Generating Distributional Adversarial Examples to Evade Statistical Detectors

Yigitcan Kaya [1 †] **Muhammad Bilal Zafar** [2] **Sergul Aydore** [2] **Nathalie Rauschmayr** [2] **Krishnaram Kenthapadi** [3 †]

Abstract

Deep neural networks (DNNs) are known to be highly vulnerable to adversarial examples (AEs) that include malicious perturbations. Assumptions about the statistical differences between natural and adversarial inputs are commonplace in many detection techniques. As a best practice, AE detectors are evaluated against *adaptive* attackers who actively perturb their inputs to avoid detection. Due to the difficulties in designing adaptive attacks, however, recent work suggests that most detectors have incomplete evaluation. We aim to fill this gap by designing a generic adaptive attack against detectors: the *statistical indistinguishability attack* (SIA). SIA optimizes a novel objective to craft adversarial examples (AEs) that follow the same distribution as the natural inputs with respect to DNN representations. Our objective targets all DNN layers simultaneously as we show that AEs being indistinguishable at one layer might fail to be so at other layers. SIA is formulated around evading distributional detectors that inspect a set of AEs as a whole and is also effective against four individual AE detectors, two dataset shift detectors, and an out-of-distribution sample detector, curated from published works. This suggests that SIA can be a reliable tool for evaluating the security of a range of detectors.

1. Introduction

Deep neural networks (DNNs) have enabled breakthroughs in many challenging machine learning (ML) problems, such as image classification (Krizhevsky et al., 2012). However, it is well-known that DNN classifiers are vulnerable to *adversarial examples* (AEs) that contain malicious perturbations that subvert the predictions. Two defensive approaches have emerged against AEs: eliminating them by making DNNs robust (Madry et al., 2018); or detecting them before they harm the rest of the system (Metzen et al., 2017; Feinman et al., 2017). Detection techniques have gained traction as both theory (Shafahi et al., 2018) and practice (Rice et al., 2020) suggest that achieving robustness is challenging.

Prior work has shown that AE detectors can often be circumvented by *adaptive* adversaries (Carlini & Wagner, 2017). These attackers have full knowledge of the detector and take active steps to craft AEs that avoid detection. This *arms race* has been influential in establishing adaptive attacks as an evaluation standard for detectors (Roth et al., 2019; Raghuram et al., 2021). However, research shows that many detectors, despite following this standard, can still be defeated by more motivated adversaries (Tramer et al., 2020). Such adversaries identify and target important defensive assumptions instead of attacking many components a typical defense combines (Tramer et al., 2020). This methodology, by tailoring attacks to specific detectors, has been effective in exposing the vulnerabilities.

We aim to alleviate this arms race and the evaluation problems that stem from non-standardized attacks. We first observe that a common assumption behind many detectors is that adversarial and natural inputs are *statistically different* with respect to the hidden layer representations of DNNs (Feinman et al., 2017; Zheng & Hong, 2018). This assumption has given rise to two detection themes: (i) inspecting multiple samples as a distribution and comparing it with the distribution of natural samples (Grosse et al., 2017; Gao et al., 2021); and (ii) combining the statistics derived from multiple DNN layers (Raghuram et al., 2021).

Building on these observations, we propose a generic adaptive attack against AE detectors: the *statistical indistinguishability attack* (SIA). SIA targets the common assumption by crafting AEs that follow the natural data distribution with respect to DNN representations. Against (i), SIA crafts multiple AEs *jointly* and minimizes their statistical distance to the distribution of natural samples. Against (ii), SIA targets *all* DNN layers at once as we show that targeting the representations from a single layer fails to evade a detector that operates on another layer. This makes SIA effective against layerwise distributional detectors, which combine the two themes; whereas prior standard and adaptive attacks are easily detected. Unless the detectors have access to a

[†]Work done while at Amazon Web Services. [1]University of Maryland College Park [2]Amazon Web Services [3]Fiddler AI. Correspondence to: Yigitcan Kaya <cankaya@umd.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

significant number of AEs (over 1000) they fail to have acceptable detection performance against SIA. Moreover, SIA is moderately successful in the black-box setting where the AEs are crafted against a surrogate model and transferred to an unknown model.

Our formulation allows SIA to defeat five published AE detectors, designed to detect whether an individual input, or a set of inputs, is adversarial. Three of these detectors have been evaluated against adaptive attacks by their authors and claimed to be secure. However, SIA, without any customization, brings their detection performance down to near chance levels. This further validates our efforts to design a standard adaptive attack against detectors.

Finally, we highlight the flexibility of SIA by compromising other detection scenarios that often do not consider adversarial pressure. First, we target the settings where distributional detectors are used to detect *dataset shift*, e.g., concept drift. SIA, by perturbing the shifted dataset to follow the original distribution, prevents detection completely. Second, we attack a state-of-the-art out-of-distribution (OOD) sample detection method by Sun et al. (2021). This detector cannot distinguish between in-distribution inputs and the OOD inputs perturbed by SIA. To our best knowledge, our attacks in these scenarios also open up new threat models against the safety of DNNs.

The rest of the paper is organized as follows: We first present our setup, the detection methodology we use to evaluate our attack, and the metrics we report in our work (§2). We then conduct a case study on prior attacks and formulate SIA step-by-step as a general-purpose attack against detectors (§3). Next, experimenting on two vision datasets and three convolutional neural network (CNN) architectures, we evaluate SIA and compare it to prior attacks in terms of its success against distributional detectors (§4) and individual detectors (§5). Finally, we apply SIA in other detection scenarios and expose the vulnerabilities in them (§6).

2. Background and Formal Setup

A deep neural network (DNN) F is a series of parametric function compositions: $F = \mathcal{S} \circ f_N \circ \phi \circ f_{N-1} \circ ... \circ \phi \circ f_1$. We refer to $F_i(x)$, the intermediate output after the i^{th} hidden layer, as the i^{th} layer *representations* (of F on the input x). As special cases, we refer to $F_{N-1}(x)$ as the *penultimate*-layer representations (**PLR**) and to $F_N(x)$ as the *logits*. The activation function ϕ applies a non-linear transformation to the layer outputs and the softmax function $\mathcal{S}$ converts the logits into a probability distribution over $\mathcal{K}$ classes. In supervised learning, F is trained on a training set, which consists of multiple input-label pairs drawn from an underlying *natural* data distribution, $\{x, y\} \sim \mathbb{N}$. Training uses back-propagation to adjust the DNN's parameters to minimize its *loss* on the training set using a loss function, e.g., cross-entropy. The output, $F(x)$, is a $\mathcal{K}$-dimensional probability vector and $F^i(x)$ is the predicted probability of x belonging to class i. Typically, the prediction of a model is $\text{argmax}_i\, F^i(x) = \hat{y}(x)$, i.e., the most probable class.

Adversarial examples in deep learning. DNNs are shown to be vulnerable to adversarial examples (**AEs**) crafted by malicious actors (Szegedy et al., 2014). Essentially, an AE is an input sample x_a that is *very similar* to a natural example (**NE**) $x_n \sim \mathbb{N}$ but $\hat{y}(x_n) \neq \hat{y}(x_a)$. The similarity between x_n and x_a is evaluated using a distance metric, $D(\cdot, \cdot)$, usually an ℓ_p-norm distance. Generally, a crafting method optimizes an objective function to iteratively *perturb* x_n into x_a to change the model's prediction while keeping $D(x_n, x_a)$ small. There have also been methods that serve additional goals. Expectation over transformation (Athalye et al., 2018) is an adaptive attack against AE defenses that rely on randomness. DeepSloth (Hong et al., 2021) aims to cause slow inferences in dynamic early-exit models. AutoAttack (Croce & Hein, 2020) offers a parameter-free method to standardize robustness evaluations. Following this line, we propose a method to craft AEs that are indistinguishable from natural data to evade AE detectors. Most similarly, Gao et al. (2021) have developed an adaptive attack and claimed that their detector, SAMMD, is secure against it.

