Training SAR-ATR Models for Reliable Operation in Open-World Environments

Nathan A. Inkawhich, Eric K. Davis, Matthew J. Inkawhich, Uttam K. Majumder, *Senior Member, IEEE*, and Yiran Chen, *Fellow, IEEE*

***Abstract*—Training deep learning-based synthetic aperture radar automatic target recognition (SAR-ATR) systems for use in an "open-world" operating environment has, thus far proven difficult. Most SAR-ATR systems are designed to achieve maximum accuracy for a limited set of classes, yet ignore the implications of encountering novel target classes during deployment. Even worse, the standard deep learning training objectives fundamentally inherit a closed-world assumption, and provide no guidance for how to handle out-of-distribution (OOD) data. In this work, we develop a novel training procedure called adversarial outlier exposure (AdvOE) to codesign the ATR system for accuracy and OOD detection. Our method introduces a large, diverse, and unlabeled auxiliary training dataset containing samples from the OOD set. The AdvOE objective encourages a deep neural network to learn robust features of the in-distribution training data, while also promoting maximum entropy predictions for adversarially perturbed versions of the OOD data. We experiment with the recent SAMPLE dataset, and find our method nearly doubles the OOD detection performance over the baseline in key settings, and excels when using only synthetic training data. As compared to several other advanced ATR training techniques, AdvOE also affords significant improvements in both classification and detection statistics. Finally, we conduct extensive experiments that measure the effect of OOD set granularity on detection rates; discuss the implications of using different detection algorithms; and develop a novel analysis technique to validate our findings and interpret the OOD detection problem from a new perspective.**

***Index Terms*—Automatic target recognition (ATR), deep learning (DL), out-of-distribution (OOD) detection, synthetic aperture radar.**

I. Introduction

IN THE design of automatic target recognition (ATR) systems for synthetic aperture radar (SAR) data, we recognize that it is critical to consider an operating environment in which test samples may or may not be from one of the known classes (i.e., an "open-world" environment). In such a setting, there are two primary goals. First, the system must be capable of accurately identifying and classifying some set of "in-distribution" (ID) target classes. These are categories of targets that the ATR models are trained on, and at test time, we seek generalization to their many views and perspectives. The second goal is to be capable of reliably detecting and rejecting any "out-of-distribution" (OOD) data observed during the model's deployment. For example, if a military-vehicle classifier encounters the signature of a minivan, it should refrain from outputting a classification and instead reject the sample. In this work, we codesign a deep learning (DL) based SAR-ATR algorithm that is both accurate on ID data and can robustly identify OOD test data under a variety of training conditions. These goals are most commonly associated with the topic areas of open-set recognition (OSR) [1], [2] and OOD detection [3]–[5], and are distinct from open world recognition (OWR) [6], [7] in that we do not strive to incrementally learn the new classes during deployment.

As motivation, we find that most current research of SAR-ATR algorithms only considers the objective of being accurate and robust on ID data [8], [9], while ignoring the fundamental OOD problem in DL-based ATR systems. Notably, Inkawhich *et al.* [10] have recently shown that purely optimizing for accuracy in an SAR-ATR system can yield substantial gains in the ID classification performance, but result in only meager gains in the OOD detection performance. Such findings lead to the observation that ID accuracy is not well correlated with OOD detection, and instigate our codesign in this work. To reason about these findings, we postulate that training deep neural networks (DNNs) on a fixed set of classes, and always striving for confident predictions, implicitly suggests a closed world assumption. There is no indication that other classes exist, nor is there any directive for what to do when such data is encountered. Thus, the lack of consideration for OOD data in the phrasing of the DNN's learning objective is an inherent weakness in their design, which we strive to mitigate.

The basic ATR system design we consider consists of two modifiable parts. The first is a DNN model that is trained as a classifier on the training dataset, and at test time attempts to categorize the input data as one of the known classes. The second is an "OOD scoring" mechanism that probes signals of the DNN (which is regarded as a feature extractor) and produces a decision about whether or not the data is ID. If the OOD prediction component determines that the data are OOD, the classifier abstains from releasing the prediction; otherwise, the prediction

Manuscript received December 17, 2020; revised February 4, 2021 and March 5, 2021; accepted March 22, 2021. Date of publication March 25, 2021; date of current version April 21, 2021. This work was supported in part by the by AFRL under contracts FA8750-18-C-0148 and FA8750-18-2-0057 *(Corresponding author: Nathan A. Inkawhich.)*

Nathan A. Inkawhich, Matthew J. Inkawhich, and Yiran Chen are with the Electrical and Computer Engineering, Duke University, Durham, NC 27708-0187 USA (e-mail: nai2@duke.edu; matthew.inkawhich@duke.edu; yiran.chen@duke.edu).

Eric K. Davis is with the Machine Learning and AI, SRC Inc., North Syracuse, NY 13212-2509 USA (e-mail: davis.ek38@gmail.com).

Uttam K. Majumder is with the Computing and Communications, Air Force Research Laboratory Information Directorate, Rome, NY 13441-4514 USA (e-mail: ukmccny@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3068944

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

is released. Many related works in OSR and OOD detection consider a similar system design, and focus expressly on the construction of the OOD scoring mechanic, while assuming that a "good-enough" pretrained feature extractor exists [1], [2], [11]. However, we posit that in a significant way, the detection is limited by the information content in the extracted features. So, in this work, we take a different approach where we focus on improving the training process of the DNN classifier/feature-extractor, for use with existing OOD scoring methods.

Intuitively, to overcome the challenge of having an implicit closed world assumption during the DNN model training, we propose to modifying the learning objective to include instructions for what to do when OOD data are encountered. We use the recently released Synthetic and Measured Paired Labeled Experiment (SAMPLE) dataset [12] as the ID task. We then augment the training set with a large, diverse, and unlabeled dataset containing SAR signatures of large ships collected across a variety of sensor platforms (note, the SAMPLE dataset contains classes of landlocked military vehicles). In essence, the models are trained to be accurate on the SAMPLE data and to produce maximum entropy predictions on the ship dataset, following a concept called outlier exposure (OE) [13]. Importantly, we develop a novel extension of OE called adversarial outlier exposure (AdvOE), which greatly improves the performance through the inclusion of active attackers within the training objective.

For the primary evaluations, we follow the Experiment 4.3 design in [12] to construct a difficult OOD detection problem using a holdout-class scheme. Critically, our AdvOE method assumes no knowledge-of the expected OOD set, making its application significantly more realistic. We also prioritize studying the impact of OOD set granularity on the detection performance by considering a spectrum of OOD data types, from highly granular/nearly-ID (e.g., other SAR data) to less-granular/obviously OOD (e.g., natural images and random noise). Ultimately, our AdvOE method outperforms all other methods across the OOD granularity spectrum, yielding state-of-the-art performance in both accuracy and detection abilities amongst competing SAR-ATR training methods.

Overall, our main contributions are as follows.

- We introduce a novel DNN training procedure called AdvOE, which simultaneously teaches the model to be accurate on ID data and have maximum entropy outputs for OOD data, all while in the presence of an active adversary.
- We show that using a large, unlabeled, and unrelated SAR dataset to the ID task can significantly boost the OOD detection capabilities of an SAR-ATR system.
- We achieve state-of-the-art performance in both accuracy and OOD detection for the SAMPLE experiment, including when using 100% synthetic training data and 100% measured test data.
- We provide a novel and informative analysis of the impacts of OOD set granularity on the detection performance.

II. Background

A. SAMPLE Dataset

As mentioned, we employ the SAMPLE dataset and the Experiment 4.3 design from [12] to evaluate our methods. This dataset contains (measured, synthetic) *pairs* from the ten MSTAR [14] target classes. In total, there are 806 training and 539 test pairs. The measured components were taken directly from the MSTAR public release dataset while the corresponding synthetic targets were constructed via electromagnetic signature prediction from meticulous computer aided design (CAD) models [12]. The class number to class name decoding used in this work is as follows: {0:"2S1"; 1:"BMP2"; 2:"BTR70"; 3:"M1"; 4:"M2"; 5:"M35"; 6:"M548"; 7:"M60"; 8:"T72"; 9:"ZSU23"}.

There are two key parameters associated with this dataset and its experiments that are referenced throughout this work. K sets the fraction of measured training data, while the test data are always 100% measured ($0 \leq K \leq 1$). For example, if $K = 0.75$ then 604/806 training pairs are represented by their measured component while the remaining 202/806 are represented by their synthetic component. As a special case, if $K = 0$ then 100% of the training samples are synthetic. To manufacture the OOD problem, parameter J is introduced to set the number of classes that are held-out from the training set ($1 \leq J \leq 8$). So, the classifiers are trained on the remaining $10 - J$ classes, and at evaluation time the test data for the held-out J classes is considered OOD while the test data for the other $10 - J$ classes is considered ID.

