Overcoming Open-Set Approaches to Adversarial Defense

Anonymous authors
Paper under double-blind review

Abstract

Machine learning (ML) models are increasingly proposed to replace or augment safety-critical sensor processing systems, yet their fragility to evasion attacks remains a well-documented open problem. This work analyzes a class of deep neural network defenses that add a none-of-the-above (NOTA) class as an open-set-inspired closed-set adversarial defense. We show that such approaches often appear far more robust than they are because standard adversarial attacks lack explicit handling for large auxiliary classes like NOTA–causing stopping criteria, target-selection, and objective function behaviors that mask true vulnerabilities. We formalize these issues in a taxonomy of evaluation pitfalls, adapt seven prominent adversarial attacks to eliminate them, and show that adding a NOTA class alone, does not solve the core challenge of defending DNNs against evasion attacks. We release our adapted attack suite to enable more rigorous future evaluations of open-set-inspired defenses.

1 Introduction

Recent years have seen a steep increase in the number of successful applications of Deep Neural Networks (DNNs) across the sciences, industry, and business. This technology has enabled strides forward in areas as disparate as machine vision (Krizhevsky et al., 2012), neuroimaging analysis (McClure et al., 2019), astronomy (Valizadegan et al., 2022), cancer diagnosis (Savage, 2020), protein folding (Callaway, 2020), and natural language processing (Brown et al., 2020). Despite these advances, efforts to leverage DNN technology in safety-critical systems have been hampered by the fact that current approaches create models that are highly susceptible to deception, particularly deception in the form of what are known as evasion attacks or adversarial examples. This is a well-documented open problem that persists to today (Carlini, 2024).

To address this, many defense methods have been proposed to increase the robustness of DNNs to adversarial examples (Costa et al., 2024). The most widely implemented type of adversarial defense is adversarial training (Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2017). These defenses seek to create adversarial examples for a particular DNN from clean training examples. These adversarial examples are then labeled with the same label as the clean training examples used to generate them and are then used to train that DNN. This is done with the goal of making the addition of adversarial noise not change the DNN's predicted label for an example. This robustness, however, may not defend against adversarial examples distant in input space from clean training examples (Costa et al., 2024).

One approach to addressing more distant adversarial attacks is to look to the open-set paradigm and create a None-of-the-Above (NOTA) class, a new class in addition to the existing classes in a particular dataset. In contrast to adversarial training, generated adversarial examples act as boundaries, are put into the NOTA class, and are used as training examples. The premise is that with carefully crafted NOTA augmentation examples, one can continuously seed the data-point-sparse space between data-dense regions of a classifier's input space throughout training and leverage the DNN to classify this vast space as NOTA. This approach is suggested as a potential remedy to the "inevitability" of adversarial examples under the standard DNN training paradigm (Shafahi et al., 2019). While covering all of the data-sparse regions of input space would be impractical, such approaches hold it might be sufficient to cover the regions that define the boundaries between and adjacent to classes and let the DNN generalize the remaining input space as NOTA.

Such defensive training strategies can be agnostic of the attack method or the neural network architecture. Instead, they seek to change the structure of how the input space is partitioned by the classifier through training with an additional NOTA class. The approach is modular and can be added to any DNN classifier and may augment most present or future defense strategies. The entire approach has a fixed computational cost which only occurs during training, similar to adversarial training defenses, thus making it an attractive potential solution. Barton (2018) first demonstrated these open-set approaches to closed-set adversarial robustness and deep neural network defense with *Boundary Padding* (BP), a data augmentation approach which sought to insert a "padding class" between various classes in input space by linearly interpolating between two images and subsequently labeling the resulting example as a padding class (i.e., NOTA). This defense was successful against several standard evasion attacks.

In this paper, we investigate, test, and evaluate this category of deep neural network defense by analyzing and testing boundary padding as well as a illustrative defense called *Adversarial NOTA Envelopment* (ANE), which seeks to additionally envelop data-dense regions of input space by creating NOTA examples in regions not already classified as NOTA. When evaluating an adversarial defense, it is critical that the evaluation does not lead to overconfidence in the effectiveness of a defense. To support this goal, leading researchers in machine learning security have put forward a number of best practices in evaluating deep neural network adversarial robustness. Evaluating potential DNN defenses against attacks that are not specifically adapted to maximally combat or thwart those defenses is of minimal value. Once a researcher has conceived and built a defense, it is then incumbent on them to "switch hats," and apply their full knowledge and effort to break the specific defense they are proposing, by altering attacks as necessary (Carlini et al., 2019). In keeping with this ethos, we analyze the mechanisms and interactions between many adversarial attacks and two NOTA defense approaches. As a result of this analysis, we provide the following contributions:

- Taxonomy of three recurring evaluation pitfalls for NOTA or open-set approaches to adversarial defense.
- Evaluation attack suite of seven NOTA-aware attack variants (code released).
- Case Study demonstrating that existing defenses–including a novel open-set approach that improves on previous methods–fail under proper evaluation.

Progress on open-set or "None-of-the-Above" approaches to adversarial defense is held back less by a lack of defensive ideas than by misaligned attack evaluations. A long line of work–from Athalye et al. (2018) through Tramer et al. (2020) to Suya et al. (2024)—shows that adversarial defenses judged "robust" often collapse once the attacker is allowed to (i) target the correct class subset, (ii) run until truly optimal distortion is reached, and/or (iii) optimize the defense's actual objective function. Without a standardized, defense-aware attack suite, researchers risk re-learning this lesson for every new defense modification or approach, burning GPU cycles on fixes that merely plug ad-hoc evaluation holes. By providing this taxonomy and a drop-in set of NOTA-aware attack variants, our work aims to raise the default evaluation bar: future NOTA defenses that survive this suite, or other attacks modified according to this taxonomy, will have demonstrated robustness to the three failure modes we identify, giving practitioners a higher-confidence starting point and allowing the community to focus on genuinely new vulnerabilities rather than repeating past mistakes.

2 Background

The DNN defenses we evaluate can be applied to many architectures. With respect to evasion attack adaptations, we introduce general concepts that can be applied to any existing attacks to prevent or mitigate NOTA-paradigm defense successes. Before introducing our adapted attack strategies, it is necessary to outline the evaluated models, defenses, and standard attacks.

2.1 Deep Neural Network Classifiers

Adversarial attacks are most commonly executed against DNN models. In general, a DNN classifier can be described as a function $f(x) : R^d -> R^c$. Where the input is $x \in R^d$, and the output is in R^c, and is often called the logits. In the image classification domain, input x is an $h * w * l$ pixel image such that $x \in [0, 1]^{hwl}$

and c is the number of object classes for the image. Object classes are denoted by integer codes ranging from 0 to $c-1$. The prediction of the DNN for an input x is given by the equation $y = \text{argmax}(f(x))$.

