any given ML model; e.g., for ATR radar systems such as those needed for Ballistic Missile Defense, the adversary controls what the defense observes when conducting system capability tests leading to the possibility of potential data poisoning attacks. This project aims to adapt and expand on existing approaches to effectively red team adversarial attacks for evaluating the robustness of Laboratory ML algorithms, while also developing techniques to build resiliency into the models.

Radio frequency (RF) sensors are used alongside other sensing modalities to provide rich representations of the world. Given the high variability of complex-valued target responses, RF systems are susceptible to attacks masking true target characteristics from accurate identification. In this work, we evaluate different techniques for building robust classification architectures exploiting learned physical structure in received synthetic aperture radar signals of simulated 3D targets.

ACKNOWLEDGMENTS

We would like to thank the Paul Monticciolo, Vijay Gadepally and the ICE Line Committee for their belief in and support of this program. We have been encouraged to push boundaries and focus on the big picture outcomes of our work. We are grateful that our vision is unconstrained as we try to expand the Laboratory's capability to address low-resource settings with data-driven approaches.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Active sensors (i.e., radar) can provide autonomous systems with a rich representation of the physical world, which can be used to augment the information collected from traditional static sensors (i.e., cameras). As a radio frequency sensor, radar offers unique capabilities to accurately measure physical attributes that other sensors cannot, such as range to target, radial velocity, and other physical characteristics [1]. Radar can be used to help with scene characterization and automatic target recognition (ATR) to classify different detected targets (e.g., cars, pedestrians, obstacles) in the presence of different types of clutter (buildings, trees, other noise sources).

ATR using Synthetic Aperture Radar (SAR) is a common radar application for classifying targets using a sensor mounted on moving vehicles such as aircraft and automobiles. ATR has long been performed with handcrafted features [2], an approach that has begun to give way to data-driven approaches following the success of deep learning architectures in image classification [3–5]. However, recent applications of these techniques to radar problems do not explicitly account for the rich physical properties of the signals provided for classification [6–8]. Using simulated radio wave interactions with 3D targets, we train a model that approximates the relationship between the radar signal response and underlying target class.

Machine learning models have been shown to be vulnerable to adversarial attacks in which inputs to the model are purposely manipulated in order to produce erroneous results [9]. In response, numerous methods have been proposed for generating attacks, building defenses, and measuring the robustness of algorithms to adversarial perturbations [10–13]. SAR systems may also be susceptible to attack given the high variability of possible complex-valued signatures for a given target. This variability results from a number of factors, including diverse environments, sensor parameters, viewing geometries, clutter, target shapes and materials, all of which impact the signal returned to the radar. This variability is difficult to model, and hence difficult to incorporate into the training data for a given radar system. Further, this high variability of possible target signatures leaves opportunity for radar systems to be fooled by an adversary.

In this work, we evaluate a suite of techniques for building ATR architectures that intend to be robust to adversarial attacks. The techniques we consider include conditional training based on target pose estimation, feature similarity embedding, and adversarial learning by perturbing the complex-valued target response before processing the image. We evaluate these techniques using physics-based simulations of SAR images for a target shape classification problem, and demonstrate their ability to increase the robustness (and accuracy) of our radar classifier.

2. PRELIMINARIES

2.1 NEURAL NETWORKS

Neural networks, most notably those used in applications of Deep Learning, have emerged as effective near-universal approximators of complex systems and functions. The primary unit of a neural network is called a *neuron*. A neuron, or *node*, can be described as a non-linear mapping over a weighted linear combination of other nodes or some external input. As such, the output of a collection of nodes, described as a *layer* in the network, can be represented as

$$\tilde{x}^{(k+1)} = \sigma\left(\tilde{W}^{(k)} \cdot \tilde{x}^{(k)} + w_b^{(k)}\right) \tag{1}$$

where $\tilde{x}^{(k)} \in \mathbb{R}^n$ is the input vector, $\tilde{W}^{(k)} \in \mathbb{R}^{m \times n}$ is the weight matrix and $w_b^{(k)}$ is the weight bias of the k^{th} layer of the network. $\sigma(\cdot)$ is an element-wise nonlinear mapping, often called an *activation function* that controls for translational invariance as well as introducing the capability of modeling more complex behavior. The output vector $\tilde{x}^{(k+1)} \in \mathbb{R}^n$ is then used as input to the next layer in the network. The collection of all weight matrices and biases across the layers of a network are known as the *parameters* of the network and may be represented by the variable Θ.

For notation purposes, we subsume the biases $w_b^{(\cdot)}$ into the weight matrices and augment the input x by adding a 1 as an additional dimension. Thus the parameters of the layer k can be compactly represented by the matrix $W^{(k)}$. The final output of an N-layer neural network can be formulated as

$$\hat{y} = f_\Theta(x) = \sigma\left(W^{(N)} \cdot \sigma\left(W^{(N-1)} \cdots \sigma\left(W^{(0)} \cdot x\right)\right)\right) \tag{2}$$

The network parameters Θ are "learned" through optimization to approximate an unknown function $f : \mathbb{R}^{n_I} \to \mathbb{R}^{n_O}$. This is done by a process called *training* in which the calculated state of the output $\hat{y}$ is compared to target values, y, corresponding to the datum as input to the network, x. The total deviation from the target value is termed as the *error* of the neural network, the signal of which is then used to update Θ. The standard algorithm for propagating the error through the network is known as *backpropagation* [14].

Based on the application there are various approaches to measure this error, known as the *loss function*. In practice, the loss function and hyperparameters used by the backpropagation algorithm are the most important aspects, followed by the structure of the nodes and edges that make up the computation graph, of training an accurate neural network and learning a good representation of the collected data. Depending on the complexity of the defined neural network, difficultly of learning the representation of the data, the size of the data set used to train the neural network, etc. influences the number of iterations needed for the backpropagation algorithm to optimize the parameters Θ appropriately.

2.2 ROBUSTNESS AND ADVERSARIAL PERTURBATIONS

The goal of any machine learning task is to train the algorithm to perform well on data in the training set, while maintaining performance on data it has not seen before (i.e., maintain high

in-sample and out-of-sample accuracy). This property is referred to as generalization [15]. Out-of-sample performance can be estimated by using a separate test set that is withheld from the training process, or through performing cross-validation of the training set.