Finally, we emphasize the unique opportunity afforded by the SAMPLE dataset to study the impacts of training SAR-ATR models on synthetic data. In many situations where rapid development is required, it may not be feasible to collect a representative set of measured training data (if constrained by time and/or money) [15]. For this reason, we prioritize the case of having exclusively synthetic training data ($K = 0$). This assumption strictly increases the complexity of our "open-world" problem by introducing a distribution-gap [10] between the training and test sets, while also contributing to practicality.

B. Detecting Unknown Inputs

In related literature, there are several research areas that focus on the similar problem of detecting novel/unknown classes of inputs during deployment: OSR [1], [2], [11], [16], OWR [6], [7], [17], [18], OOD detection [3]–[5], [19]–[21], anomaly detection [22]–[25], and in some ways meta-recognition [26], [27]. Importantly, there are several distinctions to be made between our goals and the goals of some of these related topics. Different from OWR, we do not consider the problem of incrementally learning the new classes of data that have been detected. Our stated objective of accurate classification and reliable detection of novel data can be considered a subproblem of OWR that stops short of incremental learning. Different from most anomaly detection works, we wish to integrate the multiclass classification and novel input detection into a single system, rather than only considering the binary case of predicting whether or not a test sample emanates from the training distribution. Lastly, different from meta-recognition, we do not outwardly strive to identify incorrect predictions made on ID data. Thus, our work is most similar to the topics of OSR and OOD detection.

1) Open-Set Recognition: The crux of the problem in OSR research is to manage open space risk while maintaining the

reasonable generalization performance [1], [7]. One aspect of the problem that is often assumed is that there exists a "good" feature extractor that preprocesses the input data into an informative feature space. Early OSR works were built around non-DNN-based feature extractors (e.g., histogram of oriented gradients [28], scale invariant feature transforms [29], and nonnegative matrix factorization [30]), and focused solely on techniques to best separate ID from OOD data in the given feature space. For example, Scheirer *et al.* [1] develop the 1-vs-Set machine, Scheirer *et al.* [16] introduce the compact abating probability (CAP) model and Weibull-calibrated SVM algorithm, and Dang *et al.* [31] devise a two-stage technique specifically for the SAR-ATR domain that relies on exemplar selection and kNN-style classifiers.

However, with the advent of DL algorithms and convolutional neural networks (CNNs), recent OSR techniques have focused on using CNN features to take advantage of the generalization and expressive power of such algorithms. Bendale and Boult [2] determined that the key problem in this domain is the use of the softmax function to separate ID from OOD, as computing normalized probabilities over a fixed set of classes is inherently closed world. Thus, they replace the softmax layer (in a pretrained DNN) with an OpenMax layer to provably bound open space risk using per-class CAP models defined through extreme value theory. The extreme value machine [18] then improves on [2] through inclusion of nonlinear radial basis functions [7]. In summary, even though our work shares motivational commonalities with such OSR works, we propose that in large part our approach is technically orthogonal. As emphasized previously, OSR system designs tend to take for granted the existence of an informative feature extractor. In many recent papers, this turns out to be a pretrained AlexNet [32] model, where the OSR algorithm's input features are taken from the penultimate layer (i.e., logit layer) [2], [7], [18]. In contrast, most of our effort is spent training the DNN models to better separate ID and OOD data in the feature space. An interesting direction of future work is to evaluate the effect of using models trained with our developed algorithms as pretrained feature extractors within an OSR/OWR framework, where open space risk can be formally bounded.

2) OOD Detection: Finally, our work is most related to OOD detection for DNNs. The principle distinction from OSR is that it is not necessary to provably bound open space risk, but the ultimate goal is still high generalization and reliable novelty detection. An important design detail in OOD detection research is whether or not the method trains the base classifier. Several of the most popular approaches assume that the DNN is pretrained, and focus on deriving an ID/OOD score from the signals available in the model. For example, the work in both [3] and [4] rely on thresholding values at the output layer of a DNN. In contrast, the Mahalanobis distance-based detector leverages the intermediate feature space, and creates OOD scores through measurement of Mahalanobis distance to class-conditional Gaussian distributions [5]. In this work, we do extensive experimentation using these three methods as the "OOD scoring" mechanic, and defer further descriptions to the following sections.

There are also several popular OOD detection techniques that involve training the base DNN classifier. The OE method [13] leverages a diverse unlabeled dataset of OOD samples during training to drastically improve the performance of confidence score-based detectors (i.e., [3]) in natural image settings. Critically, the OOD samples used in OE training are not sampled from the OOD test sets. More details about OE will be discussed in the following sections. Another popular training technique is from Lee *et al.* [20], who use a generative adversarial network [33] to generate "hard" OOD samples given only the ID dataset. The image classifier is then trained to have confidence calibrated predictions on the ID and OOD data (in a similar spirit to [13]). However, in order to tune hyperparameters, this method assumes access to OOD samples from each of the test OOD distributions (see [20, Appendix B]), which is beyond any assumption made in our work. Also, the OE method is shown to consistently outperform this method in [13]. Finally, Mundt *et al.* [19] operate at the intersection of DL-based OSR and OOD detection, and propose to train generative classifiers (as opposed to discriminative ones) to better separate ID from OOD data and reduce epistemic uncertainty. We leave it as a future work to investigate the utility of generative classifiers in our setting, with special attention paid to their ability to generalize with a very small amount of training samples, and in the $K = 0$ case where there is a distribution gap between the training and test data [10]. Also, it would be useful to develop a way to leverage any known OOD samples in the formulation of the variational evidence lower bound used in training such generative classifiers.

III. Methodology

Our methodological design is guided by a two-step workflow. Step 1 is to train a base classifier to be accurate on the ID dataset. Here, training is performed "offline" with no knowledge of the expected OOD testing sets. Step 2 instantiates an OOD detector for the trained classifier, which produces a real-valued "OOD score" for each test input sample. The ID/OOD decision is then made via thresholding the score. In this section, we first discuss the five training procedures we consider for Step 1, then define the OOD detection algorithm for use with the trained models.

A. Training Procedure

For notation, let $g(x;\theta)$ represent the DNN model with parameters θ, which inputs an image x and outputs a logit vector over the set of $\mathcal{C}$ ID classes (where $|\mathcal{C}| = 10 - J$). Then, $f(x;\theta) = \text{softmax}(g(x;\theta))$ is the softmax normalized output vector, which constitutes a valid probability distribution over the $\mathcal{C}$ classes. Finally, let $H(f(x;\theta), y)$ represent the cross-entropy loss between the predicted probabilities and the truth-encoded label distribution y.

1) Standard: The first training procedure we consider is referred to as the `standard` method, and will be used as a baseline for comparison for the remainder of this work. The `standard` method represents a vanilla empirical risk minimization objective over the samples in the $10 - J$ class ID

training set $\mathcal{D}_{\text{id}}$, and is described as

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{id}}} \left[H(f(x;\theta),y)\right]. \tag{1}$$

The model's parameters are updated to minimize the expected risk (as measured with cross-entropy loss) for samples in $\mathcal{D}_{\text{id}}$, where the ground truth label y is defined as a one-hot vector.

2) Label Smoothing: The second training procedure is a straightforward extension of (1) called `lblsm` [34]. This method redefines the one-hot truth label y as a smooth-label y^{LS} using equation $y_c^{\text{LS}} = y_c(1-\alpha) + \alpha/|\mathcal{C}|$, where α is an introduced smoothing parameter. We include lblsm at level $\alpha = 0.1$ because it has been shown to improve the generalization and calibration abilities of DNNs [34], and more specifically, has recently shown to improve the classification performance of SAMPLE classifiers [10].

3) Adversarial Training: The third training procedure is called adversarial training (`AT`) [35] and works to minimize an adversarial risk. `AT` operates using the same empirical dataset $\mathcal{D}_{\text{in}}$, however, adds an inner maximization term, which actively perturbs the input data to maximize the same cross-entropy loss that is being minimized through parameter updates. Mathematically, the `AT` procedure [35] is described as

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{id}}} \left[\max_{\delta\in S_{\text{id}}} H(f(x+\delta;\theta),y)\right]. \tag{2}$$

Notice, the perturbation δ that is applied to the input image is constrained to exist in an allowable perturbation set S_{id}. In this work, we define S_{id} using the ℓ_∞ norm, so $\epsilon \leq ||\delta||_\infty$ for some predefined ϵ hyperparameter that controls how large the pixel-wise perturbation is in image space.