2.2 Evasion Attacks

Evasion attacks imperceptibly modify the input to a model to produce a change in classification from the clean (i.e. originally intended) class to some other, untrue class (Chakraborty et al., 2018). From this point forward, we refer to these simply as adversarial attacks (Goodfellow et al., 2015).

The general form of evasion attacks is as follows. Adversarial examples are crafted such that given a classifier f, identifying c distinct classes of objects $y_1, y_2, y_3 ... y_m$, and a clean input x, belonging to the class y_x, an input x' can be crafted such that $f(x') = y_z$ and $y_x \neq y_z$. This is accomplished by adding some perturbation δ_x to x such that $x + \delta_x = x'$, yet x' is still recognized by humans as belonging to the original class y_x (Goodfellow et al., 2015). When attacking a NOTA-defended DNN classifier f with an image x', an attack is successful only if the prediction is not the original class y_x or the NOTA class y_N. In other words, $f(x') \neq y_x \wedge f(x') \neq y_N$ for a successful attack. The specific original attack algorithms used and modified in this work are described in detail below.

2.2.1 Threat Model

Our threat model assumes the adversary's attack occurs after training and system deployment, (i.e., the adversary cannot manipulate training data). The adversary is assumed to be able to do one of two things. One, they can change the actual artifact in the real world. Examples of this approach would be donning anti-facial-recognition glasses to defeat identification systems (Sharif et al., 2016), applying an AI-camouflage pattern to a ship or tank to evade wide area motion imagery detection, or simply applying tape to precise positions on a stop sign to fool a self-driving car's image classification system into missing the sign (Eykholt et al., 2018). Two, an adversary with insider access to the data stream can change the direct input to the model by, for instance, altering pixels by imperceptible amounts to fool a classification system into misclassifying the pictured object (Goodfellow et al., 2015; Madry et al., 2017; Carlini & Wagner, 2017).

2.2.2 Projected Gradient Descent (PGD)

Projected gradient descent is a straight-forward attack that leverages the same optimization that makes DNNs possible in the first place. In untargeted PGD, adversarial examples, x' are discovered through gradient ascent and backpropagation (Madry et al., 2017). We let $\mathcal{L}(f(x), y_x)$ represent any loss function whose minimum results in $f(x) = y_x$. Gradient ascent is then employed to iteratively adjust pixels in x such that the loss, $\mathcal{L}$ maximally increases and the resulting image $f(x') \neq y_x$.[1] For each iterative step, x is adjusted thus:

$$x' = x + \alpha \ \text{sign}(\nabla_x \mathcal{L}(f(x), y_x)),$$

where the gradient vector $\nabla_x \mathcal{L}(f(x), y_x)$ is the rate of change of the loss, $\mathcal{L}$, and α is the learning rate. This procedure produces a perturbation for x that pushes the DNN's prediction away from the true class, y_x.

After each gradient update, x' is projected to be within an L_p-bound, ϵ, of x and be in the set $[0,1]^{hwl}$. The most common L_p-norms used in PGD, and most other evasion attacks, are L_2 and L_∞. $L_2(\delta_x)$ is the magnitude of the adversarial noise and $L_\infty(\delta_x)$ is the maximum of the absolute adversarial noise.

2.2.3 AutoPGD (APGD)

Noting that PGD is a frequently-used, computationally-cheap method to test adversarial robustness in DNN classifiers, Croce & Hein (2020b) identify two failure modes that can arise and give false assurance of robustness. These are: 1) using a fixed step size and 2) limiting the loss function to cross entropy (CE). To improve attack performance, they automate a process to identify a maximally effective step-size for PGD among other variables, as well as provide a new alternative loss function they named Difference Logits Ratio (DLR). Whereas the loss most often used in PGD is cross entropy, a shift-invariant loss, meaning the order of

[1] This is equivalent to gradient decent on the negative loss.

logits does not alter the output, AutoPGD introduces DLR, which is both shift and scale invariant. Scale invariant implies that rescaling its inputs by a non-zero constant will not change the loss value. This ensures that the learning process—and by extension attacks seeking adversarial examples using gradient descent—will not be sensitive to or affected by the scale of the input data. The DLR loss is defined as:

$$\mathcal{L}_{\text{DLR}}(f(x), y) = -\frac{z_y - \max\limits_{i \neq y}(z_i)}{z_{\pi_1} - z_{\pi_3}},$$

where each z is an individual logit from the output of $f(x)$, π is the ordering of the components of $\mathbf{z}$ in decreasing order. DLR has been reported to be sometimes better-performing than CE with respect to attack success and more stable than Carlini-Wagner loss, detailed below (Croce & Hein, 2020b).

2.2.4 Carlini-Wagner Attacks

Carlini-Wagner (CW) attacks (Carlini & Wagner, 2017) are a suite of attacks that use various optimization methods to find a minimum perturbation adversarial example according to some metric $D(x - x')$, usually L_2 or L_∞, subject to the perturbation leading to a successful adversarial example. This results in the optimization objective of

$$\begin{aligned} \text{minimize} \quad & D(x, x') + \lambda_c \cdot \mathcal{C}_{\text{CW}}(x') \\ & s.t. \quad x' \in [0,1]^n. \end{aligned}$$

The CW constraint, $\mathcal{C}_{\text{CW}}$, term often takes the form

$$\mathcal{C}_{\text{CW}}(x, y) = \text{ReLU}\big(z_y - \max_{i \neq y}(z_i) + \gamma\big),$$

where γ is the "confidence" and controls how much the logit of the highest non-clean logit, $\max(z_i)$, exceeds the logit for the clean class (z_y). Carlini and Wagner add the additional requirement of the *box constraint* (in order that x' be considered a valid image, all pixels must be in the range $[0, 1]$). As a further requirement, $f(x')$ is minimized, where f is an objective function producing a minimal value when $f(x') = y_t$ – i.e., when the target class is reached and the attack succeeds. The constant λ_c is obtained through binary search and is used to increment or decrement the weight of f. The attack can be targeted where a desired non-true class is provided or untargeted in which case the non-true class with the highest predicted logit z_i output is selected as the target class. Additionally, the attacker may choose a "confidence" level in the targeted version of the attack as well, denoted above as γ, which controls the margin by which an example target class logit must be driven to exceed the true class's logit.

2.2.5 Deepfool Attacks

Researchers Moosavi-Dezfooli, Fawzi and Frossard introduced a novel attack in 2016, called *DeepFool* (Moosavi-Dezfooli et al., 2016). The attack is originally conceived and implemented as an L_2-based attack, but can be adapted to any L_p metric. The DeepFool algorithm is a greedy algorithm that attempts to approximate the minimum distance to the nearest decision boundary then cross it and produce an adversarial example. The approach does not purport to guarantee the smallest possible perturbation, but has been found in practice to yield very small perturbations. The authors believe these perturbations to be good approximations of the minimum. It is not a targeted attack, instead seeking to find the closest region of input space producing a different classification. The algorithm is described by the authors as a gradient descent algorithm using an adaptive step size which it determines at each iteration. In simple terms, DeepFool uses Newton's iterative method to compute an approximation of the vectored minimum distance to the boundary of the complement of the input space partition recognized by the model as the correct class. It then perturbs the image by adding that vector.