Undetectable attacks in cybersecurity. Security research hints that real-world defenders often deploy intrusion detection systems (IDS) against threats such as malicious network traffic (Paxson, 1999) or malware (Warrender et al., 1999). Essentially, an IDS learns the normal system behaviors when there is no attack and then it recognizes possible attacks by looking for statistical abnormalities. Facing an IDS introduces an additional constraint for the adversaries as their attacks need to be able to evade an IDS in addition to achieving the malicious outcome. To this end, the prior work (Wagner & Soto, 2002; Fogla et al., 2006) has developed automated methods that modify attacks so that they statistically match a system's normal profile to evade detection. Although undetectability is a core constraint of a realistic adversary in security, powerful AE crafting methods in ML literature are often only designed for attack success (Madry et al., 2018; Croce & Hein, 2020; Andriushchenko et al., 2020). In our work, we aim to fill this gap and develop a general-purpose AE crafting method that is an analogue of traditional evasive attacks. Our method crafts AEs while balancing between two objectives: attack success and evading statistical anomaly detectors.

2.1. Adversarial Example Detection Methods

Typical detectors, which we refer as *individual detectors* (**IDs**), operate on a single input sample to answer whether it is adversarial. Supervised IDs (Ma et al., 2018; Lee et al.,

2018) use a set of known AEs to train a binary classifier that learns to discriminate AEs from NEs. Unsupervised IDs (Feinman et al., 2017; Zheng & Hong, 2018; Roth et al., 2019; Miller et al., 2019), rely only on the statistical properties of NEs and detect non-conforming samples as AEs. A common thread in most IDs is extracting informative test statistics derived from DNN layer representations on the input samples (Raghuram et al., 2021). Further, researchers have also proposed *distributional detectors* (**DDs**) (Grosse et al., 2017; Gao et al., 2021) that inspect a set of samples as a whole. DDs assume that a set contains either only AEs or only NEs and make a determination between the two options. Although this assumption allows DDs to extract stronger statistical signals compared to IDs, it also makes DDs less practical (Carlini & Wagner, 2017).

We center our formulation around defeating more powerful DDs as we aim to develop an adaptive attack that would be effective against a broad class of detectors. We assume the role of an adversary who starts with X_0, an *initial set* of i.i.d. NEs drawn from $\mathbb{N}$ ($|X_0| = m$). With full access to the victim model, the adversary crafts an AE for each $x \in X_0$. The defender aggregates these AEs into the set X_a and deploys a DD to inspect it. Our goal is to ensure almost all AEs in X_a are successful while still bypassing the defender. We consider an AE as successful if it changes the model's output: when $\hat{y}(x_n) \neq \hat{y}(x_a)$ or when $\hat{y}(x_a) = t$ for a targeted AE with the target class t.

Distributional detection methodology. Prior DDs (Grosse et al., 2017; Gao et al., 2021) first form the set X_n by drawing natural samples from a holdout set. We consider DDs that construct X_n according to the model's predictions on X_a. This aims to prevent detection solely due to label shift, e.g., the holdout data is balanced and the non-adversarial inputs in X_a come from a single class. For each class y, this ensures $|\{x_i \in X_a | \hat{y}(x_i) = y\}| \approx |\{x_j \in X_n | \hat{y}(x_j) = y\}|$.

Next, a DD applies a two-sample test (**2ST**) between X_n and X_a. The null hypothesis of a 2ST ($\mathfrak{H}_0$) states that the two sets come from the same distribution. A common test statistic is the maximum mean discrepancy (**MMD**) (Gretton et al., 2006). MMD measures the *closeness* between two distributions in terms of a kernel k that gives point-level *similarities* of its inputs. The MMD between X_n and X_a is computed using the unbiased empirical estimator (Gretton et al., 2006), which assumes $|X_n| = |X_a| = m$. A permutation test or the wild bootstrap (Chwialkowski et al., 2014) is used to obtain the p-value of the estimate. Ultimately, a DD rejects $\mathfrak{H}_0$ if the p-value is less than a selected significance level, $0 < \alpha < 1$. A rejection implies detecting X_a as adversarial and α controls the false positive rate when X_a is natural.

Applying the MMD test in the input-space has been shown to perform poorly for distributions with complex structures (Liu et al., 2020a). Semantic-aware MMD (SAMMD), the state-of-the-art DD by Gao et al. (2021), tackles this problem by applying the MMD test on the representations of a pre-trained DNN. To tune its kernel, SAMMD splits the available data and learns the kernel parameters that maximize the power of the MMD test on the training split. Using these parameters, it then performs a 2ST on the test split.

As SAMMD specifically focuses on the PLR, we also consider more general *layerwise* DDs that can operate on any DNN layer. These DDs use the following kernel at i^{th} layer: $k_i(x, z) = exp(-\frac{1}{2\sigma_i^2}\|F_i(x) - F_i(z)\|^2)$, where x and z are the inputs. Here, σ_i is the *length-scale* of the Gaussian kernel, which, intuitively, controls how close the inputs of the kernel have to be to significantly influence its output. We also use the holdout data to standardize the representations and reduce their dimensionality with average pooling for CNNs. The splitting strategy SAMMD uses to tune its kernel reduces the test power as it leaves fewer samples for testing (Kübler et al., 2020). Therefore, layerwise DDs combine multiple kernels tuned using the median heuristic as a more efficient alternative (Kübler et al., 2020).

Relevant metrics. The attack success rate (**ASR**) measures the percentage of successful AEs among the crafted samples. The performance of a statistical test depends on the number of samples available to it, i.e., $|X_a| = |X_n| = m$. To this end, we repeat the test 100 times, each time by randomly selecting m samples from the crafted AEs and m samples from the NEs in the holdout set. Denoted by $\mathbf{DR_m}$, we then report the detection rate as the percentage of tests with sample size m that resulted in p-value less than $\alpha = 0.05$.

3. Statistical Indistinguishability Attack (SIA)

This section focuses on a case study to motivate our work and to formulate our attack methodology. First, we show how standard attacks such as PGD (Madry et al., 2018) lead to distinct representation distributions. SAMMD (Gao et al., 2021), a recent distributional detector (DD), can effectively detect these attacks. We then identify and address the challenges in crafting AEs to evade DDs.

We experiment with a ResNet-50 model on CIFAR-10 (see §4.1 for the details), start from 2000 test samples as our initial set X_0; and craft targeted AEs with random targets. We use UMAP (McInnes et al., 2018) to visually compare the representations of AEs and NEs in the holdout set. For simplicity, we visualize only the samples classified into the *horse* (randomly selected) class.

3.1. Standard Attacks Are Detectable

In Figure 1, we visualize the model's penultimate-layer representations (PLR) on the AEs crafted by three popu-

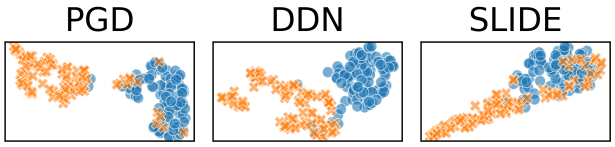

Figure 1. The penultimate-layer representations of the natural (✖) and the adversarial (●) examples crafted by three standard attacks.

lar attack algorithms. These attacks constrain the perturbations with different metrics: PGD (Madry et al., 2018) uses ℓ_∞, DDN (Rony et al., 2019) uses ℓ_2 and SLIDE (Tramer & Boneh, 2019) uses ℓ_1 distances. We present the details of these attacks in Appendix A. They also place an upper bound on the perturbations with a hyper-parameter, i.e., $D(x_n, x_a) < \epsilon$. For each attack, we find the minimum ϵ value to reach ∼100% ASR. We visually demonstrate that the distributions of these AEs are distinct from the NEs. Standard attacks optimize only for changing the model's prediction and disregard the natural distribution (Tramer et al., 2020). As a result, SAMMD obtains perfect detection rates (100%) when it has access to 30, 40 and 50 AEs against PGD, DDN and SLIDE, respectively.