To approximate the inner maximization, we use an iterative projected gradient descent (PGD) [35] adversarial attack algorithm, computed as

$$x^{t+1} = \text{clip}(x^t + \alpha * \text{sgn}(\nabla_x H(f(x^t;\theta),y)), x^{t=0} \pm \epsilon). \tag{3}$$

At each iteration, the image is perturbed by a small amount α in the direction of the gradient of the loss w.r.t. the input ($\text{sgn}(\nabla_x H(f(x^t;\theta),y))$), effectively maximizing the loss on that sample. To ensure that the perturbed sample's effective δ exists in the $S_{,}$ defined by ϵ, there is a pixel-wise clipping operator, which defines the projection onto the ℓ_∞ norm-ball. In this work, we consider two adversarial levels, ($\epsilon = 2/255, \alpha = 0.5/255$) and ($\epsilon = 8/255, \alpha = 2/255$) both with 7 perturbing iterations.

We include `AT` because it brings several potential benefits to our ATR system. First, it has shown to learn robust features of the data that maintain strong correlation with the true classification labels [36]. In the context of SAR-ATR, `AT` has also shown to improve the robustness of ATR classifiers [8], including ones trained on synthetic SAMPLE data [10]; while also learning representations that focus on the features of the SAR targets rather than features of the background clutter [10]. We posit that the improved quality of the learned features in `AT` will help in the OOD detection task.

4) Outlier Exposure: The remaining two training methods introduce a fundamentally different term from the previous three. Notice that the `standard`, `lblsm`, and `AT` methods are all trained exclusively using the $\mathcal{D}_{\text{id}}$ empirical training set, which only contains training data from the $10 - J$ ID classes. As a result, the models are implicitly trained under a "closed-world" assumption where accuracy on the training dataset is the one-and-only goal. Furthermore, these learning procedures encourage maximally confident predictions on all training data, making the models inherently ill-conceived for identifying OOD data.

The `OE` method [13] works to inform the DNN's behavior on both ID and OOD data during model training. Functionally, it introduces a new dataset containing OOD samples, called $\mathcal{D}_{\text{ood}}^{\text{OE}}$. Critically, the $\mathcal{D}_{\text{ood}}^{\text{OE}}$ set is entirely exclusive of any OOD samples we shall evaluate against (e.g., we do *not* include any data from the J hold-out classes in $\mathcal{D}_{\text{ood}}^{\text{OE}}$), which we believe to be a more realistic/useful case. The `OE` training objective is

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{id}}} \left[H(f(x;\theta),y)\right] + \lambda \mathop{\mathbb{E}}_{\tilde{x}\sim\mathcal{D}_{\text{ood}}^{\text{OE}}} \left[H(f(x;\theta),\mathcal{U}_{\mathcal{C}})\right]. \tag{4}$$

The first term in (4) is the same as (1), and works to minimize the cross-entropy loss between the predicted distribution and the one-hot label distribution for samples in $\mathcal{D}_{\text{id}}$. Intuitively, this term is striving for ID accuracy. The second term encourages the DNN model to output a uniform distribution over the classes $\mathcal{U}_{\mathcal{C}}$ for samples from $\mathcal{D}_{\text{ood}}^{\text{OE}}$. In other words, the model is trained to have a maximum entropy output for samples not from the ID classes. Here, λ is a tune-able hyperparameter that weights the importance of the two terms. Small λ values down-weight the contribution of the outlier samples and, hence, place more emphasis on ID classification term. In contrast, large λ values prioritize the model to output uniform predictions for the OOD samples and may sacrifice the ID classification performance. In this work, according to [13], we use $\lambda = 0.5$, which was shown to be a useful value for image-based classifiers while having a nondetrimental impact on accuracy.

5) Adversarial Outlier Exposure: Finally, we introduce the `AdvOE` training procedure, which combines concepts from `AT` and `OE`. The learning objective for `AdvOE` is described as

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{id}}} \left[\max_{\delta\in S_{\text{id}}} H(f(x+\delta;\theta),y)\right] + \lambda \mathop{\mathbb{E}}_{\tilde{x}\sim\mathcal{D}_{\text{ood}}^{\text{OE}}} \left[\max_{\delta\in S_{\text{ood}}} H(f(\tilde{x}+\delta;\theta),\mathcal{U}_{\mathcal{C}})\right]. \tag{5}$$

The critical difference between `AdvOE` and `OE` is the inclusion of the inner maximization terms to both components of the objective. As applied to the first term, the model is trained to be accurate on the ID data while learning the most robust features for classification. As applied to the second term, the maximizer is constructing more difficult outlier samples for the DNN to learn with. By maximizing the cross-entropy loss against a uniform truth distribution, the perturbed $\tilde{x}$ samples are pushed toward lower entropy predictions, meaning they are perturbed such that they appear more ID to the DNN. The overall goal of `AdvOE` is to learn higher quality features of the ID data for classification while also learning a more reliable confidence estimator for OOD data via training on maximally difficult samples. In this work, we use $S_{\text{id}} = S_{\text{ood}}$ for the perturbation

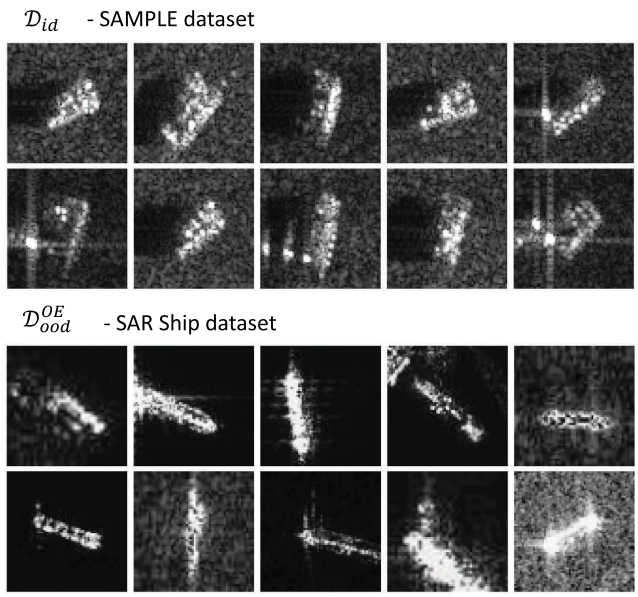


Fig. 1. Samples from both the $\mathcal{D}_{\text{id}}$ and $\mathcal{D}_{\text{ood}}^{\text{OE}}$ sets.

sets and approximate the innermaximizations using the same ℓ_∞ PGD attack and ϵ levels as the AT method.

B. Defining $\mathcal{D}_{ood}^{\text{OE}}$

From [13], the $\mathcal{D}_{\text{ood}}^{\text{OE}}$ set introduced in OE training is meant to be a large, diverse, unlabeled, yet somewhat realistic dataset containing samples from the wider OOD set. Given that our ID task is SAR target classification of military vehicles, we use data from the SAR-ship-dataset [37] to define a reasonably granular $\mathcal{D}_{\text{ood}}^{\text{OE}}$ distribution. The SAR-ship-dataset was originally proposed as a large object detection dataset that covers a variety of sensors, imaging modes, resolutions, and polarizations (with no relationship to the MSTAR collection process). In total, there are 59 535 ships from 43 819 full frame images. For our purposes, each ship was chipped from the full images using the ground truth box information. Fig. 1 shows samples from both the SAMPLE and SAR-ship-dataset which make up the $\mathcal{D}_{\text{id}}$ and $\mathcal{D}_{\text{ood}}^{\text{OE}}$ sets, respectively.

We emphasize that the SAR-ship-dataset represents a large and diverse set of SAR targets that has no special relationship to SAMPLE or MSTAR data, yet can still be useful to guide the confidence levels on unobserved OOD data. However, we do not claim that this is the "best" outlier set, and acknowledge that the selection of $\mathcal{D}_{\text{ood}}^{\text{OE}}$ may impact the performance of our methods. For example, if $\mathcal{D}_{\text{ood}}^{\text{OE}}$ is chosen such that samples from it can be easily separated from $\mathcal{D}_{\text{id}}$, or there is a lack of diversity, the OE-based methods may not have any performance gains because the model can learn a trivial solution for identifying samples from $\mathcal{D}_{\text{ood}}^{\text{OE}}$. On the other hand, if $\mathcal{D}_{\text{ood}}^{\text{OE}}$ is purposely constructed to be closer to $\mathcal{D}_{\text{id}}$, this may significantly improve OE-based methods as the models will be forced to learn higher fidelity representations of the ID target classes in order to achieve confidence calibration on the outlier samples. An important direction of future work is to measure the impact of $\mathcal{D}_{\text{ood}}^{\text{OE}}$ choice and to devise ways to construct maximally informative outlier datasets w.r.t. a given $\mathcal{D}_{\text{id}}$.