2.2.6 Square Attack

The square attack is a score-based black-box L_2 and L_∞ adversarial attack that does not use local gradient information and thus is immune to gradient masking (Andriushchenko et al., 2020). It uses a randomized search scheme and perturbations are introduced such that they lie on the boundary of the L_2-hypersphere or L_∞-hypercube before their projection back inside the box constraint ($[0,1]^d$). First, a side-length for the square that will be perturbed is chosen, according to a decreasing schedule. Next, a δ is chosen, if, on applying the δ, the loss decreases, it is accepted. If not, it is rejected. If the new image classifies in a non-true class, the image is accepted. If not, the algorithm continues, repeating until either successful or the max number of iterations has been completed. The optimization problem this attack seeks a solution for is

$$\min_{\hat{x}\in[0,1]^d} \mathcal{L}(f(x'), y), s.t. ||x' - x||_p \leq \epsilon,$$

where x' is an adversarial example (i.e., $f(x') \neq y$) that is created from x, and y is the true label.

2.2.7 AutoAttack

AutoAttack is considered the state-of-the-art adversarial evasion attack. It is a highly effective ensemble of parameter-free attacks, combining the cross-entropy-based version of AutoPGD, difference-logits-ratio-based version of AutoPGD, SquareAttack and, finally, the Fast Adaptive Boundary Attack (FAB) (Croce & Hein, 2020a). These separate attacks are used in sequence and until $\text{argmax}(f_{x_i}) \neq y_{\text{true}}$. The adversarial robustness library (ART) (Nicolae et al., 2018), which we use in our testing, substitutes the DeepFool attack in for the FAB Attack. FAB is distinct in that it extrapolates the hyperplane more precisely than DeepFool. The authors explain, “it would be similar to DeepFool except that our projection operator is exact whereas they project onto the hyperplane and then clip to $[0,1]^d$.” FAB authors Croce et al, however point out that a weakness of untargeted FAB is its extensive computational cost as dataset complexity increases and the number of classes increases.

For ease of testing and standardization, the hyperparameters for each attack in AutoAttack are constant across models, datasets, and measurement norms. Given its extensive strength and effectiveness, untargeted AutoAttack is a standardized benchmark used by the adversarial robustness community to compare model defense robustness to adversarial evasion attacks in general (Croce et al., 2020).

2.3 Relevant Adversarial Defense Paradigms

Given the extensive literature that has accumulated regarding new methods to find effective, fast, and cheap evasion attacks, significant effort has likewise been expended pursuing effective measures to mitigate or eliminate these threats. Here we describe two DNN defense paradigms which, although near-complements of one another, have significantly different implications, strengths, and weaknesses. These include the well-known and ubiquitous *adversarial training* and the less well-exercised open-set approaches, like *NOTA-training* paradigms. The former's success at defending against existing state-of-the-art attacks is used as a baseline for evaluating the latter's robustness to original and adapted versions of the same attacks.

2.3.1 Adversarial Training Data Augmentation

The best known method for increasing DNN robustness remains adversarial training (Goodfellow et al., 2015; Szegedy et al., 2013). This is a data augmentation technique which adds adversarial examples to the true label class. The most common form uses PGD, which, in many cases, leads to significantly improved adversarial robustness (Madry et al., 2017).

2.3.2 NOTA Defenses

In contrast to adversarial training, NOTA defenses generate adversarial examples or other data points to populate the NOTA class with and use as training examples. The premise is that with carefully crafted NOTA augmentation examples, during training, one can continuously sow the data-poor regions of input space with NOTA, particularly using NOTA to separate data-rich regions of input space with differing classes. The DNN then learns to classify this space along with the adversarial examples which are perturbed into it, as NOTA.

Shafahi et al. showed mathematically using isoperimetric inequalities that if the partition of each class corresponds to a partition that takes up less than half the input space, then by their proofs, *every* example in *any* class will necessarily be within an extremely small L_p distance from an adversarial example (Shafahi et al., 2019). We conceive then, that one goal of a NOTA approach would be to coax the DNN model into designating more than half of the input space as NOTA. This then, should bring every example in any other class into a, according to Shafahi et al., calculably very close proximity of the NOTA-class partition (Shafahi et al., 2019). Though, the result cannot be said to rule out the possibility of adversarial examples from other classes, it, at a minimum, begins to crowd out the space of opportunity for other classes, especially in the presence of NOTA examples specifically chosen to surround the true label class partition tightly.

2.3.2.1 NOTA, Open-Set, and Out-of-Distribution Methods

Although NOTA defenses leverage the open-set concept, their goal differs from most open-set constructs in the literature. The "I don't know category" is not generally used as a defense, but rather as a method to identify novel classes of data not already defined. Shao et al. (2020) characterize a research problem they call Open-Set Adversarial Defense (OSAD), where adversarial attacks are studied under open-set settings. In their framing, the goal is to both identify open-set samples (representing new classes) and defend against adversarial evasion attacks. They demonstrate that open-set classifiers were readily fooled using existing closed set attack methods.

A related method is out-of-distribution defenses using thresholds, but this has been shown to be ineffective against simple adapted attacks. Enevoldsen et al. (2025) demonstrate that open-set recognition models that use thresholds of maximum softmax probability or maximum logit score to identify novel classes, are also easily deceived using simple adaptations to existing adversarial attacks to create false novelty or false familiarity results. Also, Grosse et al. (2018) show the ease with which adversarial attacks can achieve high confidence and low uncertainty adversarial examples which are misclassified by ML models, but not detected by an out-of-distribution threshold approach. Additionally, they demonstrate that such examples successfully transfer between different Bayesian models and approaches. Thus, their research implies that confidence and uncertainty alone cannot be used as a basis for defense against adversarial examples.