3.2. Formulating SIA

We have shown that optimizing only for changing the model's predictions leads to detectable AEs. Ideally, an attack against a DD crafts a set of AEs that have *statistically indistinguishable* representations with respect to NEs. Towards this goal, we first note that the MMD between two sets of samples with respect to the i^{th} layer representations—$\text{MMD}_i(X_n, X_a)$—is end-to-end differentiable. Previously, this has allowed MMD to act as a loss function to train generative models by minimizing the distance between the generated and the natural distributions (Sutherland et al., 2017). Inspired by this, we design the following objective function for SIA that finds the set of perturbations Δ, where $X_a = \{x_j + \Delta_j \mid x_j \in X_0,\ \Delta_j \in \Delta,\ j = 1 \cdots m\}$ which we shortly denote as $X_0 + \Delta$.

$$\min_{\Delta} \gamma \text{MMD}_i(X_g, X_0 + \Delta) + \sum_{j=1}^{|X_0|} \mathcal{L}(\hat{y}(x_j + \Delta_j), T_j)$$

The first term aims to keep the AE and NE distributions close with respect to the i^{th} layer representations. The attacker samples a random set of NEs to construct X_g and uses it during the MMD computations as a *guide set*. To prevent overfitting, we sample three different guide sets and take the average MMD loss on them. The attack ultimately aims to defeat DDs by making X_a statistically equivalent to the guide sets. We sample the guide sets such that the predicted class labels of the samples in X_g match T, the target labels of the attack. We tune the kernel using the median heuristic, i.e., the median pairwise distance between the points in X_g and $X_0 + \Delta$.

The second loss term ensures that adding the corresponding perturbation, $\Delta_j \in \Delta$, to each initial sample, $x_j \in X_0$, changes the model's prediction into the target class, $T_j \in T$. We use cross-entropy as the loss function $\mathcal{L}$ in the second term. In this section, we perform targeted attacks by selecting $T_j \neq \hat{y}(x_j)$ randomly from the list of available classes.

The hyper-parameter γ balances between the two terms to reflect the attacker's priorities ($\gamma \geq 0$). As γ decreases the second term will start dominating at the risk of increasing the detectability; whereas as γ increases the ASR of the AEs might start dropping.

For its simplicity, we use the ℓ_∞-norm bounded PGD attack ($\|\Delta\|_\infty < \epsilon$) to minimize our objective. Unless specified otherwise, we set $\epsilon = 0.03$, following prior work (Madry et al., 2018). We perform 200 PGD iterations, use a scheduler that periodically reduces the step size, and craft 250 AEs at once as a batch. Note that SIA objective can be integrated into any other distance metric and gradient-based attack method, which we leave for future work.

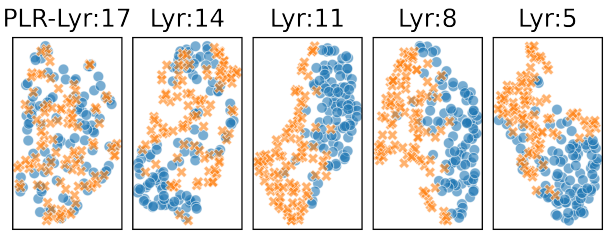

Figure 2. The reprs. of AEs crafted by SIA against the PLR.

Applying SIA against the PLR. We apply SIA against the PLR (17th layer) and visualize the resulting representations in Figure 2. We search $\gamma \in [0, 128]$ to find the value that minimizes SAMMD's DR while still achieving ∼100% ASR. Compared to Figure 1, we first observe the closeness between AE and NE distributions at the PLR. As a result, SAMMD fails to detect these AEs with 100 samples, i.e., $\text{DR}_{100} = 1\%$; whereas it could detect the standard attacks perfectly with only 50 samples. Note that SAMMD focuses only on the PLR and consumes a total of $2m$ AEs to obtain DR_m as it uses half of them for training its MMD kernel. Therefore, we also focus on the layerwise DDs (§2.1), which tune the kernels heuristically and can operate on any layer.

Turning our attention to the other layers reveals that the AEs are still detectable. Layerwise DDs obtain between 80% and 100% DR_{100} up until the 14th layer, after which the DR drops rapidly. Although applying SIA against the PLR breaks SAMMD (the most recent DD), it fails to craft statistically indistinguishable AEs at the remaining layers.

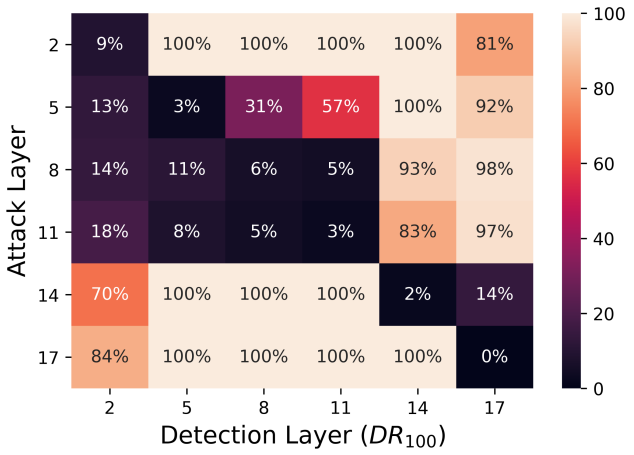

Figure 3. Crafting AEs with SIA against one layer and measuring the detection rate at the other layers.

SIA against one layer is still detectable at others. Here, we perform SIA against the i^{th} layer and measure the DR of a layerwise DD applied at the j^{th} layer. Figure 3 presents the results for a uniformly spaced subset of layers. For example, attacking the 5th layer drops the DR at this layer to 3% whereas the same AEs are still detectable at the 14th layer with 100% DR. Note how attacking certain layers also defeats the detection at some other layers, e.g., 8th and 11th layers. We hypothesize that this pattern arises from the architecture and the feature diversity among the layers (Kaya et al., 2019). The layers in the same group, e.g., 8th and 11th, have the same feature map scale and, therefore, extract similar features. Prior representation similarity metrics (Kornblith et al., 2019) also report similar findings. Ultimately, SIA perturbations against one layer create outlier representations at a *dissimilar* layer. Overall, these results lead us to conclude that we have to attack multiple layers jointly to craft indistinguishable AEs across the model.

Attacking all layers. Our previous formulation only attacks a single layer, which has failed to evade detection at others. To jointly attack all layers, we update the first term of SIA objective as $\sum_{i=1}^{N} \gamma_i \text{MMD}_i(X_g, X_0 + \Delta)$, where N is the number of layers. This minimizes the weighted sum of the MMD loss from all layers, with γ_i as the weight of the i^{th} layer. We opt for attacking all layers for simplicity as attacking only the dissimilar layers has not led to any improvements. We experimentally find that assigning equal weights to each layer performs well, i.e., $\gamma_i = \gamma$ for $i \in \{1, \cdots, N\}$. To find the ideal γ, we search in the range $[0, 128]$, while keeping the ASR above 95%. This allows us to craft successful AEs that are also indistinguishable.

Figure 4 suggests that the resulting AEs with 97% ASR follow the same distribution as the NEs across the model. Layerwise DDs at each layer result in DR_{100} between 0% and 3%, i.e., the AEs are not detectable at any layer. SAMMD also fails to detect the AEs we craft as it obtains less than 1% DR_{100}, averaged over 10 runs.

Regarding the impact of γ, we see the following trend that confirms our intuition. When we set $\gamma = 1$, the AEs we craft have 100% ASR, however, SAMMD achieves a non-trivial 20% DR_{300} against these AEs. As the sweet-spot, we have presented the results for $\gamma = 32$, where DR_{300} is 6% and the ASR is 97%. On the other extreme, when $\gamma = 128$, the DR_{300} is only 0.3%, however, the ASR drops to 58%, as we expected.

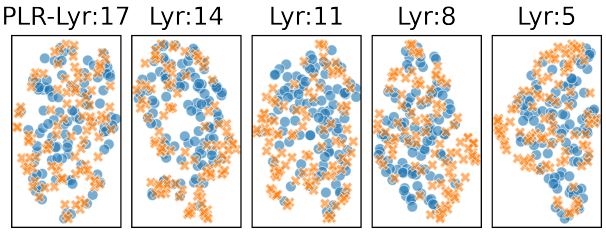

Figure 4. The reprs. of AEs crafted by SIA against all layers.

Runtime performance. The main overhead SIA has over PGD is computing MMD between the adversarial and guide sets with respect to the model's representations at each layer. Runtime of MMD computation grows linearly, quadratically and linearly with the number of dimensions in the representations, the number of samples in the sets and the number of layers in the model, respectively. This results in around 30–40% slower attack iterations compared to PGD. For example, against ResNet-50/CIFAR-10, VGG-16/CIFAR-10 and ResNet-50/TinyImageNet models (see Section 4.1 for the details of these architectures and datasets), crafting 1000 samples on a commodity GPU takes SIA (*200 iterations*) 10, 4 and 30 minutes, respectively; whereas it takes PGD (*50 iterations*) 1.5, 0.5 and 5 minutes.