A second property that is desired of a machine learning algorithm is that it will maintain its performance given small perturbations to its inputs. It is this property that has been found to be violated in many instances by adversarial perturbations [9, 16]. An adversarial perturbation for a correctly classified input, x, is a small perturbation, r satisfying $\|r\| < \epsilon$, that when applied to the input results in an incorrect classification decision, i.e., $f_\Theta(x + r) \neq f_\Theta(x)$ (see Figure 1). A classifier for which adversarial perturbations exist for many of its examples is not considered robust since the model can easily be fooled by small changes to the inputs.

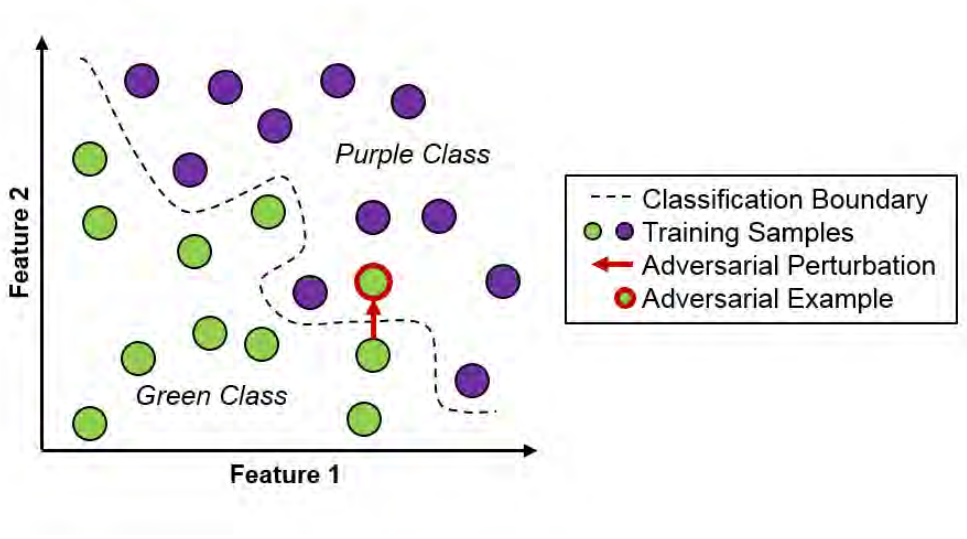

Figure 1. Notional classifier with adversarial perturbation (red arrow) applied to one of the green samples.

Once a classifier has been trained, an adversarial perturbation for a given input x can be found by using a gradient-based optimization procedure to search for minimum perturbations to the input that maximize the loss function. Methods range from quick approximations that take only a single gradient step such as Fast Gradient Sign Method (FGSM) [10], to solving the full optimization problem as in [17]. While FGSM is not guaranteed to find an adversarial example, solving the full optimization problem is nontrivial, so in [11] the authors propose an alternative method, called DeepFool, that iteratively projects the input onto the decision boundary of the locally-linearized classification model. Once adversarial perturbations have been found, they can be used in a robustness metric or to create additional training examples for the classifier. Training on adversarial examples, referred to as *adversarial learning*, has been shown to increase robustness [10].

In general, there is a trade-off between accuracy and robustness for any given classifier. Optimization techniques used during model training tend to craft highly complex decision boundaries in an attempt to precisely differentiate between each class. Such precision leads to greater accu-

racy yet introduces some deficiencies as those final, tough to classify data points now lie close to the decision boundary, only needing to be "nudged" slightly in order to be misclassified. In this manner, a highly accurate model may not be robust. By optimizing for robustness, any decision boundary learned to separate the classes is effectually kept from becoming too precise. While less accurate, a robust model is likely far more reliable in execution.

3. AUTOMATIC TARGET RECOGNITION WITH SYNTHETIC APERTURE RADAR

We focus on data derived from simulating an airborne SAR, which produces a high-resolution representation of the scene in range and cross-range [1]. Similar to the data set generated in [18], we consider the SAR scenario illustrated in Figure 2 with an example input image for an hourglass-shaped target. The target shapes we consider are described in detail Section 4.2. The aircraft flies in a circular orbit around the target of interest while sending Linear Frequency Modulated (LFM) pulses to the target, and collecting the received backscattered pulses. A SAR image is generated from the complex-valued frequency history of a given orbital segment using back-projection around the target (i.e., spotlight extraction). The goal of the ATR classifier is to determine the target class given the normalized magnitude of the image.

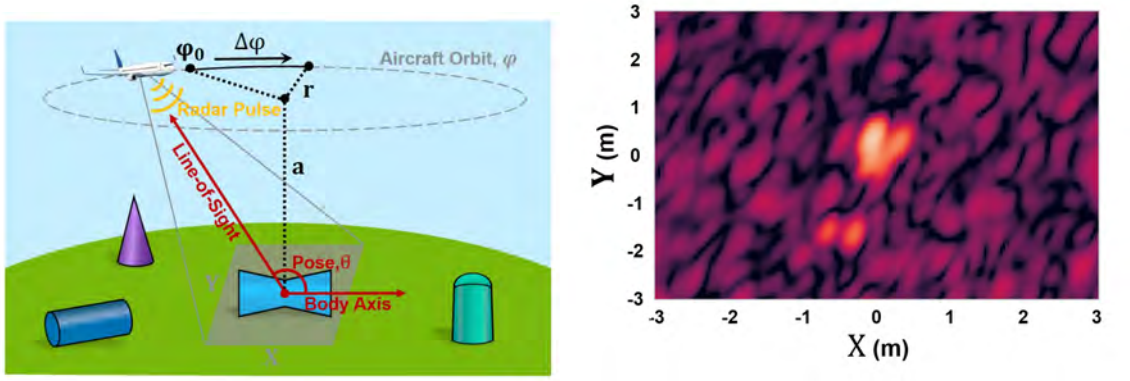

Figure 2. Notional ATR with SAR scenario (top) with simulated SAR image for an hourglass target (bottom)