C. OOD Detection Method

Step two of our workflow is to define the OOD detection method. In step one, we outlined five techniques for training DNNs to be accurate on the classification task. Now, we use the pretrained classifiers as "feature extractors" [2], and define a method for producing a real-valued "OOD Score" for each test sample. In this work, we use the popular ODIN detector [4], which scores inputs based on the maximum confidence level from a temperature scaled softmax function. However, we note that our models are not limited to use with ODIN, and may be combined with other OOD/OSR methods such as OpenMax [2] in the future. Specifically, the ODIN score $S_{\text{ODIN}}(x)$ for some test input x is computed as

$$\mathrm{S_{ODIN}(x)} = \max_{i \in \mathcal{C}} \frac{\exp(g_i(x)/T)}{\sum_{j=1}^{\mathcal{C}} \exp(g_j(x)/T)}. \tag{6}$$

Here, T is a temperature hyperparameter, which is set to $T = 1000$ according to [4]. The intuition for ODIN is simple: OOD test samples should have low-confidence predictions, while ID test samples should have high-confidence predictions. The temperature scaling is a numerical trick empirically shown to help separate the ID/OOD scores. Note, $S_{\text{ODIN}}(x)$ is real valued and bounded in $[1/|\mathcal{C}|, 1]$, so to actually make an ID/OOD detection decision, a threshold β_{thresh} must be set. If $S_{\text{ODIN}}(x) \geq \beta_{\text{thresh}}$, x is considered ID and the classification prediction is given. If $S_{\text{ODIN}}(x) < \beta_{\text{thresh}}$ the sample is deemed OOD and rejected.

Critically, there is no global optimal value for β_{thresh}, rather it must be set according the consequences of an error (e.g., a false-negative). Therefore, to measure the performance of our detector, we follow [5] and leverage two metrics: area under the receiver operating characteristic curve (AUROC) and true negative rate at a detection threshold set to achieve a 95% true positive rate (TNR@95TPR). AUROC is a threshold-independent performance metric that assess the tradeoff between true positive and false positive rates across possible thresholds. TNR@95TPR represents a situation where β_{thresh} is set to achieve 95% TPR, which constitutes the high performance at identifying ID samples, and measures what the TNR is for the OOD set at that fixed threshold.

IV. Experimental Results

The experiments in this work are broken up into two major sections. First, we extensively evaluate the accuracy and the OOD detection performance of our trained models on the J class holdout dataset, following the procedures described in [12, Experiment 4.3]. Second, we investigate the impact of OOD set *granularity*, and measure how the detection performance changes as the OOD set transitions from nearly ID to obviously OOD.

TABLE I
DETECTION RESULTS FOR DIFFERENT MODEL TRAINING ALGORITHMS AND COMBINATIONS OF K AND J (NOTATION = AUROC / TNR@95TPR)

	$K = 1$			$K = 0.5$			$K = 0$		
Training Method	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$	$J = 1$	$J = 2$	$J = 3$
standard	92.5 / 75.2	91.4 / 72.2	92.3 / 74.4	88.9 / 60.9	87.7 / 56.8	89.3 / 59.8	72.8 / 26.5	71.7 / 22.9	72.4 / 23.4
lblsm	97.6 / 88.3	97.7 / 87.5	97.8 / 87.8	95.2 / 73.1	94.5 / 71.4	95.1 / 73.5	76.1 / 22.9	77.3 / 24.2	76.9 / 24.0
AT ($\epsilon = 2$)	94.6 / 81.3	92.3 / 75.6	91.6 / 74.0	90.2 / 68.8	89.2 / 65.0	90.7 / 66.4	74.7 / 30.4	72.5 / 22.8	73.5 / 24.8
AT ($\epsilon = 8$)	96.0 / 84.5	94.0 / 81.6	93.8 / 80.4	92.8 / 77.1	91.7 / 74.1	92.5 / 74.6	73.4 / 30.7	71.8 / 25.8	73.0 / 27.9
OE	97.0 / 86.2	96.2 / 82.3	96.3 / 83.2	93.5 / 70.7	93.1 / 68.0	93.1 / 71.1	80.6 / 31.6	79.1 / 32.7	80.5 / 34.3
AdvOE ($\epsilon = 2$)	98.6 / 93.5	98.0 / 90.5	98.2 / 91.4	96.4 / 83.0	95.8 / 80.7	96.1 / 82.0	82.1 / 39.6	81.0 / 36.6	81.6 / 39.2
AdvOE ($\epsilon = 8$)	**99.2 / 96.9**	**99.0 / 95.6**	**99.0 / 95.8**	**97.5 / 89.1**	**97.8 / 89.4**	**97.7 / 90.1**	**82.9 / 41.6**	**81.3 / 39.9**	**83.7 / 45.2**

The bold entries indicate the highest performance results in each column.

A. Experimental Setup

In all following experiments, the setup is nearly identical. Regardless of the five different training procedures, all models are trained with Gaussian noise augmentation, use the ResNet18 [38] model architecture with dropout layers [39], and input 64×64 pixel crops of the SAMPLE images. Inkawhich *et al.* [10] recently show this configuration to be helpful for training accurate SAMPLE classifiers. Every model is trained with a specified $[K, J]$ setting, and for each configuration, we run 100 iterations. Within a single iteration, we randomly initialize the model parameters and select a random set of J classes to holdout from the training set. We then train the model and evaluate the detection statistics for the specific holdout dataset (and any other OOD dataset of interest). All final reported statistics are averaged over the 100 iterations. Finally, every model is trained for 60 epochs using the ADAM optimizer, and has an initial learning rate of 0.001 and a single learning rate decay step at epoch 50 to 0.0001. We do not use a validation set, so to measure accuracy and detection statistics, we simply take the final model after 60 epochs of training.

B. Holdout Set Performance

The first primary experiment is to measure ATR performance using the J classes of holdout data. We consider performance as a function of detection rates on the OOD set and accuracy on the ID task. We also measure the impact of holdout class choice.

1) Detection: To quantify detection ability across a variety of operating conditions we measure how the AUROC and TNR@95TPR detection metrics change with K, J, and training algorithm. Table I shows results for values of $K \in \{1, 0.5, 0\}$, $J \in \{1, 2, 3\}$, and all five training algorithms (for the AT-based methods, we include two levels of perturbation).

First, we notice that the value of K significantly affects the performance. It appears markedly easier for models trained at $K = 1$ and $K = 0.5$ to detect the holdout OOD data, as compared to the $K = 0$ models. As evidence, the average AUROC values for the `standard` trained models are 92, 88, 72 for values of $K = \{1, 0.5, 0.0\}$, respectively. This is a somewhat expected result, as the $K = \{1, 0.5\}$ models are able to learn high-quality features from the measured data distribution during training, which evidently makes them better equipped to detect measured OOD samples during evaluation. The $K = 0$ models learn exclusively from the synthetic data representations, so for the detection task such models must also contend with the existing distribution gap between the synthetic and measured datasets [10].

Next, it appears that within a specified value of K, the value of J does not drastically impact the performance. In this work, we only consider values of J from 1 to 3 because we wish to focus on how the detection rates change with the training method and not how detection rates are affected by extremely low training data counts. We believe that for higher values of J, our training parameter choices (i.e., model architecture and levels of Gaussian noise and Dropout) may be suboptimal, which will inadvertently affect detection rates. We leave the study of training dataset size versus the detection performance as a future work.

Finally, we look across training procedures, where we consider the `standard` model as the baseline for comparison. Before discussing the implications of OE, we note that across all combinations of K and J, training with `AT` ($\epsilon = 8$) leads to detection improvements over the `standard` model. On average, `AT` ($\epsilon = 8$) improves AUROC / TNR@95TPR by: 2.5 / 8.2 for $K = 1$; 3.7 / 16.1 for $K = 0.5$; and 0.4 / 3.8 for $K = 0$. We postulate that the quality of the learned robust features is a driving reason for the improved detection performance of the `AT` ($\epsilon = 8$) model. In all but one case, `lblsm` also improves the performance over the `standard` model by similar margins.

When vanilla `OE` is included, the models are also better than the `standard` model across all combinations of K and J. However, in most cases `OE` performs only slightly better than the `AT` ($\epsilon = 8$) model, and in the $K = 0.5$ setting, `OE` underperforms `AT` ($\epsilon = 8$) in the TNR@95TPR statistic. Lastly, we see that our `AdvOE` ($\epsilon = 8$) training procedure is the top performer in all cases. On average, `AdvOE` ($\epsilon = 8$) improves AUROC / TNR@95TPR over the `standard` model by: 7.0 / 22.1 for $K = 1$; 9.0 / 30.3 for $K = 0.5$; and 10.3 / 17.9 for $K = 0$. Its margins over the vanilla `OE` model are also considerable, meaning that the addition of the innermaximizers proves substantive.