3.3. Comparison With the Feature-Level Attack (FLA)

Similar to our attack, the FLA (Sabour et al., 2016) aims to craft AEs that match the representations of NEs at a layer. For an initial sample x_n, it randomly selects a natural guide sample x_g and crafts the perturbation δ that minimizes $\|F_i(x_n + \delta) - F_i(x_g)\|_2$. The FLA has been an effective attack against typical detectors that operate on individual samples (Tramer et al., 2020). To evaluate whether it can also defeat DDs, we apply the ℓ_∞-norm bounded FLA with random target labels against each layer. However, regardless of the parameters, layerwise DDs against the FLA result in over 60% DR_{100} at most layers. We present the detailed results in Appendix E. Overall, although the AEs crafted by the FLA evade individual detectors, collectively they are from a different distribution than the NEs.

Further, our next experiment suggests that attacking mul-

tiple layers simultaneously with the FLA might also be challenging. We first randomly sample NEs from the *ship* and *horse* classes and compute the pairwise ℓ_2 distances between their representations at different layers. For each ship sample, we then find the Spearman's rank correlation between its distances to the horse samples at layers i and j. Finally, for all ship samples, we take the average of this measurement which results in a metric we denote by $COR_{(i,j)}$. Having $COR_{(i,j)}$ close to one implies that if two samples are close at layer i, they are close at layer j as well. However, we see that the distances are poorly correlated between certain layers. For example $COR_{(2,5)} = 0.77$, whereas $COR_{(2,14)} = 0.16$ and $COR_{(5,8)} = 0.81$ whereas $COR_{(5,14)} = 0.29$. This experiment produces consistent results for other classes as well. As the FLA selects only a single guide sample for each AE, an AE that is close to the guide at one layer might fail to be close to it at another. In Appendix C, we present a similar layerwise analysis for SIA to show that it crafts an AE by implicitly selecting different guides at different layers.

4. Empirical Evaluation

We evaluate our attack on image classification tasks with CNNs, where adversarial attacks are most well-studied. Our main focus in this section is evaluating SIA against DDs. We consider individual AE detectors and other detection scenarios in the following sections.

4.1. Experimental Setup

Datasets. We experiment on two popular datasets: CIFAR-10 (Krizhevsky et al., 2009) and Tiny-ImageNet (Deng et al., 2009). CIFAR-10 consists of 32×32 pixels colored natural images (scaled between 0 and 1), drawn from 10 classes, each of which has 5K training and 1K testing samples. Tiny-ImageNet consists of a subset of ImageNet images resized at 64×64 pixels. There are 200 classes, each of which has 500 training and 50 testing images. For both datasets, we randomly select and hold out 10K training samples, which we will use to model the distribution of NEs.

Architectures and hyper-parameters. We run most of our experiments on a standard ResNet-50 (He et al., 2016) architecture. The model has 16 convolutional blocks, a pooling layer and a fully connected layer that produces the logits. We also evaluate MobileNetV2 (Sandler et al., 2018), with 19 blocks, and VGG-16 (Simonyan & Zisserman, 2015), with 14 blocks. We augment the training data by using padding, random cropping, RGB intensity scaling and random horizontal mirror. Our models are comparable to the models in the prior work, e.g., ResNet-50 models on CIFAR-10 and Tiny-ImageNet reach 94% and 65% accuracy, respectively.

Attack details. We craft 1000 targeted AEs, starting from a random subset of the test samples as our initial set X_0. We select the target labels T randomly from the list of available classes. In Appendix D, we also show the results on selecting the target T_j for $x_j \in X_0$ as the *next* class, i.e., $T_j = \hat{y}(x_j)+1 (\text{mod } \mathcal{K})$. These popular strategies reflect the average-case difficulty in targeted AEs (Ma & Liu, 2019).

Detector details. Against our attack, we apply the layerwise DDs at each layer and report the average and the highest detection rates (DR) among them. We observe that combining the p-values from each layer using popular methods from Fisher et al. (1948) or Brown (1975) does not improve the DR over the best layer. We also evaluate SAMMD (Gao et al., 2021) as it is the state-of-the-art DD. To understand how consuming more samples affects DDs, we report DR_{100}, DR_{200} and DR_{300}. These detectors have less than 5% (false) DR when they operate on NEs instead of AEs.

4.2. White-Box Scenarios

In Table 1, we present the detection results against SIA in the white-box setting, i.e., the attacker has full access to the victim model. We see how SIA evades the DDs across the board, while still achieving a high ASR. The low DR of layerwise DDs shows that the AEs we craft are not detectable at any layer. The more complex the task gets (e.g., larger inputs or more classes), performing adversarial attacks becomes easier as the models become more sensitive to perturbations (Balaji et al., 2019). As a result, our attacks against Tiny-ImageNet models are even more successful and SAMMD completely fails. Moreover, in Appendix E, we extensively compare SIA with three baseline attacks—PGD, FLA and AutoAttack (Croce & Hein, 2020)—in terms of detection rates. This comparison shows that DDs can consistently detect these attacks with 100% DR, with less than 100 samples.

Table 1. **The performance of DDs against SIA**. Column [A] includes three architectures: **R**esNet-50, **V**GG-16 and **M**obileNet. [AVG LYR] presents the average DR across the layers of the model and [MAX LYR] presents the highest DR among all layers.

A	ASR	AVG LYR	MAX LYR	SAMMD
		RANDOM CLASS TARGETED		
		CIFAR-10		
		$\text{DR}_{100\,/\,200\,/\,300}$	$\text{DR}_{100\,/\,200\,/\,300}$	$\text{DR}_{100\,/\,200\,/\,300}$
R	97%	1 / 2 / 3 %	3 / 5 / 7 %	0 / 2 / 6 %
V	94%	1 / 1 / 1 %	3 / 4 / 4 %	0 / 5 / 28 %
M	100%	3 / 2 / 3 %	7 / 5 / 9 %	0 / 0 / 1 %
		TINY-IMAGENET		
R	100%	3 / 4 / 4 %	9 / 8 / 9 %	0 / 0 / 0 %
V	100%	2 / 2 / 3 %	6 / 5 / 9 %	0 / 0 / 0 %
M	100%	2 / 4 / 2 %	7 / 8 / 6 %	0 / 0 / 2 %

Applying SIA against a robust model. Robustness aims to prevent AEs by making the model less sensitive to perturba-

tions (Szegedy et al., 2014). Although it has been studied separately from detection, recent work has formed a connection between the two approaches (Yin et al., 2020; Tramèr, 2021). Here, we apply SIA and standard PGD against a ℓ_∞-norm adversarially-trained robust model, provided by Madry et al. (2018). When $\epsilon = 0.03$, SIA and PGD craft AEs with 42% ASR, on which SAMMD obtains 37% and 100% DR_{300}, respectively. A higher ϵ increases the ASR without making SIA fully detectable, e.g., when $\epsilon = 0.07$, ASR is 82% and DR_{300} is 46%. This shows that PGD is still easily detectable even at a low ASR level, whereas SIA can still evade DDs that rely on representations from robust models.

Giving more samples to the detectors. A DD performs better as the number of samples available to it increases. Here, we experiment with our ResNet-50 model on CIFAR-10 and quantify how many AEs a DD needs to be able to consistently detect SIA. When SAMMD consumes 2000 AEs (1000 for kernel tuning and 1000 for testing), it achieves 78% DR_{1000}, with 4% false DR. The highest DR_{1000} among the layerwise DDs is 27%, with around 8% false detection rate. Note that these DDs have a perfect (100%) DR against the standard attacks with only 50 samples. As a realistic defender might only have a limited number of AEs, this casts doubt on whether detecting SIA is practical. We present the full detection trend in Appendix B.

4.3. Transferability of SIA

Transferability of AEs implies that they can still hurt a model that they were not crafted on (Tramèr et al., 2017; Liu et al., 2017). Even though white-box attacks are important to expose a vulnerability, transfer attacks, by requiring fewer assumptions, are more practical. Here, we investigate whether SIA can craft transferable AEs that also avoid detection at an unknown model. To this end, we first craft the AEs against a *surrogate* model and transfer them to a *target* model. We then measure (i) the ASR of these AEs against the target model and (ii) the performance of the layerwise DDs that operate on the target model's representations. As a baseline, we also craft AEs via PGD and tune its perturbation-bound ϵ to achieve a similar ASR to SIA.