3.1 OVERVIEW OF SAR

The principle behind SAR is is to use a traditional mono-static radar with a LFM pulse that provides high range resolution and utilize the motion path of the host platform to produce an "simulated" large aperture that can also provide high cross-range resolution. Without the motion path the angular resolution of the mono-static processed data will be coarse. For a SAR platform following a motion path, and observing a stationary target, the antenna phase center is defined as,

$$X_p = [x_p, y_p, z_p] \tag{3}$$

where there are $p = 1, \ldots, N_p$ collects across the across the synthetic aperture. The distance to the radar phase center is then given by

$$R_p = \sqrt{(x_p - x_0)^2 + (y_p - y_0)^2 + (z_p - z_0)^2} \tag{4}$$

where the position of the stationary target is $X_0 = [x_0, y_0, z_0]$ defined to be the geometric center of the target shape. The output of the receiver at time t_p is a sequence of frequency samples

delayed by the round trip time between the transmitted signal and the back-scattered response of the target. There are K frequency samples per received signal denoted by f_k. The received signal for each sequence can formulated as

$$E_{k,p} = E_T(f_k) \exp\left(-i4\pi f_k R_p / c\right) \tag{5}$$

where E_T is the complex response for target T and R_p is the distance to the phase center defined above. There exists a number of techniques to convert these complex valued frequency histories into a 2D image. We utilize the back-projection algorithm described in [19].

3.2 TARGET CLASSIFICATION ARCHITECTURE

Given a set of complex valued frequency histories across a synthetic aperture illuminating a single target, $S = \{E_{k,p} | k = 1, \ldots, K \text{ and } p = 1, \ldots, N_p\}$, process the collection through a signal processing function, $g : \mathbb{R}^{K \times N_p} \rightarrow \mathbb{R}^{N \times N}$, that to produce a 2D real valued image for target classification. For this paper, the function g consists of the back-projection algorithm and a transformation from a complex to real valued image:

$$x_c = \texttt{backprojection}(S), \tag{6}$$

$$x = (20 \log_{10}(|x_c|) - \mu)/D. \tag{7}$$

where μ and D are set such that most of the values of x fall within the range of $[-1, 1]$.

The input image, x, is assumed to only contain a single target of class y out C different target classes. To classify the target within the processed image, execute a neural network consisting of a feature extractor, $f : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^M$, and classifier, $c : \mathbb{R}^M \rightarrow \mathbb{R}^C$, given by,

$$h = f_{\Theta_f}(x), \tag{8}$$

$$\hat{y} = c_{\Theta_c}(h), \tag{9}$$

where $\hat{y}$ is the estimated class probability vector. The functions f and c are neural networks whose parameters, Θ_f and Θ_c, are estimated by minimizing by the following cross-entropy loss using a form of backpropagation,

$$L_{clf}(x, y; \Theta_f, \Theta_c) = -\sum_{i}^{P} \hat{y}_i \log(y_i). \tag{10}$$

3.3 ROBUSTNESS TECHNIQUES

Pose Estimation The first approach we consider to improve robustness and provide better feature learning is to jointly estimate the target class *and* pose, $\theta \in [0, 2\pi]$, which is the angle between the target body axis and radar line-of-sight (see Figure 2). Joint training may improve feature learning by forcing the neural network to output features that best represent the information needed to classify a target *and* estimate its pose. We discretize the angle into T bins and use the categorical distribution to estimate the pose, $\hat{\theta}$, via a neural network,

$$\hat{\theta} = p_{\Theta_p}(h) \ , \qquad \theta \in \mathbb{R}^T. \tag{11}$$

The parameters of the pose estimator is trained along with the feature and classification parameters by minimizing an additional loss function,

$$L_{pose}(x, y; \Theta_p) = -\sum_{i}^{T} \hat{\theta}_i \log(\theta_i) \tag{12}$$

Similarity Embedding It is expected that similar inputs will result in feature vectors that are close given a distance metric. Learning a feature space that embeds this property will improve classifier robustness to small changes in target phenomenology (see appendix of [12]). Similarity depends on the specific application; for ATR using SAR, we define input similarity based on three properties: 1) targets belong to the same class, 2) targets have similar size, *and* 3) the targets have similar pose. To embed similarity into our network, we define a binary similarity label, s, (0 if inputs are similar, 1 if not) and consider the contrastive loss between two extracted feature vectors h_1 and h_2, from separate SAR images x_1 and x_2,

$$L_{sim}(h_1, h_2, s; \Theta_f) = (1-s)\|h_1 - h_2\|_2^2 + s\min(1 - \|h_1 - h_2\|_2, 0)^2. \tag{13}$$

This loss is minimized in conjunction with the classification loss.

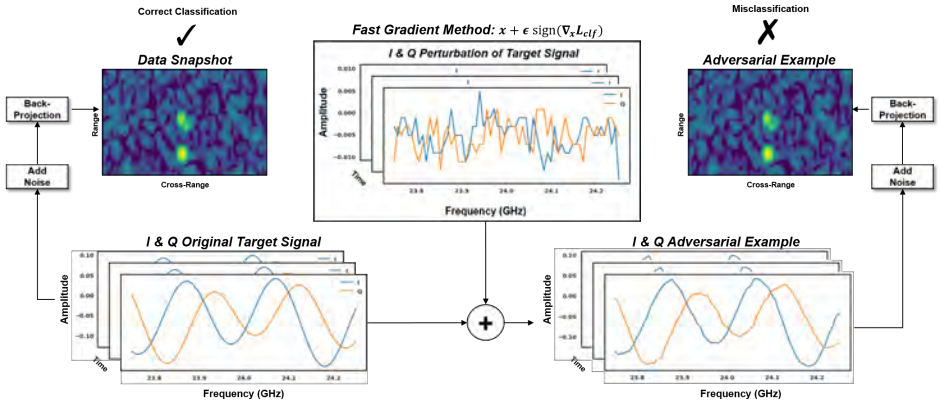

Figure 3. Overview of applying FGSM on radar signals.