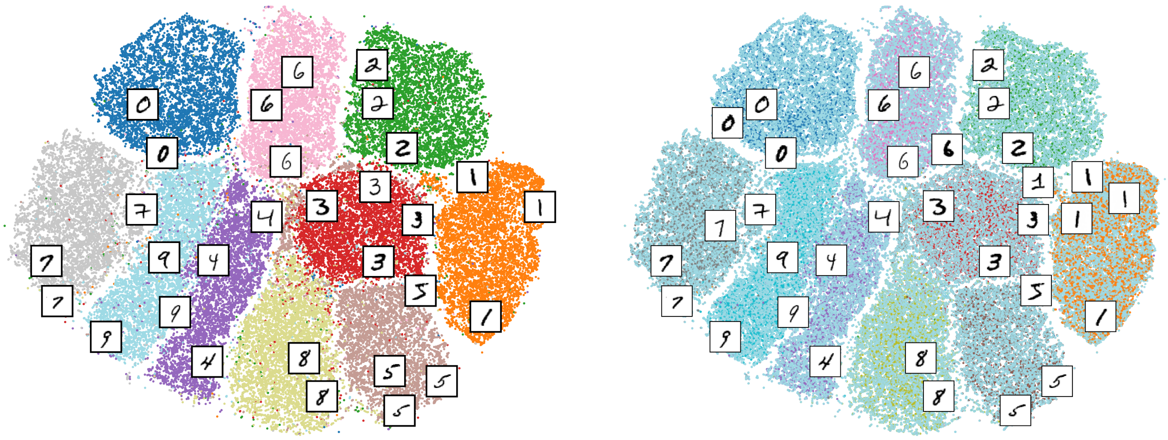
NOTA-type defenses are different in that they leverage the open-set concept to provide closed-set adversarial defense. NOTA defenses, therefore, do not facilitate or enable the identification of novel categories, nor do they use logit or uncertainty thresholds to identify adversarial examples. Rather, NOTA defenses leverage an additional none-of-the-above class to serve as the label for all adversarial examples, their derivatives, and open-set examples, relying on the DNN to generalize and identify adversarial examples as the NOTA class.

2.3.2.2 Boundary Padding

BP was conceived after preceding research showed that various methods of creating NOTA class examples—such as using linear interpolation in the input space, or mixing methods in latent lower dimensional space using auto-encoders—showed promise at defending DNNs against adversarial examples produced by the CW attack suite using confidences of 20 or higher (Barton, 2018; Barton et al., 2021). However these methods struggled to perform against low-confidence Carlini Wagner L_p attacks.

Another influence for BP resulted from Zhang et al. (2020), they introduce *Mixup*, an algorithm for instantiating linear behavior between training examples and increasing regularization and resistance to adversarial attack. Although BP discards the label-mixing aspect, it uses the simple mixing expression, $\lambda \cdot x_1 + (1 - \lambda) \cdot x_2$ for the image data as well as randomization of λ in a new way to create its NOTA class examples. BP then is an attempt to more closely surround correctly labeled regions of input space with NOTA class examples. However, instead of 'mixing' two separate training examples, as well as their labels, and training as that label combination, as is done in mixup, BP 'mixes' a single training set example x_1 with an adversarial example,

x'_1, derived using the PGD attack. The resulting BP image is labeled as the pure NOTA class and added to training on the fly. Note that *no mixing of the labels occurs*, all produced examples are instead labeled as the NOTA class and trained as such.



No Defense

Boundary Padding (NOTA examples in teal)

Figure 1: tSNE plots of MNIST digits data

In BP two variations of NOTA are produced on the fly and added to the training batch before batch-training commences, *mean BP* and *uniform BP*. In mean BP, λ is set to 0.5 and limited gaussian noise is added to the resulting image. In uniform BP, λ is set to a random number between 0.05 and 0.95 and a weighted average of the clean and adversarial example is performed. The resulting NOTA to clean data ratio is two to one, making it by far the largest class in the dataset. This large representation of NOTA vs any other class reflects the intuition that, regardless of the number of finite classes that are defined, the vast majority of possible inputs in the input space do not correspond to any of the specified classes and, thus, should instead be assigned to the NOTA class.

3 Methods

We study open-set enabled closed-set adversarial defenses that add a none-of-the-above (NOTA) class to a DNN classifier and show that their reported robustness is often overstated because standard attacks lack explicit handling for large auxiliary classes like NOTA. We present a three-part taxonomy of evaluation pitfalls and an attack-adaptation template we then apply across seven prominent benchmark attacks. Finally we introduce Adversarial NOTA Envelopment (ANE), a NOTA defense variant designed to eliminate an observed weakness in previous NOTA defenses, i.e. NOTA "clumping."

3.1 Taxonomy of Attack Failure Modes for NOTA Defenses

During our evaluations we found that attacking NOTA-defenses, both BP and ANE, with unadapted attacks often resulted in stark, overwhelming defense successes against a broad range of attacks. On close inspection of attack code however, we realized that most attacks shared common mechanisms which lacked necessary specification to handle an open-set defense incorporating a large all-encompassing class like NOTA.[2] We found that one or a combination of the following common mechanisms often resulted in a less-than-thorough search of the input space for a potentially successful adversarial example: the subject attack's *target selection*, its *stopping criteria*, and/or its *objective function*. By adapting the failing attack mechanisms according to our NOTA template, we expect attacks to get substantially closer to a more accurate assessment of the

[2] That is, NOTA is a class which is intended to encompass all adversarial examples from every other class as well as intermediate input space between classes and surrounding the data manifold.

Failure mode	**Mechanism (typical)**	**Effect in NOTA setting**	**Fix (attack-agnostic)**
F1. Target Selection	Choose target from any non-true class (or predicted label)	NOTA becomes an (easy) target or predicted "clean" label	Exclude y_{NOTA} from candidate targets and from any label inference.
F2. Stopping Criteria	Success declared when $\arg\max f(x') \neq y_{\text{true}}$	Early termination on NOTA predictions	Require $\arg\max f(x') \neq \{y_{\text{true}}, y_{NOTA}\}$; continue otherwise.
F3. Objective Function*	Loss encourages movement toward nearest non-true partition	Optimization steers into NOTA "honeypot"	Modify objective to avoid using NOTA logits as attractors, or explicitly repulse from y_{NOTA} (e.g., anti-NOTA term).

Table 1: Taxonomy of evaluation pitfalls for NOTA-style defenses and attack-agnostic fixes. *Note: Out of these three the objective function alone is optional. It is not always possible and should be implemented only if all else fails or proves insufficient.

adversarial robustness of these NOTA models as well as future models of the NOTA and/or open-set defense genre.

NOTA Attack-adaptation Template

For each baseline attack we apply, in order:

1. **Target Selection**: remove y_{NOTA} from any target set or possible initial label prediction;
2. **Stopping Criteria**: adopt NOTA-aware stopping criteria;
3. **Objective Function**: (*Optional*) If applicable and necessary, adjust losses that implicitly steer into NOTA (or add or replace with anti-NOTA repulsion term).