In Table 2, we present the results on crafting AEs with random targets against CIFAR-10 models. We first note that, for similar ASR levels, SIA overall performs better than PGD against the detectors. Moreover, compared to the white-box setting, the attacks are more detectable and have less ASR. For example, when using two ResNet-50 models as both the surrogate and the target, SIA achieves 46% ASR. On the AEs, the DDs achieve at most 84% DR_{100} against SIA; which was only 3% in the white-box setting. Further, we see that AEs are more transferable, for example, from ResNet-50 to MobileNet than from ResNet-50 to VGG-16. The similarities between different DNN architectures explain the transfer success between different models (Liu et al., 2017). We believe combining SIA objective with the recent methods that craft more transferable AEs (Inkawhich et al., 2019; Wu et al., 2020) is a promising direction to boost the transferability of SIA.

Table 2. **Transferability performance of SIA**. [S-T] column lists the architectures of the surrogate [S] and target [T] models.

S-T	SIA			PGD		
	ASR	MAX LYR $DR_{50\,/\,100}$	AVG LYR $DR_{50\,/\,100}$	ASR	MAX LYR $DR_{50\,/\,100}$	AVG LYR $DR_{50\,/\,100}$
R-R	46%	19 / 84%	4 / 20%	44%	100 / 100%	63 / 82%
R-V	24%	21 / 89%	7 / 42%	23%	99 / 100%	63 / 90%
R-M	41%	41 / 94%	14 / 56%	41%	100 / 100%	70 / 85%
V-M	46%	60 / 99%	13 / 58%	42%	100 / 100%	65 / 82%

Cross-statistic transferability. So far, we have applied DDs that perform a 2ST using MMD as their test statistic. As SIA also uses MMD to craft indistinguishable AEs, we ask whether these AEs would evade a 2ST that uses a different statistic. To this end, we apply a classifier two-sample test (**C2ST**) by Lopez-Paz & Oquab (2017) on the representations at each layer. A C2ST first constructs a dataset by pairing the samples in X_a and X_n with positive and negative labels, respectively. It then trains a binary classifier on this dataset and uses its accuracy on test samples to compute the p-value, which we use to obtain the DR. We use 300 samples to train the classifier and 300 samples to obtain the p-value. Applied on a CIFAR-10 ResNet-50 model, C2ST achieves at most 6% DR_{300} against the AEs we craft; whereas the MMD-based DD achieves 7%. This demonstrates that SIA is transferable to test statistics other than MMD.

5. Defeating Individual AE Detectors

Although we have formulated SIA against DDs, individual detectors (IDs) that inspect a single sample are more realistic and, as a result, more common (Carlini & Wagner, 2017). A typical ID quantifies how statistically similar a sample is to the known NEs and rejects it if this score is below a threshold. In this section, we aim to show how its formulation makes SIA effective against IDs as well.

We experiment with a ResNet-50 model on CIFAR-10, train the detectors on 10K NEs (the whole holdout set) and test them on a balanced set that contains 500 AEs and 500 NEs. We consider the AEs crafted by SIA (in Figure 4) and, as a baseline, by PGD (in Figure 1). Both attacks craft AEs that have near 100% ASR. We report the following popular metrics to assess the detection performance: (i) the false positive rate (FPR) at 95% true positive rate (**FPR95**); and (ii) standardized partial area under the receiver operating characteristic curve below 20% FPR (**pAUC**). These metrics reflect the realistic requirements of a detector, e.g., low FPR and high TPR, and remove the need for tuning the threshold

manually. The scores for a chance level detector are 0.5 pAUC and 95% FPR95. For the perfect detector, they are 1.0 pAUC and 0% FPR95.

We evaluate the following four detectors from the literature.

Density artifacts (Feinman et al., 2017) is one of the early detectors which assumes that the AEs lie far from the natural data manifold. It trains a kernel density model based on the PLR of the holdout samples and rejects a sample if its density is smaller than a threshold.

I-Defender (Zheng & Hong, 2018) characterizes the properties of the natural data distribution at the PLR using a Gaussian mixture model (GMM) for each class. It rejects a sample if its likelihood estimated using GMMs is less than a threshold. For this detector, we report the average results over all classes.

The odds are odd (Roth et al., 2019) relies on the distribution of a DNN's logit values, in particular, it assumes that the logits on AEs are less robust to noise than on NEs. Prior work has shown that the adaptive evaluation in this paper is incomplete and designed an attack against the defense (Tramer et al., 2020). We select this method to demonstrate how SIA, with no additional effort, can evade detectors that apply input noise.

JTLA (Raghuram et al., 2021) is the most recent detector we experimented on, using the implementation by its authors. JTLA (stands for joint statistical testing across DNN layers for anomalies) computes statistics regarding whether a given sample is similar to NEs in terms of its nearest neighbors at a layer. It then aggregates the statistics from all layers into the detection score for this sample.

5.1. Evaluation

We present the results in Table 3. We see that all detectors have worse performance against SIA, compared to their performance against PGD. Moreover, Density detector has near-chance level performance against SIA. Regarding JTLA, we observe that it already performs poorly against PGD, compared to other detectors. As a sanity check, we apply JTLA on the AEs crafted by DDN (in Figure 1), which results in over 0.9 pAUC. This shows JTLA has inconsistent performance even against standard attacks. Overall, these results show that SIA, without customization, can be used as a benchmark to evaluate individual detectors as well as distributional detectors.

6. Attacking Other Detection Scenarios

The previous sections show how SIA evades methods designed to detect AEs. Here, we apply SIA in other detection scenarios to demonstrate its flexibility. In Appendix F, we display some AEs crafted in this section.

Table 3. **The performance of four individual detectors**. We present the detection results against PGD and SIA by reporting two metrics: FPR95 (lower is better) and pAUC (higher is better).

	PGD		SIA	
Detector	FPR95	pAUC	FPR95	pAUC
Density	40%	0.75	95%	0.53
I-Defender	34%	0.77	92%	0.61
Odds	56%	0.71	88%	0.65
JTLA	83%	0.62	90%	0.60

6.1. Dataset Shift Detectors

Seemingly subtle changes in the data distribution are known to hurt the performance of modern DNNs (Hendrycks & Dietterich, 2018). Prior work has developed methods to detect such *dataset shifts* that give a critical opportunity for practitioners to act (Rabanser et al., 2019). We focus on two different shift scenarios and highlight the security risks of shift detectors as a new threat model.

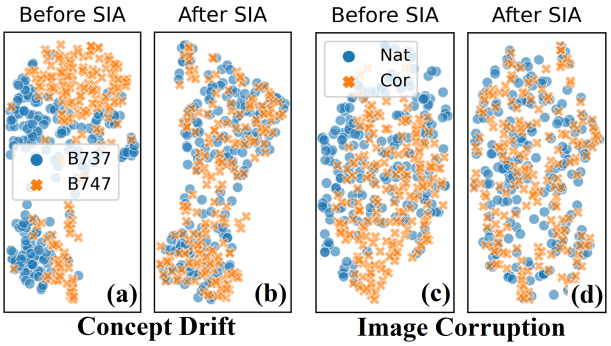

Figure 5. Attacking dataset shift detection with SIA.

Concept drift detection. Concept drift refers to the problem when the relationship between the underlying data distribution and concept being learned changes over time. Adapting the setup by Kirchler et al. (2020) and using the dataset by Maji et al. (2013), we consider the drift from the images of the 'Boeing-737' aircraft to the 'Boeing-747' aircraft. In this setup, the detector applies a C2ST on the features it extracts from the images using the PLR of a pre-trained DNN on ImageNet. Figure 5 (*a*) shows how the two aircraft types have different feature distributions. As a result, with 100 samples from each distribution, the detector easily detects the drift with 100% DR. To evade this detector, we include the B-737 samples in our guide sets and perturb the B-747 samples using SIA against the DNN. We omit the second

term in SIA objective as the detector discards the DNN's predictions. Figure 5 (*b*) hints that the perturbed B-747 and the B-737 distributions are the same. This brings the DR down to 0%, even with 300 samples, and confirms that SIA has made the drift detection ineffective.