Adversarial Learning Adversarial Learning with FGSM [10] has shown an ability to improve the robustness of a classification model. We utilize FGSM to perturb the complex-valued target frequency history before back-projection, providing a more "realistic" adversarial perturbation of the target response and aiming to improve robustness against small variations in a target's complex-valued phenomenology. That is, for a radar, we want to consider perturbations of the signal being received at the sensor rather than on the "pixels" of the input image to the target classifier, Figure 3 outlines the approach to generating perturbations of the signal. To do this, define the signal processing function, $x = g(s)$, that takes the received signal, s, and processes the signal

through the back-projection algorithm and normalizing functions defined in Equations 6 and 7. The perturbation defined by FGSM is given by,

$$\eta = \epsilon \text{sign}(\nabla_x L_{clf}(x = g(s), y; \Theta_f, \Theta_c)). \tag{14}$$

Then the perturbed input image the neural network is

$$x'_{\text{FGSM}} = g(s + \eta). \tag{15}$$

For adversarial training, we minimize the additional loss term,

$$L_{adv}(x'_{\text{FGSM}}, y; \Theta_f, \Theta_c) = -\sum_{i}^{P} \hat{y'}_i \log(y_i) \tag{16}$$

where y is the label of the original image, x, and $\hat{y'}_i$ is the label of the perturbed image, x'_{FGSM}.

4. EXPERIMENTS

4.1 DATA

The data set consists of 715 unique 3D targets of various size corresponding to four shape classes shown in Figure 4: cylinder, cone, dome-cylinder, and hour-glass. For each individual target, 1000 SAR images are generated, resulting in a total of 715,000 images to train and test our classifiers. This data set varies the target pose θ, radar altitude a, orbital radius r, initial orbit location φ_0, and background noise (see Figure 2). Each SAR image represents a six meter window in the xy-plane with 160 samples (depiction of SAR scenario and sample image shown in Figure 2).

4.2 RF SIMULATION

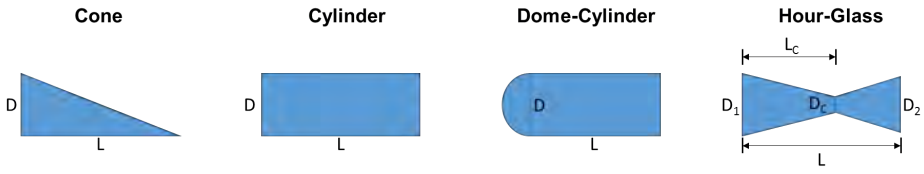

Figure 4. Description of shapes in data set.

This data set utilizes four different shape classes that are depicted in Figure 4. These shapes are modeled in 3D by assuming the targets are symmetric along the body-axis (roll symmetric). To generate a random sample of target shapes we define the distribution of parameters shown in Figure 4 as:

$$L \sim U[1,4] \tag{17}$$
$$D \sim U[1,2] \tag{18}$$
$$D_1 \sim U[1,2] \tag{19}$$
$$D_2 \sim U[1,2] \tag{20}$$
$$L_C = L/2 + 0.1 \tag{21}$$
$$D_C = 0.1. \tag{22}$$

The distribution of parameters are defined to challenge a classification algorithm to estimate the shape class independent of the objects sizes. In addition to modeling the geometric shape, half the samples will include basic "ring" (e.g., notch or groove) randomly along the body axis and is also roll symmetric.

We utilize an RF simulation tool developed internally to provide the frequency response for a given look angle and target shape. The simulations utilize Geometric Diffraction Theory (GDT) [20] to model the responses for a select number of scattering centers for a given shape. GDT is applicable in the high frequency region we focus on in this data set, the electromagnetic field can be written

as

$$E(f_k, s_n) = A(f_k, s_n, \theta) \exp\left(-i4\pi f_k r_n / c\right) \tag{23}$$

where there are K frequency samples per signal denoted by f_k, s_n is the scattering center at a given range r_n, $A(f_k, s_n, \theta)$ is the complex amplitude response of the scattering center for given line-of-sight θ to the radar (see Figure 5), and c is the speed of light constant. See [21] and [22] for an example of how to model the complex amplitude of cones.

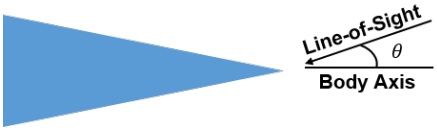

Figure 5. Description of line-of-sight.

An RF simulation is conducted for a randomly sampled shape, frequencies, and rotation angles about the geometric center of the object. For this data set, the center frequency is 24 GHz, bandwidth is $B = 0.5$ GHz, and the number of frequency samples is $K = 64$. A simulation for a sampled target, T, is then given by

$$E_T^P(f_k, \phi) = \sum_{n=1}^{N} A(f_k, s_n, \theta) \exp\left(-i4\pi f_k r_n / c\right) \tag{24}$$

where the scattering center range, r_n, is defined to be relative to the geometric center of the target, and P is one of the four possible polarization combinations: HH, HV, VH, and VV. For images generated in this paper, we utilize the circular polarized signal: $E^T = 0.5(E_T^{HH} + E_T^{VV})$. An example of simulated RF responses for each shape are show in Figure 6 as a function range relative to the geometric center of the shape versus the line-of-sight angle. As desired, the data set will challenge ATR, that is, the classification algorithm must distinguish between the different shapes by extracting features that separate the shapes based subtle changes observed across the viewing angles. For example, rear viewing angles will be challenging due to similarities in phenomenology between the different shapes while forward viewing angles provide the most variability.

4.3 TARGET CLASSIFICATION MODEL

The feature extraction architecture is a simple convolutional neural network (CNN) with layers C(16, 20, 1, 0) - C(32, 3, 2, 1) - C(64, 3, 2, 1) - C(128, 3, 2, 1) - C(256, 3, 2, 1) - P(5), where C(n, k, s, p) is a convolution layer followed by a ReLU non-linearity where n is the number of output channels, k is the kernels size in both dimensions, s is the stride, and p is the padding. The last layer is an average pooling layer with a kernel size such that the output is a vector of size 256.

The classifier function is a fully connected neural network of two linear layers: L(64) followed by a ReLU, and L(4) followed by a soft-max layer.

TABLE 1

Summary of accuracy and robustness results for ATR with SAR

	Accuracy	Robustness
BASIC	0.896 ± 0.011	0.0201 ± 0.0011
POSE	0.899 ± 0.009	0.0209 ± 0.0006
SIM	0.921 ± 0.013	0.0204 ± 0.0008
POSE+SIM	0.912 ± 0.013	0.0204 ± 0.0018
ADV	0.871 ± 0.006	0.0213 ± 0.0011
ADV+SIM	0.889 ± 0.005	0.0224 ± 0.0026

The pose estimator is a fully connected neural network of two linear layers: L(64) followed by a ReLU, and L(180) followed by a soft-max. We discretize the angle space into $T = 180$ angle bins.