2) Accuracy: Besides detection ability, accuracy on the ID test sets is a key priority. Table II shows the average classification accuracy of the $K = 0$ models on the $10 - J$ class measured test datasets. In agreeance with the accuracy versus K results from [10], all of the $K = 1$ and $K = 0.5$ models are near or above 99% accurate, and thus, we do not include them here. Interestingly, the primary observation from Table II is that our

TABLE II
ACCURACY OF $K = 0$ MODELS FOR EACH TRAINING ALGORITHM

Training Method	$K = 0$		
	$J = 1$	$J = 2$	$J = 3$
standard	92.56 ± 1.64	92.19 ± 2.61	92.68 ± 3.08
lblsm	93.37 ± 1.58	92.85 ± 2.12	93.21 ± 2.24
AT ($\epsilon = 2$)	93.23 ± 1.42	93.05 ± 2.48	93.43 ± 2.41
AT ($\epsilon = 8$)	92.47 ± 1.68	92.19 ± 2.47	92.89 ± 2.69
OE	92.70 ± 1.88	92.29 ± 2.87	93.04 ± 2.76
AdvOE ($\epsilon = 2$)	**93.47 ± 1.54**	**93.64 ± 2.30**	93.83 ± 2.31
AdvOE ($\epsilon = 8$)	93.06 ± 1.56	93.34 ± 2.04	**93.96 ± 2.46**

The bold entries indicate the highest performance results in each column.

AdvOE training procedure is also capable of yielding the most accurate models. On average, the AdvOE ($\epsilon = 2$) models are over 1.1% more accurate than the standard and 0.97% more accurate than the OE models, across values of J. We also see that unlike in the detection results, which preferred $\epsilon = 8$, in the accuracy results the AdvOE ($\epsilon = 2$) models are slightly better. This finding coincides with [8], who observed that using AT with an ℓ_∞ $\epsilon > 2/255$ may yield slight accuracy degradation but can induce significant robustness advantages.

Although it may seem counter-intuitive that including OOD data in training would help accuracy on the ID task, we believe that in a similar spirit to semisupervised learning [40] theory, there is information in the unlabeled data that the model can leverage to improve the performance on the classification task. For example, a standard model may learn overgeneralized and nonrobust [36] features of $\mathcal{D}_{\text{id}}$ that are also present in $\mathcal{D}_{\text{ood}}^{\text{OE}}$, and thus, would clearly not be suitable for robust classification. So, training with samples from $\mathcal{D}_{\text{ood}}^{\text{OE}}$ would force the model to learn higher quality/more-robust features of the ID classes that may in-turn improve generalization. We note that [13] also observed similar ID accuracy gains due to the inclusion of outlier data during training. Lastly, recall that the OE and AdvOE models are trained with hyperparameter $\lambda = 0.5$ according to [13]. While this value elicits a productive tradeoff between accuracy and OOD detection as evident in Tables I and II, we remark that tuning λ to prioritize accuracy or OOD detection (as discussed in Section III-A) may be useful depending on one's operational requirements.

3) Holdout Class Dependence: In an effort to assess the accuracy and OOD detection ability of the SAMPLE classifiers from a different viewpoint, we examine the impact of holdout class choice. To isolate the influence of each class on OOD detection rates, we train standard and AdvOE ($\epsilon = 8$) models with $[K = 0, J = 1]$ and collect the detection statistics separately for each choice of holdout class. Fig. 2 shows the detection rates split by which class comprises the OOD set. First, we notice that holdout class choice *does* significantly impact the detection performance. For example, classes 1 and 6 are consistently difficult to detect, whereas classes 3 and 5 are more reliably detected. Interestingly, between the standard and AdvOE ($\epsilon = 8$) models the trends are not exactly consistent. With the standard model class 4 is detected often, while in the AdvOE ($\epsilon = 8$) model class 4 is among the most difficult to distinguish.

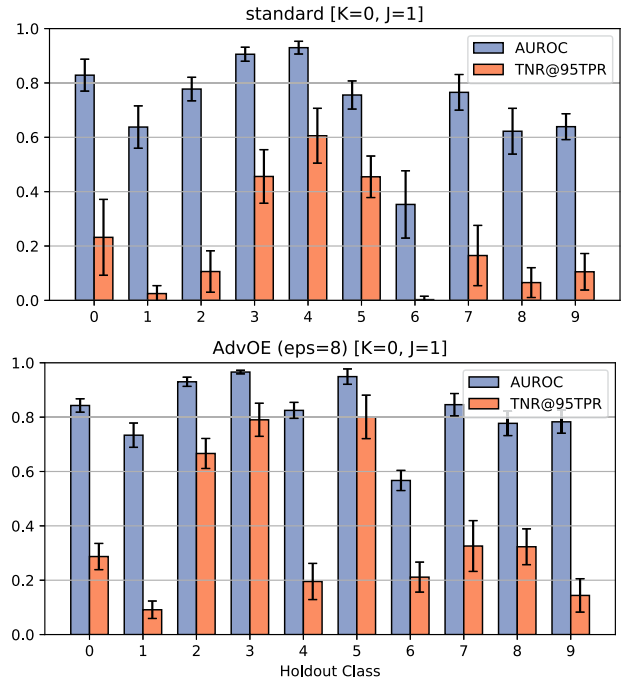


Fig. 2. Impact of holdout class choice on detection statistics.

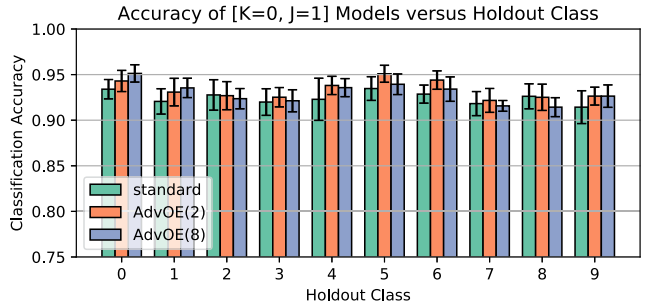


Fig. 3. Impact of holdout class choice on classification accuracy.

Next, to isolate the influence of each hold-out class on ID classification accuracy, we train standard, AdvOE ($\epsilon = 2$), and AdvOE ($\epsilon = 8$) models with $[K = 0, J = 1]$ and measure the average accuracy as a function of which class was held-out (i.e., accuracy over the remaining 9 ID classes). Fig. 3 shows the results of this experiment, where it is important to note that the average accuracy across holdout classes for a given model matches the numbers reported in Table II. Different from the detection results, accuracy appears to be relatively invariant to the choice of holdout class. However, we cannot say that one training method is superior across all choices of holdout, but on average the AdvOE ($\epsilon = 2$) is the most accurate ATR classifier. We believe the differences in class-wise detection rates and accuracies across the models may lie in unforeseen interactions between $\mathcal{D}_{\text{id}}$ and $\mathcal{D}_{\text{ood}}^{\text{OE}}$.

4) Discussion: Lastly, we would like to discuss our findings w.r.t. several related SAR OSR works that use the MSTAR and/or

SAMPLE datasets. First, Zelnio and Pavy [11] design an OSR baseline for the SAMPLE dataset that is comprised of a simple pretrained CNN classifier/feature-extractor (trained under similar settings as our `standard` models) and a set of One Class SVMs to do the classification/OSR steps. Their experiments are run using only one $[K = 1, J = 5]$ configuration (i.e., 100% measured training data and a fixed set of 5 ID classes) and their results show the classification performance in the range of 75% (see [11, Fig. 4]). Although we do not consider the $J = 5$ case in this work, we point out that in our most similar $[K = 1, J = 3]$ results from Tables I and II, our models achieve almost perfect OOD detection (99% AUROC) and over 99% classification accuracy. Next, Dang *et al.* [31] evaluate the OSR performance on the MSTAR dataset ($K = 1$), and design an exemplar-based kNN-style classification/OSR system (they do not use a DNN classifier/feature-extractor). Their experiments are run in an (approximate) $[K = 1, J = 7]$ configuration, and the results show that their system operates at about 97% accuracy (see [31, Table IV]). While impressive, it is unclear how this non-DL-based method would scale as the number of ID classes increases, or how the exemplar selection stage would be affected by the use of synthetic-only training data ($K = 0$) that may not perfectly resemble the measured data encountered during testing [10].

C. Impact of Granularity

The holdout class method of manufacturing an OOD problem yields a highly granular, challenging, and somewhat realistic situation that is certainly worth studying. However, all of the DNN models, we train will produce predictions over the $10 - J$ ID classes for any data that is formatted as a 64×64 px gray-scale image. In this section, we study how the granularity of the OOD data w.r.t. the training task affects the detection ability of our ATR systems. We define granularity as an intuitive metric for how comparable the OOD dataset is to the ID training dataset. A highly granular problem means the OOD set is qualitatively similar to the training set. A low-granularity problem means the OOD set is obviously OOD. Our motivation is to show that in the general case, OOD detection for DNNs is a highly nontrivial problem, and to introduce several novel observations about how granularity affects the detection performance.