Corruption detection. Another type of shift occurs when environmental factors start corrupting the inputs. As a result, image corruptions, such as fog or snow, are shown to hurt the performance of DNNs (Hendrycks & Dietterich, 2018). This makes detecting such corruptions crucial in safety-critical DNN applications (Rabanser et al., 2019). In our setup, we consider the snow corruption (Hendrycks & Dietterich, 2018) applied on the test samples in Tiny-ImageNet. To detect the shift, we apply a DD on the PLR of a ResNet-50 Tiny-ImageNet model. Figure 5 (*c*) shows how the corrupted samples follow a different distribution than uncorrupted NEs. As a result, the DD reaches 100% DR_{100} and effectively detects the shift. Similar to the previous scenario, we use NEs in our guide sets and apply SIA to perturb the corrupted samples. The resulting perturbed samples have the same distribution as the NEs, as we show in Figure 5 (*d*). This leads to 0% DR_{300} and prevents the detection.

6.2. Out-of-Distribution (OOD) Detection

Due to its importance in enhancing the safety of DNNs, detecting OOD inputs has received much attention lately. ReAct (Sun et al., 2021), the state-of-the-art OOD detector, rectifies the penultimate-layer activations at an upper limit c. The authors claim that this empirically and theoretically leads to a better separation between the energy scores (Liu et al., 2020b) of OOD and in-distribution inputs. Following the original setup, we apply ReAct on a ResNet-50 CIFAR-10 model and use the SVHN (Netzer et al., 2011) dataset as the OOD inputs. An energy-based detector achieves 24% FPR95, which goes down to 19% with ReAct. We then perturb the SVHN samples with SIA, using CIFAR-10 samples in the guide sets. We reduce the perturbation-bound to $\epsilon = 0.01$, instead of $\epsilon = 0.03$, for better imperceptibility. As a result, ReAct achieves 95% FPR95 on the perturbed OOD samples, i.e., chance level performance. This shows that SIA can turn OOD inputs into in-distribution inputs, which exposes the vulnerabilities of many OOD detectors that rely on representation statistics.

7. Conclusions

We introduce the *statistical indistinguishability attack* (SIA) as a general-purpose adaptive attack against a wide range of detectors, including distributional and individual ones. SIA essentially crafts adversarial examples (AEs) that closely follow the distribution of natural samples with respect to hidden layer DNN representations. This allows our attack to defeat detectors that rely on observing statistical differences between adversarial and natural inputs. Thanks to its generic formulation, SIA compromises detection methods in various settings, with little-to-no customization. As detecting AEs is becoming increasingly critical for safeguarding DNNs, we believe our work will provide a reliable baseline tool to evaluate the security of detectors against adversaries who wish to avoid them.

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Balaji, Y., Goldstein, T., and Hoffman, J. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

Brown, M. B. 400: A method for combining non-independent, one-sided tests of significance. *Biometrics*, pp. 987–992, 1975.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 3–14, 2017.

Chwialkowski, K. P., Sejdinovic, D., and Gretton, A. A wild bootstrap for degenerate kernel tests. In *Advances in neural information processing systems*, pp. 3608–3616, 2014.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Fisher, R. A. et al. 224a: Answer to question 14 on combining independent tests of significance. 1948.

Fogla, P., Sharif, M. I., Perdisci, R., Kolesnikov, O. M., and Lee, W. Polymorphic blending attacks. In *USENIX security symposium*, pp. 241–256, 2006.

Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., and Sugiyama, M. Maximum mean discrepancy test is aware of adversarial attacks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 3564–3575, 2021.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19: 513–520, 2006.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Hong, S., Kaya, Y., Modoranu, I.-V., and Dumitras, T. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. In *International Conference on Learning Representations*, 2021.

Inkawhich, N., Wen, W., Li, H. H., and Chen, Y. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7066–7074, 2019.

Kaya, Y., Hong, S., and Dumitras, T. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, pp. 3301–3310. PMLR, 2019.

Kirchler, M., Khorasani, S., Kloft, M., and Lippert, C. Two-sample testing using deep learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1398. PMLR, 2020.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012.

Kübler, J., Jitkrittum, W., Schölkopf, B., and Muandet, K. Learning kernel tests without data splitting. *Advances in Neural Information Processing Systems*, 33, 2020.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. Learning deep kernels for non-parametric

two-sample tests. In *International Conference on Machine Learning*, pp. 6316–6326. PMLR, 2020a.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020b.

Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations*, 2017.

Lopez-Paz, D. and Oquab, M. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

Ma, S. and Liu, Y. NIC: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019)*, 2019.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Maji, S., Kannala, J., Rahtu, E., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. Technical report, 2013.

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*, 2017.

Miller, D., Wang, Y., and Kesidis, G. When not to classify: Anomaly detection of attacks (ADA) on DNN classifiers at test time. *Neural computation*, 31(8):1624–1670, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Paxson, V. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23-24):2435–2463, 1999.

Rabanser, S., Günnemann, S., and Lipton, Z. C. Failing loudly: An empirical study of methods for detecting dataset shift. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 1396–1408, 2019.

Raghuram, J., Chandrasekaran, V., Jha, S., and Banerjee, S. A general framework for detecting anomalous inputs to DNN classifiers. In *International Conference on Machine Learning*, pp. 8764–8775. PMLR, 2021.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

Roth, K., Kilcher, Y., and Hofmann, T. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, pp. 5498–5507. PMLR, 2019.

Sabour, S., Cao, Y., Faghri, F., and Fleet, D. J. Adversarial manipulation of deep representations. In *ICLR (Poster)*, 2016. URL http://arxiv.org/abs/1511.05122.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Shafahi, A., Huang, W. R., Studer, C., Feizi, S., and Goldstein, T. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34, 2021.

Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A. J., and Gretton, A. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR (Poster)*, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Tramèr, F. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML 2021 Workshop on the Prospects and Perils of Adversarial Machine Learning*, 2021.

Tramer, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32:5866–5876, 2019.

Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33, 2020.

Wagner, D. and Soto, P. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 255–264, 2002.

Warrender, C., Forrest, S., and Pearlmutter, B. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE symposium on security and privacy (Cat. No. 99CB36344)*, pp. 133–145. IEEE, 1999.

Wu, D., Wang, Y., Xia, S.-T., Bailey, J., and Ma, X. Skip connections matter: On the transferability of adversarial examples generated with ResNets. In *International Conference on Learning Representations*, 2020.

Yin, X., Kolouri, S., and Rohde, G. K. GAT: Generative adversarial training for adversarial example detection and robust classification. In *International Conference on Learning Representations*, 2020.

Zheng, Z. and Hong, P. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7924–7933, 2018.

A. Attack Details

(i) Projected Gradient Descent (PGD) attack (Madry et al., 2018), is an iterative attack that perturbs the inputs according to the sign of the gradients. After each iteration PGD bounds the ℓ_∞-norm of the distance between the AE and the NE. PGD has been the de facto standard attack for training and evaluating robust models. We apply the PGD attack for 50 iterations and set the step size as $\epsilon/\sqrt{50}$, where ϵ is the perturbation bound. Our proposed attack, SIA, and the baseline attack, the FLA, also use PGD iterations but we apply them for 200 iterations instead to ensure better performance. For our experiments in Section 3 and in Section 4, we set $\epsilon = 0.02$, which crafts AEs that reach 100% ASR.

(ii) Decoupling Direction-Norm (DDN) (Rony et al., 2019) is a recent attack that iteratively minimizes the ℓ_2-norm of the perturbation that changes the model's prediction. It has been shown to be competitive to the popular Carlini&Wagner attack, while being more computationally efficient. Typically, this attack does not have any bound on the maximum allowed ℓ_2 norm of the perturbations, which allows it to always have 100% ASR. We apply the DDN attack for 200 iterations. For our experiments in Section 3, the average ℓ_2 perturbation for an AE is 0.18 when DDN achieves 100% ASR.

(iii) Sparse ℓ_1 Descent (SLIDE) (Tramer & Boneh, 2019) first observes that the default ℓ_1 version of PGD is highly inefficient as each iteration updates only one pixel. Thus the authors design a new attack with finer control over the sparsity of the update step that also uses projection onto an ℓ_1 ball to bound the perturbations. We apply SLIDE for 50 iterations and set the sparsity level to 99%. For our experiments in Section 3, we set the perturbation-bound as $\|\delta\|_1 = 25$ to achieve 100% ASR.