The normalizing parameters in Equation 7 are $\mu = -40$ and $D = 50$. For adversarial learning, the scale of the perturbation in Equation 14 is set to $\epsilon = 0.001$.

4.4 EVALUATION METRICS

We perform 4-fold cross-validation to train and estimate the out-of-sample accuracy of each classifier. To evaluate the robustness of a classifier to adversarial perturbations, we use the metric, $\hat{\rho}_{adv}(f)$, introduced in [11]:

$$\hat{\rho}_{adv}(f) = \frac{1}{|D|} \sum_{x \epsilon D} \frac{\|\hat{r}(x)\|_2}{\|x\|_2}. \tag{25}$$

$\| \cdot \|_2$ represents the Euclidean (i.e., L2) norm. The minimum adversarial perturbation, $\hat{r}(x)$, for each SAR image, x, in the validation data set, D, is computed using the DeepFool algorithm. When comparing two classifiers, if $\hat{\rho}_{adv}(f_1) > \hat{\rho}_{adv}(f_2)$, we conclude classifier f_1 is more robust than f_2.

4.5 RESULTS

We compare the basic architecture of feature extractor followed by classifier (BASIC) with the following augmented training schemes: pose estimation (POSE), similarity embedding (SIM), pose estimation and similarity embedding (POSE+SIM), adversarial learning with FGSM (ADV), and adversarial learning and similarity embedding (ADV+SIM). Overall results are summarized in Table 1 and Figure 7.

First lets examine results qualitatively to gain some insight. Figure 8 demonstrates both classification and pose estimation for the different shapes and viewing angles. Results appear to be intuitive, for instance, by examining the two dome-cylinder results, we see that small viewing angles lead to uncertainty in the pose due to the strong spherical response of the target while higher viewing angles are more uncertain in the target classification due to the similarities in

phenomenology with the other shapes. A look at the cylinder results shows an expected bi-modal distribution in the pose estimate since the cylinder target is symmetric. The hour-glass result shows a multi-modal pose estimate distribution due to the complex and varying symmetry of the shape. Lastly, the cone example demonstrates both accuracy in classification and pose estimation from smaller viewing angles due to the conic nature of the target that is different from the other shapes. From these results, we can see that by adding pose, we learn a better representation of the data that also supports our intuition.

Examining the overall results in Table 1, we see that each of the augmented training techniques leads to an increase in robustness over the basic classifier. Adversarial learning with similarity embedding has the highest robustness to adversarial perturbations, which is expected because this approach directly optimizes a loss function that applies small perturbations to the classifier input. However, since FGSM is a form of regularization, adversarial learning results in a slight drop in accuracy compared to the basic classifier. On the other hand, pose estimation and similarity embedding both result in increases in accuracy. We theorize that by conditioning the classifier on information such as pose and similarity properties, we learn more effective representations of the data and hence achieve higher accuracy.

The set of shapes used for this experiment were designed to be challenging (see Section 4.2). For viewing geometries with small pose angles (front viewing) each of the objects exhibit the highest amount of variability in their phenomenology, but all objects look similar from rear viewing geometries, e.g., the base of the cone and cylinder look similar. Therefore we expect better classification performance in front viewing geometries. Figure 6 illustrates the varying phenomenology between the different shapes and viewing angles. Performance across different viewing geometries are shown in Figure 7. As expected, classification performance is better for front viewing geometries for all classification architectures and degrades as viewing angles increase. Additionally, as described above, we see a drop in overall accuracy for the both types of adversarial learning architectures, yet robustness increases across all viewing angles. Broadside is usually specular in nature and therefore exhibits little phenomenology to perform classification between differing shape, therefore small variations in the signal can lead to miss-classification.

Examining the behavior across the different training methods, adversarial learning clearly improves the robustness of the classifier over all viewing angles while reducing the overall accuracy. Yet, the results demonstrate that adding similarity embedding to adversarial learning improves the robustness while also improving the accuracy over basic adversarial learning. Adding pose is expected to improve accuracy and remains to be seen in future analyses if robustness also improves. In addition to augmenting the training architecture, additional studies are needed to examine the accuracy and robustness as a function of the magnitude of the perturbation. Figure 9 demonstrates an initial study to show the impact on classification of the cone when increasing the magnitude of the perturbation. As the perturbation increases, the amount of reduction in accuracy also increases. Yet, by using adversarial learning with similarity embedding we see an improvement in the accuracy as the perturbation increases. We plan to expand on this study in future work while examining techniques to visualize the results via some form of class activation map.

5. DISCUSSION AND FUTURE WORK

In this project report, we present a convolutional neural network architecture and selection of training techniques for learning accurate and robust representations of 3D targets in active sensing environments, such as radar. We investigate these techniques using a simulated SAR for ATR scenario, and find adversarial learning to be the approach that achieves the highest robustness to adversarial attack, while pose estimation with similarity embedding increases the robustness while also achieving the highest accuracy.

To this point, this program has investigated the generation of adversarial examples using existing state-of-the art techniques such as Fast Gradient Sign Method (FGSM). While the developments this past year have just touched on the capabilities of these approaches to generate attacks, our technical focus going forward will expand on these techniques to develop a framework for producing attacks that effectively counter defenses trained to account for adversarial examples. This line of inquiry will assist in future development of ML algorithms that are resilient against this nature of attack.

Future work will include incorporating additional robustness metrics, performing similar analysis on other existing radar data sets (e.g., MSTAR [2]), and exploring the applicability of generative modeling for adversarial data augmentation that avoid the need to calculate the gradient. In-line with other current work, we are developing generative models, such as Generative Adversarial Networks (GAN) [23], to sample simulated radar observations that are within the target distribution but fool our classifiers [6, 13].