1) Evaluation Datasets: To evaluate across the spectrum of granularity, we introduce several additional OOD datasets. Samples from each are shown in Fig. 4. First, we continue to use the J-class *Holdout* dataset to represent the most granular and intuitively challenging case. We also include SAR data from the civilian vehicle data domes (CVDome) [41] set from both the HH and VV polarizations as a medium-to-high granularity task. For the lower granularity tasks, we use *Random* noise samples drawn from a Uniform distribution, handwritten-digit samples from the MNIST dataset, and gray-scale natural images from the CIFAR10 dataset; all of which are rather obviously OOD from the perspective of a human.

2) Detection Results: Using the same experimental setup conditions described in Section IV-A, Fig. 5 shows the detection ability of models from the five training procedures on all of the

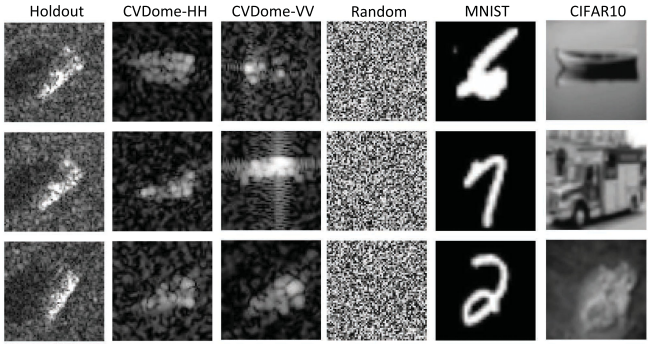


Fig. 4. Samples from granularity test datasets.

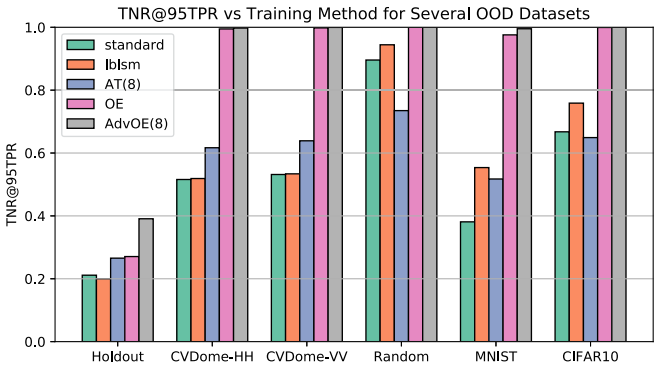


Fig. 5. Detection rates for several OOD datasets.

evaluation datasets. The models are trained with $[K = 0, J = 1]$ and results are averaged over 100 iterations.

First, notice the intuitive result that samples from the Holdout dataset are the most difficult to detect, while samples from the Random dataset are the easiest to detect (for non-`OE` models). This coincides with the two extremes of the granularity spectrum. Next, we see that for the non-`OE` models, the detection rates for the low-granularity datasets (i.e., Random, MNIST, CIFAR10) are relatively poor compared to how trivial this task would be for a human observer. For some models, samples from the MNIST dataset are even more difficult to detect than CVDome samples, despite possessing no SAR-like qualities. Since the ODIN detector fundamentally uses the confidence of predictions to make ID/OOD decisions, these results confirm that non-`OE` models tend to output high-confidence predictions even when the input data clearly does not belong to any of the training classes.

Finally, observe that the `OE` and `AdvOE` trained models are capable of almost perfectly detecting samples from all other OOD datasets besides Holdout. Even though the $\mathcal{D}_{\text{ood}}^{\text{OE}}$ training dataset contains only SAR ships, the method is able to help detect many other forms of OOD data. In some sense, we may say that the `OE` and `AdvOE` methods have learned a generalized concept of OOD beyond $\mathcal{D}_{\text{ood}}^{\text{OE}}$, which aligns with the findings in [13].

3) Analysis: Detection Algorithm Choice: Thus far, the outlier detection component of our ATR system has been built

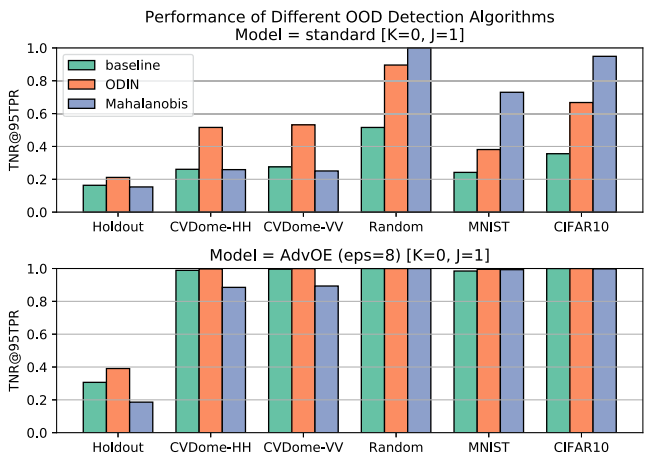


Fig. 6. Impact of choice of OOD detection algorithm.

around the ODIN OOD detection algorithm [4]. However, as discussed in Sections II-B and III-C, there are many alternative choices to consider for algorithms that estimate OOD scores using DNN feature-extractors. In this section, we compare the ODIN detector to two other popular OOD detection algorithms: the softmax-threshold (baseline) method [3] and the Mahalanobis distance detector [5]. The baseline method relies on the same intuition as the ODIN detector in that OOD samples should yield lower confidence predictions than ID samples. Functionally, the baseline detector is a nontemperature scaled version of ODIN, meaning it is equivalent to ODIN when $T = 1$ in (6). By comparing baseline to ODIN, we are evaluating the efficacy of temperature scaling. The Mahalanobis detector has a decidedly different methodology, which works to detect OOD samples using the model's feature space; and in some cases, it has shown state-of-the-art performance [5]. Conceptually, it relies on the intuition that OOD data should fall in low density regions of the training data's class-conditional feature distributions. First, the method learns the parameters of each class' Gaussian feature distribution at several layers across feature space using $\mathcal{D}_{\text{id}}$. Then, at test time, the OOD score is computed as the proximity of an input sample's feature representation to the nearest class distribution as measured by Mahalanobis distance. In this scheme, ID test samples tend to lie closer in proximity to the modeled distributions of the training data.

Fig. 6 shows the performance of each detection algorithm via the TNR@95TPR statistic for `standard` and `AdvOE` $(\epsilon = 8)$ models with $[K = 0, J = 1]$. From the `standard` model results, it is clear that granularity impacts each detection algorithm differently. While the baseline detector consistently underperforms ODIN, ODIN appears to be most adept at detecting high-granularity OOD data (i.e., Holdout, CVDome-HH/VV). Meanwhile, the Mahalanobis detector is the top performer on low-granularity data (i.e. Random, MNIST, CIFAR10). We point-out that this observation has not been discussed previously, yet we consider it to be an intuitive and useful finding. The highly granular OOD sets would seem more likely to activate the DNN's learned filters in an expected way, making feature space detection particularly difficult. On the other hand, the obviously OOD data may not activate the learned filters in a way similar to the ID data, making it more reasonable to detect such data using the feature space.

For the `AdvOE` $(\epsilon = 8)$ model, ODIN appears to be equivalent to, or better than, the other detectors in all settings. To reason about this result, we emphasize that the core concept of `OE` and `AdvOE` is to train the models to have low-confidence predictions on OOD data. Thus, it is not surprising that a confidence-based detector (i.e., ODIN) outperforms a feature space-based detector (i.e., Mahalanobis) when coupled with a DNN trained with `AdvOE`. With that said, we do not propose that ODIN is the "best" OOD scoring method for use with our models, and integration with more formal OSR techniques may boost performance.

4) Analysis: Trajectory Plots: To further explain/interpret our observations regarding the impacts of granularity, and the difficulty of OOD detection in general, we develop a novel analysis technique called trajectory plots. Consider a pretrained DNN classifier as a sequence of feature extraction layers followed by a linear classification layer. When we input a sample to the DNN, its signal propagates through the layers in the form of intermediate feature maps until it reaches the output layer. Following this intuition, a trajectory plot measures how the features of an input sample change as they propagate through the layers.

To create a trajectory plot, we start with a $10 - J$ class pretrained base classifier with fixed weights. At each layer, we probe the feature space and train a small $10 - J$ class model (called a FeatureNet) to predict which ID class the intermediate feature map belongs to. Thus, the input to each layer's FeatureNet is the base classifier's intermediate feature map, and the output is a predicted probability distribution over the $10 - J$ ID classes. In this work, each FeatureNet is made of two convolutional layers and a linear layer, and is trained for six epochs with a cross-entropy loss and lblsm over the $\mathcal{D}_{\text{id}}$ set. We intend the intuition for the FeatureNets to be simple: given some layer's feature map, attempt to classify it as one of the ID classes.