B. Giving More Samples to the DDs

Figure 6 presents the full detection trend for the experiment we perform in Section 4.2 of our main paper.

C. The Layerwise Analysis of SIA

In Figure 7, we present images from a case study we conduct to understand the behavior of SIA. We visualize two clean initial samples from X_0 and the corresponding AEs SIA crafts against all layers of the DNN. We then find the holdout images *closest* to the initial clean, and adversarial, images with respect to the representations at different layers. This reveals two behaviors that make SIA perform better than the FLA in evading layerwise DDs.

First, for a clean image, the closest images at different layers are not classified into the same class as the clean image. For example, for the clean *sports car*, the closest image at the 2nd and 7th layers is an image of a *teapot*. These results align with our findings using the COR metric in Section 3.3 of our main paper. This behavior demonstrates why selecting a single guide per initial sample makes it challenging for the FLA to evade multiple layers.

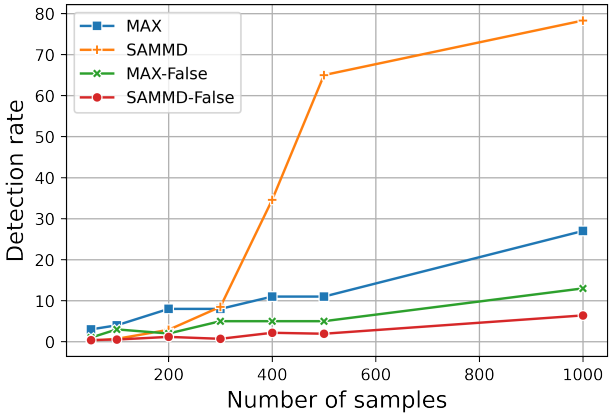

Figure 6. The detection performance of distributional detectors as a function of number of available samples. We experiment with a ResNet-50 model trained on CIFAR-10. We measure the false detection rates by providing NEs to the detectors instead of AEs.

Second, for an adversarial image, the closest images at different layers change from layer to layer. For example, for the adversarial *bow tie* (originally a *candle*), we see the closest images are *orange* and *lampshade* and *bow tie* at the 7th, 12th and the 17th layers, respectively. This shows that SIA implicitly selects different guide samples at different layers to craft AEs.

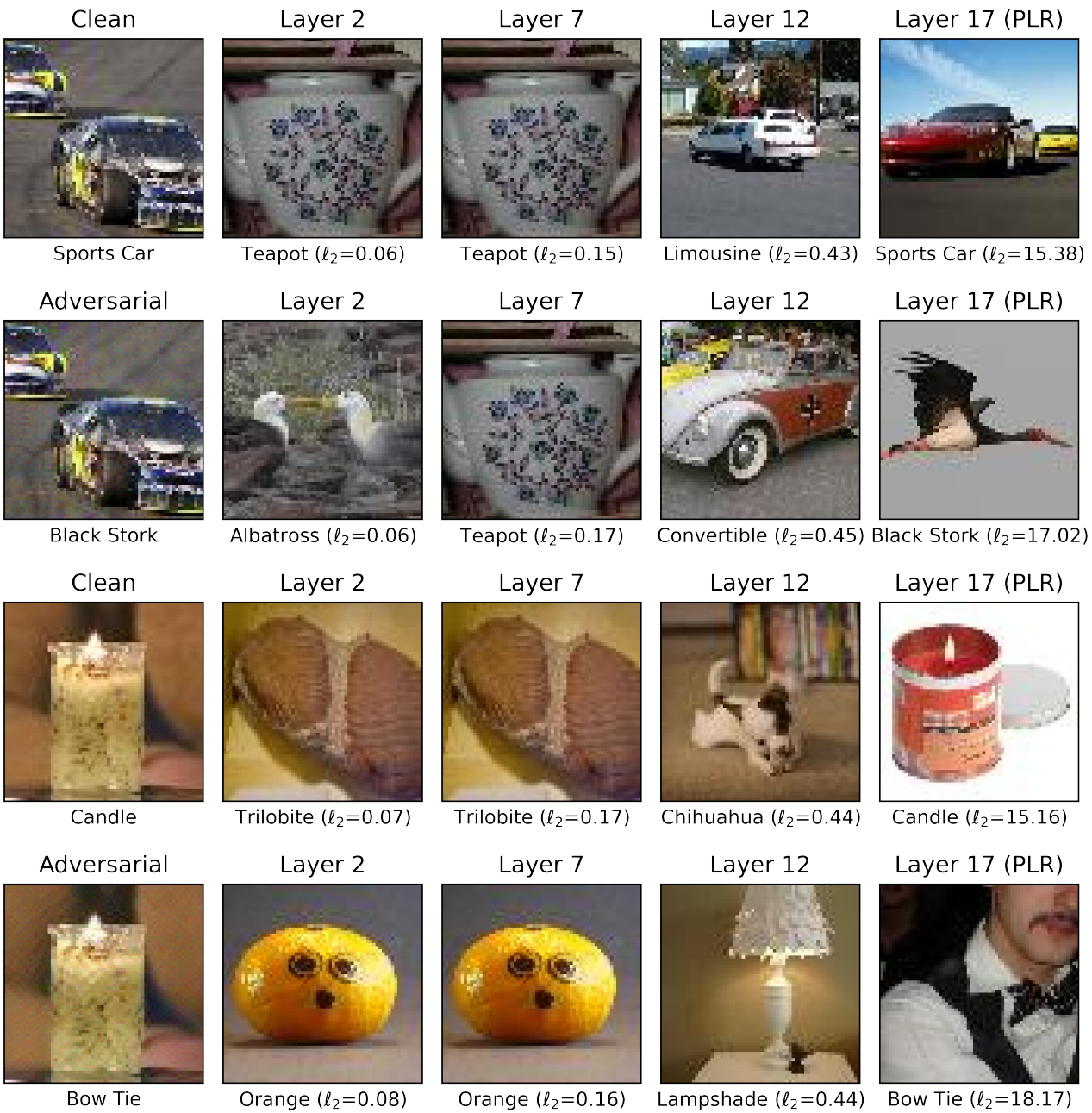

Figure 7. Displaying the holdout images closest to initial natural and the corresponding adversarial examples at different layers. We craft the AEs using SIA against all layers of a ResNet-50 model trained on Tiny-ImageNet. At the bottom of each image, we display the model's predicted class on that image. We find the closest images by measuring the ℓ_2 distance between two images with respect to the model's representations.

D. Results on Next-Class Targeted SIA

We present the detection results in Table 4 for this target selection strategy. We see that the distributional detectors perform poorly against these AEs, similar to selecting the targets randomly as we show in Section 4.2 of our main paper.

Table 4. **The performance of DDs against SIA**. Column [A] includes three architectures: **R**esNet-50, **V**GG-16 and **M**obileNet. [AVG LYR] presents the average DR across the layers of the model and [MAX LYR] presents the highest DR among all layers.

A	ASR	AVG LYR	MAX LYR	SAMMD
		NEXT CLASS TARGETED		
		CIFAR-10		
		$DR_{100\,/\,200\,/\,300}$	$DR_{100\,/\,200\,/\,300}$	$DR_{100\,/\,200\,/\,300}$
R	99%	0 / 3 / 2 %	2 / 7 / 4 %	0 / 3 / 11 %
V	94%	1 / 2 / 3 %	5 / 4 / 9 %	1 / 1 / 14 %
M	100%	4 / 3 / 3 %	10 / 8 / 7 %	0 / 0 / 2 %
		TINY-IMAGENET		
R	100%	3 / 4 / 7 %	6 / 9 / 14 %	0 / 0 / 0 %
V	100%	2 / 3 / 4 %	5 / 10 / 12 %	0 / 0 / 0 %
M	97%	2 / 4 / 5 %	5 / 12 / 13 %	0 / 1 / 1 %

E. Applying Distributional Detectors (DDs) Against PGD, FLA and AutoAttack

FLA (Sabour et al., 2016) results. We present the detection results in Table 5 for attacking different layers with the FLA. We set the ℓ_∞-norm perturbation-bound to $\epsilon = 0.03$, same as SIA. For an initial sample, x_n, we select the corresponding guide, x_g, randomly among the holdout samples such that $\hat{y}(x_g) = t$, where t is the attack's target class. We see how the FLA is successful against the deeper layers in evading the DDs; however, it fails in earlier layers. This aligns with Sabour et al.'s findings that its is more challenging to apply the FLA against earlier layers as they are less sensitive to perturbations. Moreover, attacking all layers with the FLA also cannot evade the DDs. These results demonstrate that the FLA is ineffective against DDs.