FGSM and other gradient-based approaches used to generate adversarial examples typically introduce small variations in the target or environment which leads to slight changes across the entire radar signal response (see Fig. 10(left)). While this approach may be effective in producing adversarial examples that stress classification algorithms, this signal variation may not correspond to any one physical feature that an adversary may add to a target to confuse algorithmic defenses. To better control for specific changes that an adversary may make to the target design (e.g., bolts, grooves, antennas, etc.) we hope to constrain our generative model to place realistic structure in the signal scattering response that may lead to misclassification of the target (see Fig. 10(right)). We hypothesize that training with these generative models will help increase robustness in data-starved radar applications by supplementing the training data with stressing adversarial examples that represent a fuller distribution of possible physical realizations of potential targets.

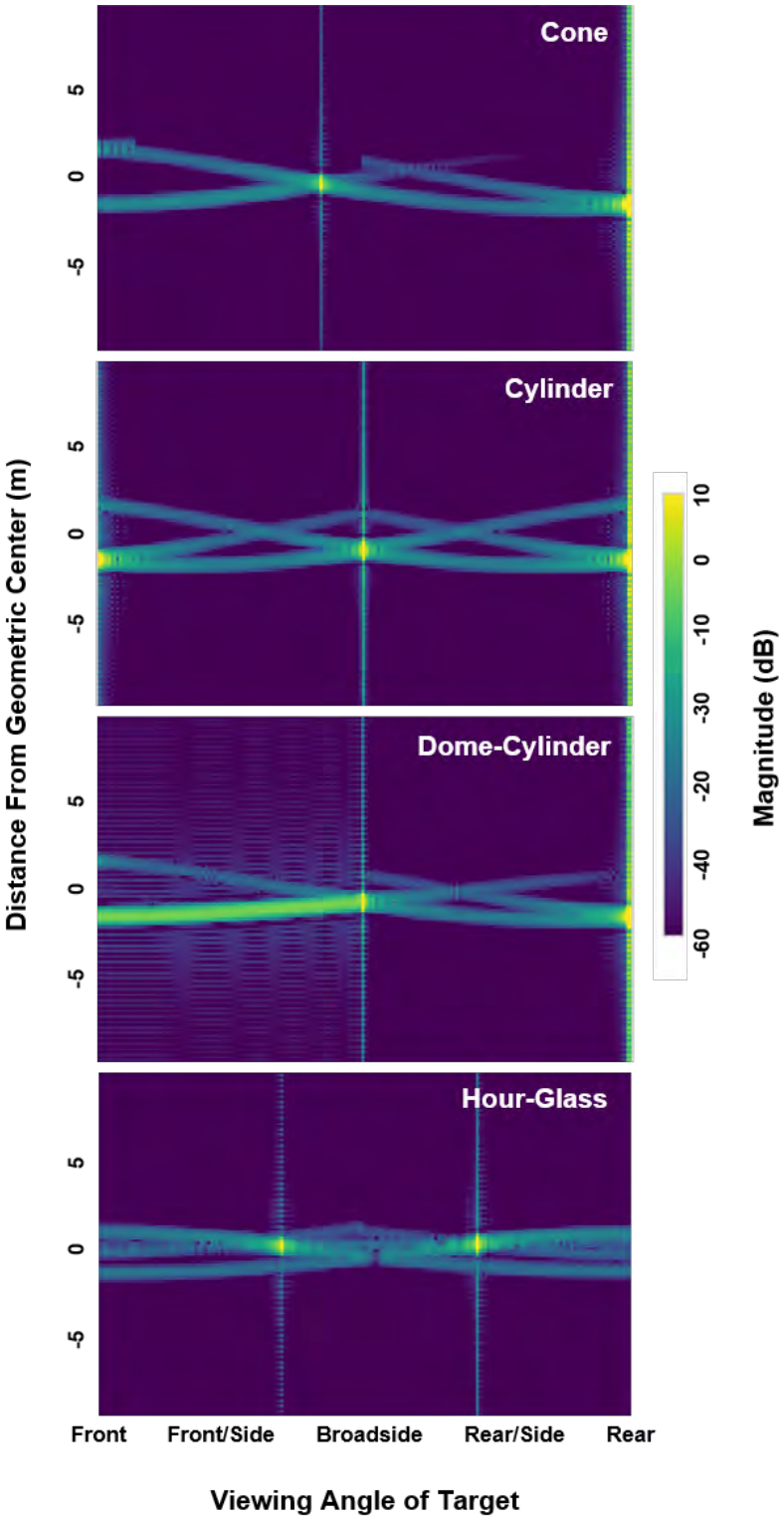

Figure 6. Example of simulated RF responses for each shape

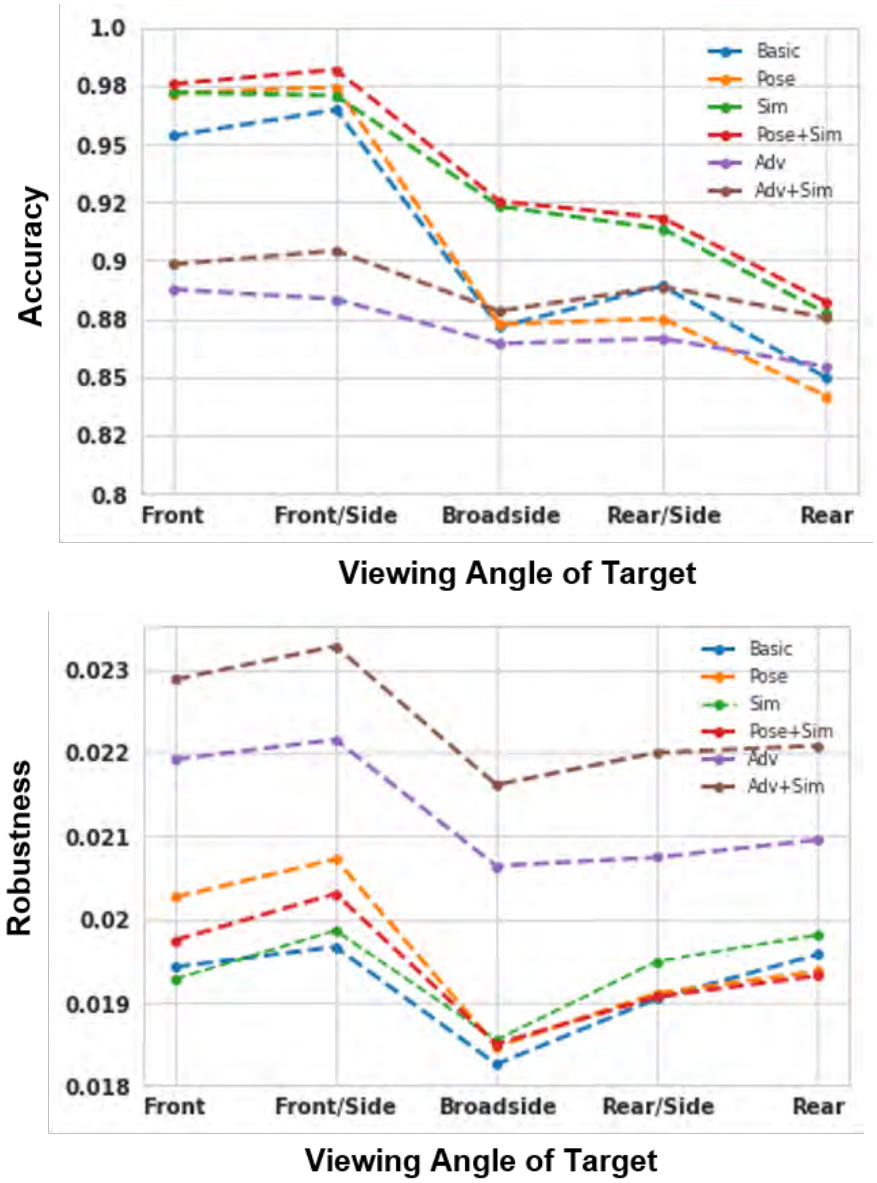

Figure 7. Accuracy (top) and Robustness (bottom) results of the described classification architectures for ATR of the simulated SAR images

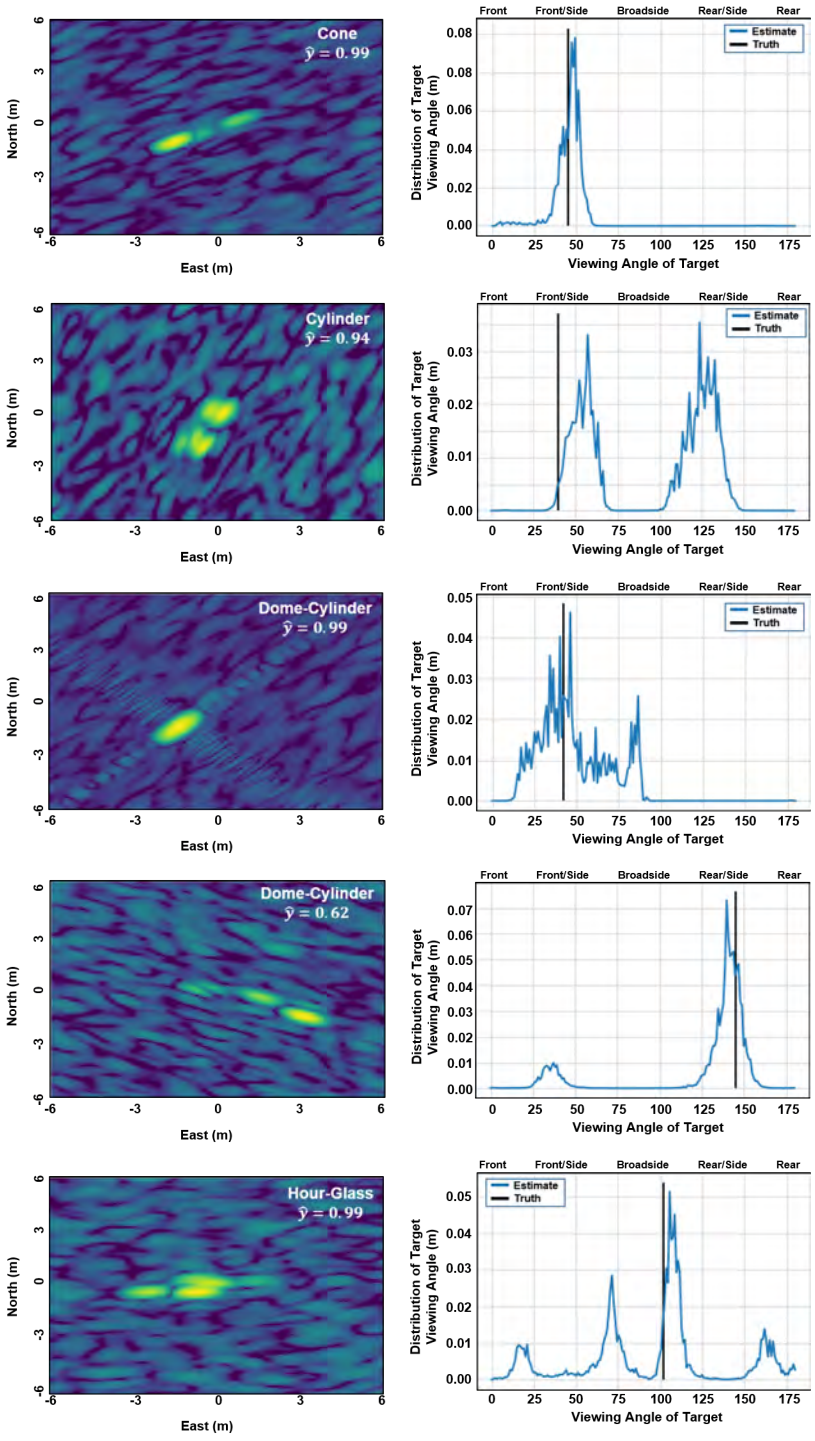

Figure 8. Example results across shape and viewing geometries.