3.1.1 Target Selection

In many attacks, if a particular target class is not provided (i.e., a desired false class to perturb the image into), the attack chooses a 'best,' random, or otherwise determined class that is not the true label. If present, such a mechanism must be altered to additionally exclude the NOTA class as a potential target. This was discovered to be true in CW attacks (Carlini & Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), as well as targeted AutoPGD and targeted AutoAttack (Croce & Hein, 2020b). Closely related is the issue when true labels are not provided into an attack. The attack must use the model to predict what the initial true label is and perturb the image out of that class. If NOTA is the initial predicted true label, the attack may perturb the image into the true class. Again the solution is to exclude NOTA as a potential true label class as well.

3.1.2 Stopping Criteria

Stopping criteria are the predetermined circumstances under which attack code will presume success and cease exploring the input space by continuing to modify the adversarial example. It is no longer the case under the NOTA paradigm that just any class other than the true class will results in a successful evasion attack. In fact, the most abundant class, the one that is most likely to represent the closest decision boundary or next highest probability or logit, is likely to no longer correspond to a successful attack, as the goal of NOTA approaches is that such criteria will result instead in the NOTA class. For instance, some untargeted attacks set one condition for success to be to change the predicted class such that $\text{argmax}(f(x')) \neq y_{True}$.

Properly restated in a NOTA paradigm this should be changed to the following:

$$(\text{argmax}(f(x')) \neq y_{NOTA}) \wedge (\text{argmax}(f(x')) \neq y_{True})$$

This prevents attacks from immediately stopping when NOTA is the predicted class.

3.1.3 Objective Function

In some cases, an objective function which does not account for the presence or abundance of a NOTA class can result in driving perturbations directly into the NOTA class, rather than through it or away from it. As the non-true class with the closest data points to every training example, NOTA frequently has the steepest gradient away from the correct class and can act as a local minimum, or honeypot. This may require adjustment to the objective function. This is true in CW attacks and can be applied to other gradient-descent-based attacks, such as AutoPGD (Croce & Hein, 2020b).

3.2 Adversarial NOTA Envelopment

Given BP's existing loss formulations, as NOTA regions are planted and reinforced over the course of training epochs, it is likely that new NOTA examples will be planted in nearly the same input space locations epoch after epoch. Each new NOTA example then results in a steeper gradient to that portion of the input space, which in turn, will result in higher likelihood that future NOTA data augmentation will be created in close to the same place. This can have a reinforcing effect creating essentially a funnel or entrapment zone. As a result, NOTA regions clump near but not surrounding a class's partition space. Instead, the desired behavior of a NOTA defense should be to *surround* or envelop homogenously-classed datapoint-dense regions of input space with NOTA, while also populating data-sparse regions between them with NOTA.

Toward this end, the ANE defense retains all previous characteristics of BP, except it switches between two different losses in creating its NOTA data. One PGD loss maximizes the cross entropy (CE) with respect to the true label class as in BP. The second loss maximizes the cross entropy loss with respect to the NOTA class. This strategy is elegant in its simplicity, alternating between pushing away from the true label class to plant NOTA, and pushing away from existing NOTA partition to plant NOTA examples where it does not already exist.

$$\mathcal{L}(y_{\text{true}}, f(x)) = \begin{cases} \text{CE}(y_{\text{true}}, \text{softmax}(f(x))) & \text{when } \beta \leq 0.5 \\ \text{CE}(y_{NOTA}, \text{softmax}(f(x))) & \text{otherwise} \end{cases}$$

where $\beta \sim U(0, 1)$.

3.3 Applying the Taxonomy: per-attack NOTA adaptations

Below we summarize how each baseline attack triggers the taxonomy and the minimal changes we implement in the Adversarial Robustness Toolbox (Nicolae et al., 2018) implementations of each. Code is released with the paper. A summary table of all of these changes is provided in the appendix.

3.3.1 NOTA Adapted Carlini-Wagner L_∞ and L_2

For a description of Carlini and Wagner's attacks see Section 2.2.4 (Carlini & Wagner, 2017). Each L_p version of this attack requires an adjustment of the code it uses to select a target class (**F1**). When the target logit is selected, the NOTA class must be eliminated from consideration, since this logit represents the class that the attack perturbs the image into. Additionally, the attack's stopping criterion (**F2**) must be modified so that the attack is not registered as successful until the target logit is greater than the highest of either the logit of the true label class plus the input confidence score or the logit of the NOTA class plus the input confidence score.

3.3.2 NOTA Adapted Deep Fool

DeepFool is described in detail in Section 2.2.5 (Moosavi-Dezfooli et al., 2016). This attack required both an adjustment regarding code for target selection (**F1**)and an adjustment to the attack's stopping criteria (**F2**).

We first removed NOTA as an option when the algorithm ranks potential targets in its selection process. We also ensure that the attack will not stop in the case where NOTA is the predicted class after perturbation, unless it has reached the maximum iterations without finding a successful adversarial example.

3.3.3 NOTA Adapted Square Attack

In analyzing Square Attack (Andriushchenko et al., 2020), we first prevented the NOTA class from being chosen as the initially correct label (**F1**), in the event the true labels are not provided. Next, we evaluated the stopping criteria (**F2**). We altered the criteria so that the attack was only successful if it resulted in a classification other than the clean class *and the NOTA class*. We investigated changes to the loss function (**F3**) to incorporate a term that incentivized increased loss between the adversarial example and the NOTA class in addition to the clean class, but this resulted in less effective attacks against NOTA defenses. We believe this was due to the fact that the loss was used as a litmus test to determine if a change was an improvement or not and the two separate terms rarely reinforced and, more often, destructively interfered with one another resulting in no applied alterations. The strongest attack resulted from eliminating NOTA as a legitimate stopping class in the stopping criteria and eliminating it as a possible clean class.

3.3.4 NOTA Adapted Auto Projected Gradient Decent (APGD) CE, and DLR

Both the CE and DLR versions of the APGD attack (Croce & Hein, 2020b) are described in Section 2.2.3. In both versions of the APGD attack, it is necessary to ensure that if the labels are not provided, that the model does not predict the NOTA class as the clean example's class in creating the labels (**F1**). More importantly, the stopping criteria must be adjusted such that the attack will not stop when the NOTA class is predicted (**F2**), but instead keep iterating toward a successful attack. For the CE version of APGD it was worth investigating an adjustment to the loss function considering the introduction of a massive class like NOTA. We adjusted the loss to maximize a combination of the clean label class loss and the NOTA label class loss by taking their mean. We investigated this and other combinations to try and find a more effective implementation of the APGD-CE attack. These combination strategies were no more potent in the end, so we retained the original loss term for adapted NOTA APGD-CE. For APGD-DLR, we ensure that the NOTA class is not designated as any of the three logits that are used to calculate the difference logits ratio loss (**F3**).