PGD (Madry et al., 2018) results. We present the detection results in Figure 8 for increasing perturbation-bounds (ϵ) of PGD. We see how the ASR increases as we increase ϵ and how, as a result, the AEs become more detectable. Even with low ASR levels when $\epsilon < 0.01$, the layerwise DDs still have moderate success against PGD.

AutoAttack (Croce & Hein, 2020) results. We present the detection results in Table 6 for increasing perturbation-bounds (ϵ) of AutoAttack. We found that maximizing *difference of logits ratio* loss, instead of cross-entropy, allows AutoAttack to be less detectable than PGD in the final layers. However, in the remaining layers, AutoAttack is still easily detected: e.g., 100% detection rate at the 11th layer (SIA has only 2% detection rate). Overall, these results validate our formulation to evade detection at multiple layers.

Table 5. **The detection performance (DR_{100}) of DDs against the FLA**. We attack individual layers with the FLA to craft targeted AEs with random targets. We apply the layerwise DDs and measure the detection rates at the attacked layers. We also report the average and the highest DRs among the layerwise DDs. In the last row (ALL), we attack all the layers in the DNN. We experiment with a ResNet-50 model on CIFAR-10.

ATTACKED LAYER	ASR	ATTACKED LAYER DR	AVG LYR	MAX LYR
2	98%	40%	89%	100%
5	98%	63%	88%	100%
8	100%	91%	93%	100%
11	100%	100%	94%	100%
14	100%	0%	75%	100%
17	100%	0%	79%	100%
ALL	99%	-	78%	100%

Table 6. **The detection performance (DR_{100}) of DDs against AutoAttack**. We apply the attack with different perturbation bounds (ϵ) and measure the detection rate at different layers of the model. We experiment with a ResNet-50 model on CIFAR-10.

DETECTION LAYER					
2	**5**	**8**	**11**	**14**	**17**
$\epsilon = 0.005$, ASR=79%					
4%	10%	35%	36%	10%	4%
$\epsilon = 0.01$, ASR=99%					
5%	92%	100%	100%	63%	10%
$\epsilon = 0.02$, ASR=100%					
6%	100%	100%	100%	100%	64%

F. Displaying the AEs

Here, we present some of the AEs we crafted in Section 6 of our main paper.

Concept drift detection. We present the images in Figure 10.

Corruption detection. We present the images in Figure 11.

OOD Detection against ReAct. We present the images in Figure 9.

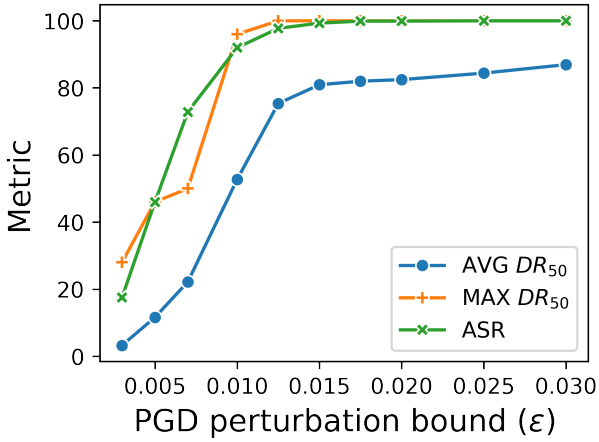

Figure 8. Detection performance (DR_{50}) of the layerwise detectors against PGD with increasing perturbation-bounds (ϵ). AVG and MAX report the average and the highest DR of the layerwise DDs. We experiment with a ResNet-50 model trained on CIFAR-10.

Figure 9. The AEs crafted by SIA against the out-of-distribution (OOD) sample detector. The in-distribution samples are from the test set of CIFAR-10, the OOD samples are from SVHN. We perturb the OOD samples using SIA to avoid detection.

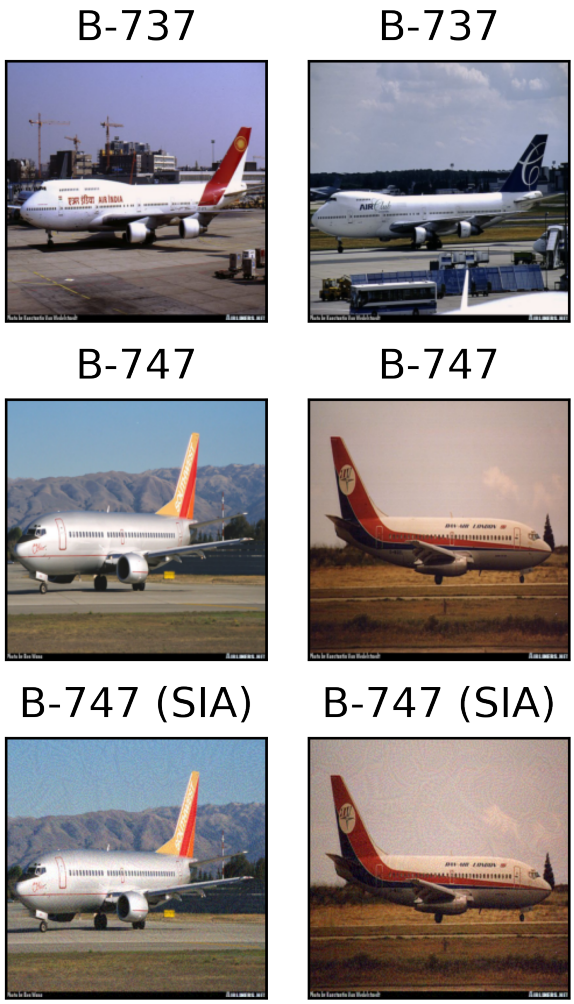

Figure 10. The AEs crafted by SIA in the concept drift detection scenario. The images of the Boeing-737 are from the original distribution and the images of the Boeing-747, which are perturbed by SIA, are from the shifted distribution.

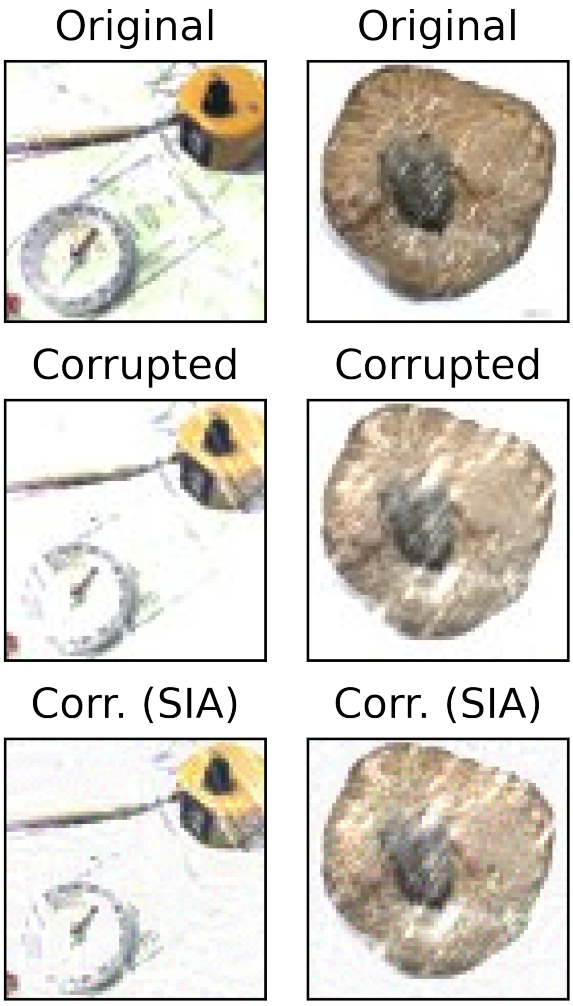

Figure 11. The AEs crafted by SIA in the corruption detection scenario. The original images are from the test set of Tiny-ImageNet. The corrupted images include the highest intensity *snow* from the dataset provided by Hendrycks & Dietterich (2018). We perturb the corrupted images using SIA to avoid detection.