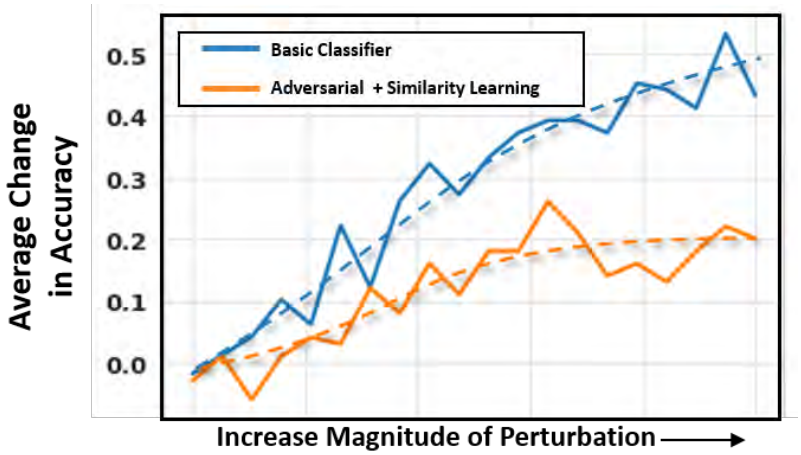

Figure 9. Accuracy of classification of cone shape as the magnitude of the perturbation increases.

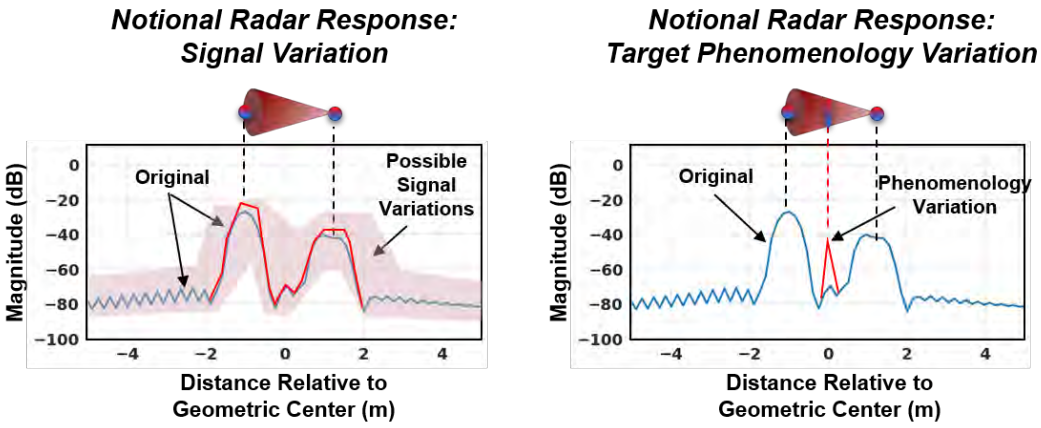

Figure 10. Approaches to radar signal variation.

REFERENCES

[1] M. Skolnik, "Introduction to radar," *Radar Handbook* 3 (2008).

[2] Y. Yang, Y. Qiu, and C. Lu, "Automatic target classification–experiments on the MSTAR SAR images," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005*, IEEE (2005), pp. 2–7.

[3] M. Wilmanski, C. Kreucher, and J. Lauer, "Modern approaches in deep learning for sar atr," in *Algorithms for Synthetic Aperture Radar Imagery XXIII*, International Society for Optics and Photonics (2016), vol. 9843, p. 98430N.

[4] A. Profeta, A. Rodriguez, and H.S. Clouse, "Convolutional neural networks for synthetic aperture radar classification," in *Algorithms for Synthetic Aperture Radar Imagery XXIII*, International Society for Optics and Photonics (2016), vol. 9843, p. 98430M.

[5] E. Mason, B. Yonel, and B. Yazici, "Deep learning for radar," in *Radar Conference (RadarConf), 2017 IEEE*, IEEE (2017), pp. 1703–1708.

[6] J. Guo, B. Lei, C. Ding, and Y. Zhang, "Synthetic aperture radar image synthesis by using generative adversarial nets," *IEEE Geoscience and Remote Sensing Letters* 14(7), 1111–1115 (2017).

[7] D. Morgan, "Deep convolutional neural networks for atr from sar imagery," in *Algorithms for Synthetic Aperture Radar Imagery XXII*, International Society for Optics and Photonics (2015), vol. 9475, p. 94750F.

[8] S. Wagner, K. Barth, and S. Brüggenwirth, "A deep learning sar atr system using regularization and prioritized classes," in *Radar Conference (RadarConf), 2017 IEEE*, IEEE (2017), pp. 0772–0777.

[9] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*, Springer (2013), pp. 387–402.

[10] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *ArXiv e-prints* (2014).

[11] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2574–2582.

[12] B. Wang, J. Gao, and Y. Qi, "A theoretical framework for robustness of (deep) classifiers under adversarial noise," *CoRR* abs/1612.00334 (2016), URL http://arxiv.org/abs/1612.00334.

[13] C. Xiao, B. Li, J.Y. Zhu, W. He, M. Liu, and D. Song, "Generating Adversarial Examples with Adversarial Networks," *ArXiv e-prints* abs/1801.02610 (2018), URL http://arxiv.org/abs/1801.02610.

[14] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Technical rep. (1985).

[15] Y.S. Abu-Mostafa, M. Magdon-Ismail, and H.T. Lin, *Learning from data*, vol. 4, AMLBook New York, NY, USA: (2012).

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199* (2013).

[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *CoRR* abs/1608.04644 (2016), URL http://arxiv.org/abs/1608.04644.

[18] K.E. Dungan, J.N. Ash, J.W. Nehrbass, J.T. Parker, L.A. Gorham, and S.M. Scarborough, "Wide angle SAR data for target discrimination research," *Proc.SPIE* 8394, 8394 – 8394 – 13 (2012), URL https://doi.org/10.1117/12.925077.

[19] L.A. Gorham and L.J. Moore, "SAR image formation toolbox for MATLAB," *Proc.SPIE* 7699, 7699 – 7699 – 13 (2010), URL https://doi.org/10.1117/12.855375.

[20] J. Keller, "Geometrical theory of diffraction," *J. Opt. SOC. Amer.* 52 (1962).

[21] M.E. Bechtel, "Application of geometric diffraction theory to scattering from cones and disks," *Proceedings of the IEEE* 53(8), 877–882 (1965).

[22] T.B.A. Senoir and P.L.E. Uslenghi, "Further studies of backscattering from a finite cone," *Radio Science* 8(3), 247–249 (1973), URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/RS008i003p00247.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems* (2014), pp. 2672–2680.