3.3.5 NOTA Adapted APGD-AN

However, the process above did inspire the investigation of using the same NOTA-Aware APGD mechanism (**F1/F2**) with a new loss term, which exclusively maximizes the categorical cross-entropy between the DNN's prediction for an adversarial example and the NOTA label (**F3**). We call this loss term *Anti-NOTA* (AN). Our intuition is that, after training, the NOTA class examples have largely enveloped the entire manifold on which the dataset exists. Therefore, maximizing the cross-entropy loss with the NOTA label should push the input toward explicitly non-NOTA, off-manifold regions of input space, within the ϵ-bound, where there is a better chance of encountering an adversarial example that will not classify as NOTA.

3.3.6 NOTA Adapted AutoAttack

As AutoAttack (Croce & Hein, 2020a) leverages four separate parameter-free subordinate attacks, as described in Section 2.2.7, it is necessary to ensure that the adapted NOTA version of AutoAttack calls the *most effective* adapted versions of each of these subordinate attacks. Additional changes are required in the AutoAttack procedure which wraps the subordinate attacks. As in other attacks, the stopping criteria (**F2**) evaluated after each subordinate attack is completed must be adjusted to exclude the prediction of NOTA as a successful outcome for the attack. We test both a NOTA-aware version of AutoAttack with NOTA-Aware APGD-CE and a NOTA-aware version of AutoAttack with NOTA-aware APGD-AN, substituted into APGD-CE's place.

4 Experiments and Results

In this section we evaluate the effectiveness of our NOTA-adapted attacks against NOTA-defended models trained separately on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). Specifically, we compare the resulting effectiveness of these attacks on NOTA-defended models to the effectiveness of the original attacks against NOTA-defended models, adversarial training-defended models and fully undefended models. We first evaluate all defenses and models against seven prominent highly effective unmodified attacks. We then implement NOTA adaptations on all seven attacks as identified and discussed in Section 3.3, evaluating the effectiveness of these attacks at mitigating or eliminating NOTA defenses and reporting all results in tables 3 and 4.

4.1 Experimental Setup

We use standard Wide Residual Networks (WRN) with dropout and batch normalization, configured as suggested by Zagoruyko & Komodakis (2016) as the base model architecture for our experiments. Dropout is set to 30% drop probability during training. For all models, we use a wide resnet, 12 units deep and 6 units wide, i.e., WRN-12-6, a common setup for these datasets. For each model we use an $ADAM$ (adaptive moment estimation) optimizer with default settings, $b_1 = 0.9$ and $b_2 = 0.999$ (Kingma & Ba, 2017) with sharpness-aware minimization (SAM) (Foret et al., 2021). ADAM assists the model in efficiently converging on a solution, whereas SAM, by seeking out minima of the training loss landscape and minimizing loss curvature, smooths the boundaries between partitions expressed in input space. When performing mini-batch gradient descent training, two separate batches of 32 are drawn from the training set for each training cycle. The first uses standard dataset augmentation, a random up to 10% shift up or down, and left or right, as well as random horizontal flipping and a random, up to a 15 degree rotation clockwise or counter clockwise. The second batch is a 'clean' batch drawn from a separate iterator without data augmentation. Each batch is used both as a benign training batch and also to create NOTA examples based on each training example.

We perform experiments using the CIFAR-10 and CIFAR-100 datasets. Our datasets are split before training such that 4% of the former training set are quarantined as a validation set to enable early-stopping model selection, based on a combination of best validation accuracy and best validation adversarial robustness. In very close models, we slightly favor best validation accuracy over validation adversarial robustness. Every 150 batches during training, the same 30 examples are used (previously separated from the validation set) to create 30 untargeted, zero-confidence CWL_2 adversarial examples with a maximum of 10 iterations. These adversarial examples are then used to calculate a validation 'attack success rate' (ASR), which is used in model selection. ASR is calculated by determining the number of adversarial examples that successfully drove the model to classify the image as a class *other* than the NOTA class or the true label. The number of successful adversarial examples divided by the total number of attempted adversarial examples results in the ASR. The process detailed here is precisely the same for all model training and model selection, whether NOTA, adversarial training, or undefended models.

Table 2: Clean Model Test Accuracies

CIFAR-10	
Model	**Accuracy**
No Defense	92.31%
Adversarial Training	90.71%
Boundary Padding	90.93%
Adversarial NOTA Envelopment	92.20%
CIFAR-100	
No Defense	70.53%
Adversarial Training	66.47%
Boundary Padding	68.13%
Adversarial NOTA Envelopment	69.89%

All models were WRN-12-6, with dropout of 30% and batch normalization.

The test set is strictly reserved for testing a specific model that has been previously selected using only ASR and accuracy performance from the validation set. In testing, accuracy is calculated from the full test set. ASR is calculated on the preselected model using adversarial examples created using sufficient test set samples to ensure reasonably small confidence intervals.

We state our findings along with their 95% binomial confidence intervals. We test against Carlini Wagner L_2, L_∞ (Carlini & Wagner, 2017), AutoPGD-CE, AutoPGD-DLR (Croce & Hein, 2020b), DeepFool (Moosavi-Dezfooli et al., 2016), Square Attack (Andriushchenko et al., 2020), and AutoAttack (Croce & Hein, 2020a). We adapt each of these attacks to counter the NOTA defense, and add an additional variant of AutoPGD, AutoPGD-AN, all as specifically described in sections 3.3.1—3.3.6. In parameterized attacks, we set max iterations to 100, i.e., CW L_p attacks, square attack and DeepFool (L_2). All AutoPGD attacks, Square Attack, and AutoAttack are performed in both L_2 with max epsilon of 0.5 (maximum distance between x and x' by specified L_p metric) and L_∞ with max epsilon of 8/255, the distances specified for each by Robust Bench (Croce et al., 2020). Finally, DeepFool is tested in default L_2 with the standard max $\epsilon = 0.5$.

4.2 Results

In tables 3 and 4 attack success rates (ASR) are reported for original attacks against both undefended and defended models (adversarial training models, boundary padding models, and adversarial NOTA envelopment models, respectively).