Once we have trained a FeatureNet for each layer of the base classifier, we can forward pass a sample and collect/assemble the FeatureNet outputs into an interpretable image. Consider the top-left subplot of Fig. 7, which shows the trajectory plot for a single ID test sample as measured on a `standard` model. Since we train with $[K = 0, J = 1]$ settings, the x-axis is divided into 9-bins representing the remaining ID classes. The row of pixels corresponding to a y-axis position shows a FeatureNet output vector for a layer of the base classifier, where the sum of pixel values is one and the intensity encodes the confidence in a class. Critically, by stacking the FeatureNet outputs from early layers (e.g., Layer $= 1$), to late intermediate layers (e.g., Layer $= 8$), and finally including the output softmax vector from the base classifier as the "out" row, we can monitor the input signal's trajectory/path through the network.

Fig. 7 shows trajectory plots as measured on `standard` (top section) and `AdvOE` $(\epsilon = 8)$ (bottom section) $[K = 0, J = 1]$ models. Each column corresponds to a different test dataset, where "ID" is the ID test set and the others are all OOD. For

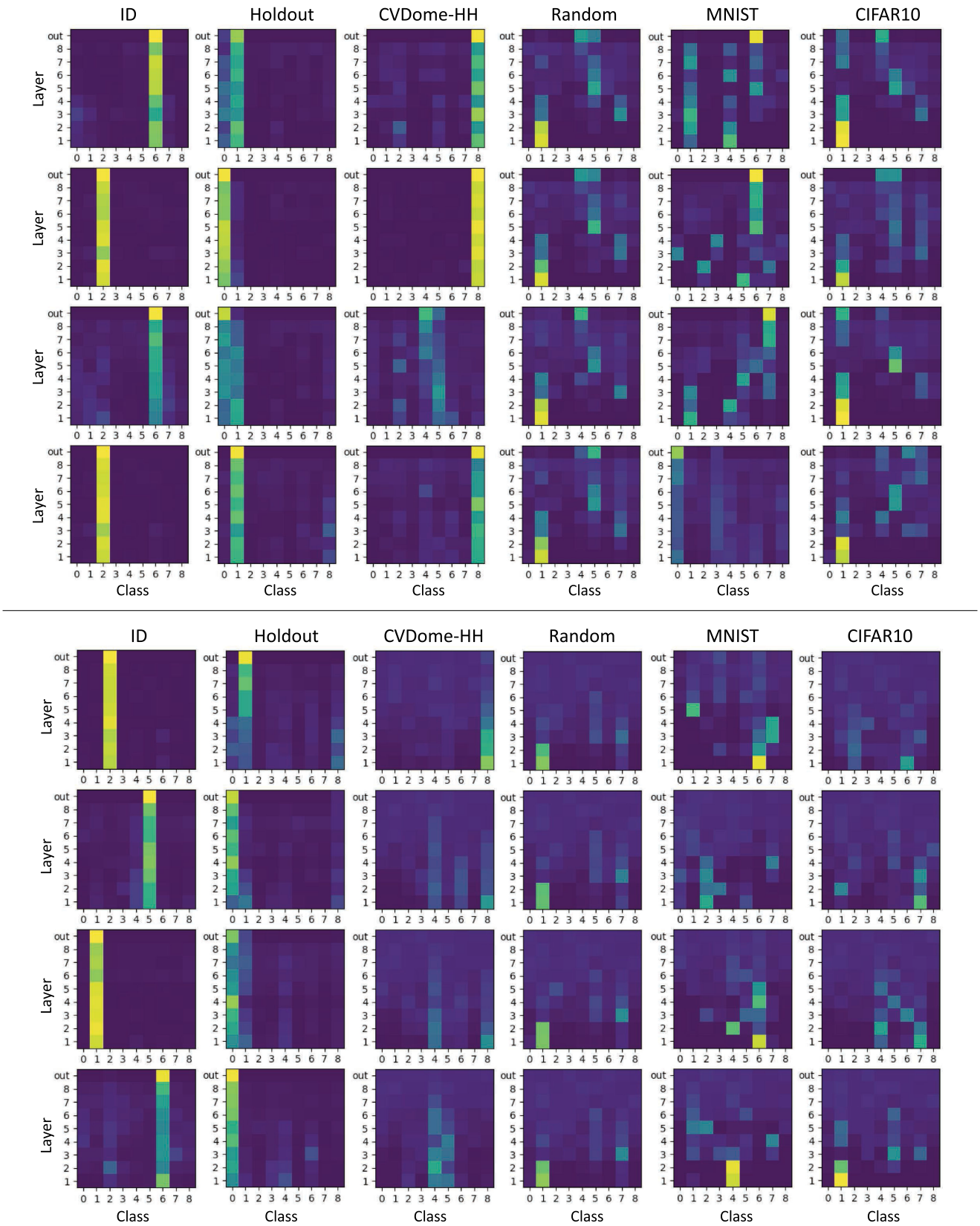


Fig. 7. Trajectory plots for several samples from each dataset. The top section is measured on a `standard` $[K = 0, J = 1]$ model. The bottom section is measured on an `AdvOE` $(\epsilon = 8)$ $[K = 0, J = 1]$ model. Each column corresponds to a different test dataset. Each row represents a different random sample from the dataset.

each dataset and model, we create trajectory plots for 4 random samples.

First, consider the `standard` model's plots, and how they differ across test datasets. The ID samples form strong column-based trajectories, meaning the features propagate through the network always "looking-like" a single class. Interestingly, the Holdout and CVDome-HH samples also tend to form column-based trajectories, meaning they propagate through the network in a way very similar to the ID data. This finding explains the general difficulty of detecting such OOD data in the previous experiments. Through the lens of granularity, it also confirms that highly granular data, which is most difficult to detect as OOD, is processed by the model in a way similar to the ID data. Now, consider the coarsely granular Random, MNIST, and CIFAR10 samples. The trajectories are not column-like, and the FeatureNet predictions have low confidence in the latter intermediate layers. However, the "out" layer's prediction remains confident in most cases. This is indicative of undesirable DNN model behavior, where even if the sample does not propagate naturally, the model still makes a high-confidence prediction. We also believe the scattered nature of these coarse OOD sample trajectories to be the reason that the Mahalanobis detector works well for the `standard` model in Fig. 6 results.

Second, consider the `AdvOE` $(\epsilon = 8)$ model's trajectory plots. As expected, we still see column trajectories for the ID data, which may be indicative of high model accuracy. Unfortunately, we also see column-like trajectories for the Holdout data samples, which explains why the Holdout dataset results remained relatively low in the Fig. 5 findings. Critically, for all other OOD datasets including the granular CVDome-HH, we clearly see that there are no column trajectories and no spurious high-confidence predictions at the output layer. In fact, these trajectories are indicative of intuitively "good" behavior in the `AdvOE` $(\epsilon = 8)$ model. In the first few layers, the model is extracting simple and general features, so some confidence in the FeatureNet outputs is expected. However, toward the later and output layers, the extracted features do not compose to form an "ID" target representation. So, instead of forcing the feature map to be part of a class, the model distributes the uncertainty across all of the classes and the predicted confidence is very low. In other words, unlike in the `standard` model, the `AdvOE` $(\epsilon = 8)$ model has the capability of articulating the sentiment of "I do not know."

Overall, the findings in the trajectory plot analysis confirm and even explain several of the trends found in previous experiments. In the future, we propose that this form of analysis can be very useful in quantifying granularity, as a prediction interpretability method, and even as a potential starting point for creating new OOD detection algorithms.

V. Conclusion

Our primary objective in this work is to design a methodology for training DL-based SAR-ATR models that makes them amenable to reliable operation in "open-world" operating environments. That is, to train models that are capable of accurate classification on a set of known ID classes, whilst also being able to reliably detect and reject samples from unknown classes (a.k.a., OOD data). Such ATR models are commonly composed of two components: a DNN classifier/feature-extractor and an OOD detection algorithm. In this work, we focus on the training of the DNN classifier and leverage the powerful ODIN algorithm [4] for the OOD detection component.

To codesign our DNNs for both accuracy and OOD detection, we develop the `AdvOE` training method and introduce a useful OOD dataset for SAR-ATR models to learn from [37]. The `AdvOE` method simultaneously learns the robust features of the ID data for the high classification performance, while also learning to output maximum entropy predictions for generic OOD samples. Through extensive evaluations, we find that our improved training method significantly boosts the accuracy and OOD detection capabilities of SAMPLE [12] classifiers, including ones trained exclusively on synthetic data. Furthermore, we analyze the impacts of OOD set granularity and find that our `AdvOE` models have learned a generalized concept of OOD, and are capable of almost perfectly identifying samples from a variety of OOD sets. Finally, we investigate the influence of detection algorithm and develop a novel analysis called trajectory plots to both explain the poor detection behavior of standard models, and provide insights into why the `AdvOE` models are top-performers.