Table 3: Attack Success Rates (ASRs) for Original and Adaptive NOTA Attacks for CIFAR-10

Models	**C&W Suite**		**APGD CE**		**APGD AN**		**APGD DLR**		**Square Attk**		**DF**	**AA, Untrgtd**		**AA-AN, Untrgtd**	
	L_2, Conf:0	L_∞, Conf:0	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031
No Defense vs. Orig. Attacks	99.0%, ±2.0%	100.0%, ±0.0%	99.5%, ±1.0%	100.0%, ±0.0%	—	—	85.5%, ±4.9%	95.0%, ±4.3%	12.0%, ±6.4%	56.0%, ±9.7%	98.0%, ±2.7%	100.0%, ±0.0%	100.0%, ±0.0%	—	—
Adv. Train vs. Orig. Attacks	94.6%, ±2.0%	100%, ±0.0%	53.2%, ±4.4%	96.6%, ±1.6%	—	—	49.4%, ±4.4%	87.8%, ±2.9%	12.4%, ±2.9%	35.6%, ±4.2%	92.4%, ±2.3%	53.4%, ±4.4%	96.8%, ±1.5%	—	—
BP vs. Orig. Attacks	16.0%, ±7.2%	1.0%, ±2.0%	6.0%, ±4.7%	6.0%, ±4.7%	—	—	42.0%, ±9.7%	55.0%, ±9.8%	14.0%, ±6.8%	64.0%, ±9.4%	9.0%, ±5.6%	6.0%, ±4.7%	6.0%, ±4.7%	—	—
ANE vs. Orig. Attacks	12.0%, ±6.4%	0.0%, ±0.0%	5.0%, ±4.3%	5.0%, ±4.3%	—	—	15.0%, ±7.0%	39.0%, ±9.6%	5.0%, ±4.3%	40.0%, ±9.6%	5.0%, ±4.3%	5.0%, ±4.3%	5.0%, ±4.3%	—	—
BP vs. NOTA Attacks	**99.0%,** **±2.0%**	1.0%, ±2.0%	9.0%, ±2.5%	9.2%, ±2.5%	**18.8%,** **±3.4%**	**91.8%,** **±2.4%**	**91.0%,** **±5.6%**	**100.0%,** **±0.0%**	15.0%, ±7.0%	63.0%, ±9.5%	13.0%, ±6.6%	**93.4%,** **±2.2%**	**99.6%,** **±0.6%**	**92.8%,** **±2.3%**	**99.6%,** **±0.6%**
ANE vs. NOTA Attacks	9.0%, ±5.6%	5.0%, ±4.3%	8.6%, ±2.5%	9.4%, ±2.6%	**46.8%,** **±4.4%**	**74.4%,** **±3.8%**	**55.0%,** **±9.8%**	**91.0%,** **±5.6%**	5.0%, ±4.3%	37.0%, ±9.5%	9.0%, ±5.6%	**52.6%,** **±3.4%**	**95.4%,** **±1.8%**	**54.0%,** **±4.4%**	**95.0%,** **±1.9%**

The blanks represent original attacks, for which ANTI-NOTA (AN) loss does not exist as it is novel to this paper and its attacks.

Table 4: Attack Success Rates (ASRs) for Original and Adaptive NOTA Attacks for CIFAR-100

Models	**C&W Suite**		**APGD CE**		**APGD AN**		**APGD DLR**		**Square Attk**		**DF**	**AA, Untrgtd**		**AA-AN, Untrgtd**	
	L_2, Conf:0	L_∞, Conf:0	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_2, ϵ:0.5	L_∞, ϵ:0.031	L_2, ϵ:0.5	L_∞, ϵ:0.031
No Defense	98.3%, ±1.0%	100.0%, ±0.0%	98.8%, ±0.3%	100.0%, ±0.0%	—	—	99.7%, ±0.5%	100.0%, ±0.0%	41.3%, ±3.9%	87.3%, ±2.7%	83.5%, ±3.0%	99.8%, ±0.3%	100%, ±0.0%	—	—
Adv. Train vs. Orig. Attacks	95.6%, ±1.8%	100%, ±0.0%	78.6%, ±3.6%	97.2%, ±1.5%	—	—	77.0%, ±3.7%	96.0%, ±1.7%	32.4%, ±4.1%	62.2%, ±4.3%	89.0%, ±2.7%	78.8%, ±3.6%	97.2%, ±1.5%	—	—
BP vs. Orig. Attacks	51.8%, ±4.4%	2.2%, ±1.3%	33.2%, ±4.1%	33.2%, ±4.1%	—	—	78.4%, ±3.6%	69.4%, ±4.0%	47.6%, ±4.4%	36.8%, ±4.2%	24.2%, ±3.8%	33.2%, ±4.1%	33.2%, ±4.1%	—	—
ANE vs. Orig. Attacks	40.2% ±4.3%	0% ±0.0%	31.2% ±4.1%	31.2% ±4.1%	—	—	51.8% ±4.4%	54.6% ±4.4%	31.2% ±4.1%	66.0% ±4.2%	16% ±3.2%	31.2% ±4.1%	31.2% ±4.1%	—	—
BP vs. NOTA Attacks	54.4%, ±4.4%	**34.6%,** **±4.2%**	33.2%, ±4.1%	33.2%, ±4.1%	**58.2%,** ±4.3%	**98%,** ±1.2%	**99.6%,** **±0.6%**	**99.4%,** **±0.7%**	48.2%, ±4.4%	**49.2%,** **±4.4%**	**32.8%,** **±4.1%**	**97.8%,** **±1.3%**	**98.8%,** **±1.0%**	**98.4%,** ±1.1%	**99.4%,** ±0.7%
ANE vs. NOTA Attacks	**49.4%,** **±4.4%**	**30.3%,** **±3.7%**	31.2%, ±4.1%	31.2%, ±4.1%	**83.4%,** **±2.6%**	**96.6%,** **±1.6%**	**89.8%,** **±2.7%**	**90.4%,** **±2.6%**	31.3%, ±3.7%	68.0%, ±4.1%	20.6%, ±3.0%	**89.2%,** **±2.7%**	**96.6%,** **±1.6%**	**94.6%,** **±2.0%**	**98.6%,** **±1.0%**

The blanks represent original attacks, for which ANTI-NOTA (AN) loss does not exist as it is novel to this paper and its attacks.

4.2.1 Unmodified Attacks vs. Defenses

As recorded in the middle two rows of Tables 3 and 4, BP and ANE result in a stunning and ostensibly noteworthy performance in defending against a broad cross-section of the unmodified attacks. Unmodified AutoAttack results in an astoundingly low L_2 and L_∞ ASR of 6% and 6% for the CIFAR-10 dataset. This is in comparison to 100% and 100% against the undefended model, and 53.4% and 96.8% against the adversarial training defended model. Likewise, in Table 4, unmodified AutoAttack is only able to manage an L_2 and L_∞ ASR of 33.2% and 33.2% respectively against the BP defense, each of which is within the confidence interval of the model's natural error rate. Contrast this with the undefended model's 99.8% and 100% ASRs, and the adversarial training defended model's 78.8% and 97.2% ASRs against the same attack. With respect to other mainstream attacks, BP and ANE record substantial reductions to ASRs for nearly all attacks across both data sets, with ANE consistently out-defending or statistically matching adversarial training with respect to each attack. For the L_2-bounded Square Attack in the CIFAR-10 dataset, there is no significant statistical difference between the adversarial-trained defense, and the best NOTA defended model, ANE. The unmodified L_∞-bounded Square Attack ASR is likewise a statistical tie between adversarial training and ANE defenses. Overall, considering the substantially successful defense against these benchmark attacks, NOTA defenses would seem an impressive advance in favor of increased adversarial robustness in classification systems.