In closing, we highlight several important directions for future work. The first is to investigate the implications of changing $\mathcal{D}_{\text{ood}}^{\text{OE}}$, and to develop a methodology for assembling maximally informative outlier sets w.r.t. a given set of ID classes/data. The second is to integrate our `AdvOE` models as feature extractors for use with formal OSR algorithms, e.g., [2]. Unlike the ODIN, Softmax-Threshold and Mahalanobis OOD detectors used in this work, OSR algorithms place formal bounds on open space risk and have the potential to produce "OOD scores" that better separate the ID and OOD data through more complex modeling techniques. Finally, one may leverage the `AdvOE` training method within an OWR SAR-ATR system. In such a system, rather than having a static model that detects and rejects OOD data based on a fixed training set, the system would dynamically collect and incrementally learn the new classes of data encountered during deployment. Our `AdvOE` method may be well-poised for such a system, as the $\mathcal{D}_{\text{id}}$ and $\mathcal{D}_{\text{ood}}^{\text{OE}}$ sets could be updated with authentic data and outliers from the actual deployment environment.

Acknowledgment

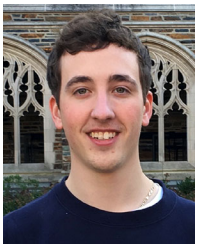
Disclaimer: The views expressed in this article are those of the authors and do not reflect official policy of the United States Air Force, Department of Defense or the U.S. Government. Public release number: AFRL-2020-0519

References

[1] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[2] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1563–1572.

[3] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[4] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[5] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7167–7177.
[6] A. Bendale and T. E. Boult, "Towards open world recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1893–1902.
[7] T. E. Boult, S. Cruz, A. R. Dhamija, M. Günther, J. Henrydoss, and W. J. Scheirer, "Learning and the unknown: Surveying steps toward open world recognition," in *Proc. Conf. Artif. Intell.*, 2019, pp. 9801–9807.
[8] N. Inkawhich, E. Davis, U. Majumder, C. Capraro, and Y. Chen, "Advanced techniques for robust SAR ATR: Mitigating noise and phase errors," in *Proc. IEEE Int. Radar Conf.*, 2020, pp. 844–849.
[9] S. Chen, H. Wang, F. Xu, and Y. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
[10] N. Inkawhich *et al.*, "Bridging a gap in SAR-ATR: Training on fully synthetic and testing on measured data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2942–2955, 2021, doi: 10.1109/JSTARS.2021.3059991.
[11] E. Zelnio and A. Pavy, "Open set SAR target classification," *Proc. SPIE*, vol. 10987, 2019, pp. 63–68, Art. no. 109870J.
[12] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, "A SAR dataset for ATR development: The synthetic and measured paired labeled experiment (SAMPLE)," *Proc. SPIE*, vol. 10987, 2019, Art. no. 109870H.
[13] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. Int. Conf. Learn. Representations*, 2019.
[14] T. Ross, S. Worrell, V. Velten, J. Mossing, and M. Bryant, "Standard SAR ATR evaluation experiments using the MSTAR public release data set," *Proc. SPIE*, vol. 3370, 1998, pp. 566–573.
[15] U. K. Majumder, E. P. Blasch, and D. A. Garren, *Deep Learning for Radar and Communications Automatic Target Recognition*. Norwood, MA, USA: Artech House, 2020.
[16] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.
[17] S. Dang, Z. Cao, Z. Cui, Y. Pi, and N. Liu, "Open set incremental learning for automatic target recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4445–4456, Jul. 2019.
[18] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, "The extreme value machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 762–768, Mar. 2018.
[19] M. Mundt, I. Pliushch, S. Majumder, and V. Ramesh, "Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019.
[20] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *Proc. Int. Conf. Learn. Representations*, 2018.
[21] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," 2018, *arXiv:1802.04865*.
[22] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.
[23] M. Markou and S. Singh, "Novelty detection: A review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.
[24] L. Ruff *et al.*, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.
[25] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of Gaussian kernel for one-class SVM," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 941–953, May 2015.
[26] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult, "Meta-recognition: The theory and practice of recognition score analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1689–1695, Aug. 2011.
[27] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3566–3573.
[28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 886–893.
[29] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157.
[30] S. Dang, Z. Cui, Z. Cao, and N. Liu, "SAR target recognition via incremental nonnegative matrix factorization," *Remote Sens.*, vol. 10, no. 3, 2018, Art. no. 374.
[31] S. Dang, Z. Cao, Z. Cui, and Y. Pi, "Open set SAR target recognition using class boundary extracting," in *Proc. Asia-Pac. Conf. Synthetic Aperture Radar*, 2019, pp. 1–4.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.
[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
[34] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4696–4705.
[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[36] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 125–136.
[37] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "A SAR dataset of ship detection for deep learning under complex backgrounds," *Remote Sens.*, vol. 11, no. 7, Mar. 2019, Art. no. 765.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[39] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[40] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.
[41] K. E. Dungan, C. Austin, J. Nehrbass, and L. C. Potter, "Civilian vehicle radar data domes," *Proc. SPIE*, vol. 7699, 2010, pp. 242–253, Art. no. 76990P.



Nathan A. Inkawhich received the B.S. degree in computer engineering from Clarkson University, Potsdam, NY, USA, in 2016. He is currently working toward the Ph.D. degree with the Electrical and Computer Engineering Department, Duke University, Durham, NC, USA, advised by Dr. Y. Chen of the Computational Evolutionary Intelligence (CEI) Lab.

He was with the Air Force Research Laboratory Information Directorate (AFRL/RI), Rome, NY, USA. His main areas of research are in machine learning algorithms and security, deep learning, anomaly detection, and robust automatic target recognition. For more information please visit: nathan.inkawhich@duke.edu.

Eric K. Davis received the B.S. degree in electrical engineering from Northeastern University, Boston, MA, USA, in 2013 and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2017.

Since 2017, he has been with SRC, Inc., where he is currently a Lead Systems Engineer with the Machine Learning and Artificial Intelligence team. Prior to SRC, he was with the General Electric Company developing embedded sensor systems for scientific and commercial applications. His current projects involve applications in machine intelligence, signal processing, and high-performance embedded computing.

Matthew J. Inkawhich received the B.S. degree in software engineering from Clarkson University, Potsdam, NY, USA, in 2017. He is currently working toward the Ph.D. degree in electrical and computer engineering from Duke University, Durham, NC, USA.

His research focus is building more robust and accurate convolutional network backbones for object detection models.

Uttam K. Majumder (Senior Member, IEEE) is a senior electronics engineer at Air Force Research Laboratory (AFRL). He received Bachelor of Science (BS) degree from the Department of Computer Science, The City College of New York (CCNY), in June 2003 graduating in Summa Cum Laude. He earned an M.S. degree from Air Force Institute of Technology (2007) and a Ph.D degree in electrical engineering from Purdue University (2014), West Lafayette, Indiana. He also earned an MBA degree from the Wright State University (2009).

He is currently a Senior Electronics Engineer with the U.S. Air Force Research Laboratory (AFRL), Wright-Patterson Air Force Base, OH, USA. His research interests include in artificial intelligence/machine learning (AI/ML), synthetic aperture radar (SAR) algorithms development for surveillance applications, radar waveforms design, and high performance computing for SARC-based automatic target recognition (ATR). He has recently authored or coauthored a book on *Deep Learning for Radar and Communications Automatic Target Recognition* (Artech House, 2020).

Dr. Majumder was the recipient of the, among various awards, AFOSR "STAR Team" award, Air Force Distinguished Civilian Award, and AFRL Science and Technology Achievement Award for radar systems development. For more information please visit: https://www.majumderfoundation.org/

Yiran Chen (Fellow, IEEE) received the B.S and M.S. degrees from Tsinghua University, Beijing, China, in 1998 and 2001, respectively, both in electronic engineering, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2005.

After five years in industry, he joined the University of Pittsburgh, Pittsburgh, PA, USA, in 2010 as an Assistant Professor and then promoted to Associate Professor with tenure in 2014, held Bicentennial Alumni Faculty Fellow. He is currently the Professor with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA, and the Director of NSF Industry-University Cooperative Research Center (IUCRC) for Alternative Sustainable and Intelligent Computing (ASIC), a Distinguished Lecturer of IEEE CEDA, and Co-Director of Duke University Center for Computational Evolutionary Intelligence (CEI), focusing on the research of new memory and storage systems, machine learning and neuromorphic computing, and mobile computing systems. He has authored or coauthored one book and more than 400 technical publications and has been granted 96 U.S. Patents.

Dr. Chen is or was the Associate Editor of several IEEE and ACM transactions/journals and served on the technical and organization committees of more than 50 international conferences. He is currently the Editor-in-Chief of IEEE Circuits and Systems Magazine. He was the recipient of seven best paper awards, one Best Poster Award, and 15 Best Paper Nominations from international conferences and workshops, and the NSF CAREER Award, ACM SIGDA Outstanding New Faculty Award, the Humboldt Research Fellowship for Experienced Researchers, and the IEEE SYSC/CEDA TCCPS Mid-Career Award. He is the distinguished member of ACM.