Further analysis of the resilience of these defenses to the unmodified attacks reveals some curious observations, however. First, Square Attack remains largely effective against naive BP and, yet, AutoAttack, which leverages square attack as one of its components, does not register a commensurate ASR, or even above the model's natural error rate in either data set. One key to understanding how this can occur is that NOTA defense approaches, in general, rest on a fundamental change to the existing training paradigm, i.e., they add a NOTA class to the model and assert that inputs which are predicted in that class are to be considered adversarial examples. Thus, simply driving an input into a class other than its true label is no longer a sufficient strategy for an attack, it must also exclude the NOTA label as a successful result. This explains AutoAttack's lower ASR, even though Square Attack is one of its component attacks, those adversarial examples that are deemed successful with earlier component attacks are not altered further, so an example that results in a NOTA class with an earlier attack will not be further iterated with a potentially more successful later subordinate attack. This and other similar insights described in section 3.1 provided ample opportunity to adapt each attack to maximize its ASR.

4.2.2 NOTA-Aware Adaptive Attacks vs. Defenses

As is evident in Tables 3 and 4, adapting the selected adversarial evasion attacks to the NOTA paradigm results in substantial increases in attack success rate for most attacks against both NOTA-defended models. However, some exceptions do arise, such as APGD-CE, CW L_∞, and DeepFool, where, significant robustness is retained in the NOTA defenses, showing either non-statistically different results with the original attack or only a slightly higher ASR. In the case of APGD-CE, the same code (with different loss, however) is used in executing APGD-DLR and our newly introduced APGD-AN, both of which show strong increases in ASR against NOTA defenses in each dataset. This, therefore, validates the shared mechanism in the three attacks and reveals that the defenses do have a particular strength against *standard* gradient-based *cross-entropy* attacks. This finding is no surprise as both NOTA defenses tested use variations on gradient-based cross-entropy to create NOTA training examples. Overall, the fact that a majority of these attacks substantially increase ASR, at times achieving 99% ASR, and that no attack performs statistically poorer with the implemented changes to stopping criteria, target selection, or objective function, validates the attack adaptations against both NOTA defenses.

4.2.2.1 Adapted Attacks vs. BP

Observing BP's defense of CIFAR-10, NOTA-Aware APGD-DLR increased the ASR by 49% in L_2 to 91% and by 45% in the L_∞ norm to 100%. NOTA-Aware AutoAttack, likewise, increased the ASR, by 94% in both L_2 and L_∞. NOTA-Aware CW L_2 increased ASR by 84%. The lack of substantial increase in CW L_∞ and APGD-CE ASR can be attributed to the fact that the NOTA class is created from modified cross-entropy-based PGD. BP's defense of CIFAR-100 showed very similar results with some variation. NOTA-Aware CW L_2 is a statistical tie with the original attack at just above 50% ASR. NOTA-Aware CW L_∞ does however show an improvement over the original attack of greater than 30%. NOTA-Aware APGD-DLR is restored to above 99% ASR for both L_2 and L_∞, an increase of 21% and 30% ASR respectively. NOTA-Aware Square attack resulted again with statistically insignificant results compared with the original attack. DeepFool saw a greater than 8% increase in ASR, but still no greater than natural error. NOTA-Aware AutoAttack saw an increase greater than 64% in both norms, to about 98% ASR.

4.2.2.2 Adapted Attacks vs. ANE

ANE, though initially promising and more effective against unadapted attacks, collapses once the evaluation pitfalls are fixed. With respect to the CIFAR-10 dataset, although ANE does recover some general robustness when compared to BP in defending against NOTA-aware attacks, overall, the majority of the attacks are still substantially or even fully potent. NOTA-aware AutoPGD-DLR results in an ASR of 55% in the L_2 and 91% in the L_∞ bounds. Likewise, untargeted NOTA-Aware AutoAttack achieves 53% L_2 and 95% L_∞ ASR. NOTA-Aware Carlini and Wagner L_2 was not as successful against ANE as it was against BP, with its ASR reduced from 99% to just 9%, or roughly natural error. NOTA-Aware C&W L_∞ remained ineffective, likely for the same reasons described in section 4.2.2.1. NOTA-Aware Square attack and Deep Fool again remain roughly statistically equivalent to the original attack versions against ANE, indicating that whatever robustness is conferred by the defense against these attacks is not due to the failure modes for stopping criteria and target selection that we identify and correct in their code. However, as revealed by NOTA-Aware versions of APGD-DLR, APGD-AN, Square Attack, and Auto Attack (AA and AA-AN), adversarial examples for CIFAR-10 nevertheless exist in abundance and are readily discovered by these adapted attacks.

ANE reveals much the same story in its performance defending against NOTA-Aware attacks for the CIFAR-100 dataset.[3] One notable difference in CIFAR-100 is the improved performance of NOTA-Aware C&W L_2, which in this dataset achieved 49.4% ASR against ANE, approximately 20% above natural error in the model. Again, APGD-CE ASR was only commensurate with natural error, however, APGD-AN and APGD-DLR, which use the same code only leveraging different losses, each substantially defeat the defense. APGD-AN results in ASRs of 83% in L_2 and 97% in L_∞, whereas APGD-DLR achieves 90% in L_2 and 90% in L_∞. The Anti-NOTA APGD-AN variation showed that it alone among the adapted NOTA-Aware attacks was more successful against ANE than BP (only in the L_2 bound), with BP showing a still significantly compromised ASR of 58%, but against ANE an ASR of 83%. NOTA-Aware Square Attack and Deep Fool show little or no statistically significant difference with their original versions against CIFAR-100. Finally, both AutoAttack with APGD-CE and AutoAttack with APGD-AN are extremely effective with ASRs near or above 90% in both bounds.

4.2.3 Comparing Adversarial Training with Adversarial NOTA Envelopment

With the advent of effective adapted attacks against the best NOTA defenses, a far more balanced comparison can be made between the standard for adversarial defense, adversarial training, and this complement of its defense paradigm, NOTA. First and foremost, our testing confirms that neither is a solution to the problem of adversarial attacks alone. Nevertheless, the comparison apparent in tables 3 and 4 are still of interest. Overall, there are some mixed performances reported here, with most, though not all, resulting in adversarial training (AT) providing less robustness against standard attacks than ANE provides against NOTA-Aware adapted versions of those attacks.

Looking first at CIFAR-10 results, the adversarial trained (AT) model is less robust when compared to ANE in C&W L_2 (AT: 95%, ANE: 9%), C&W L_∞ (AT: 100%, ANE: 5%), APGD-CE L_2 (AT: 53%, ANE: 9%), APGD-CE L_∞ (AT:97%, ANE: 9%), and DeepFool (AT: 92%, ANE: 9%). AT and ANE have statistically nonsignificant differences in results for attacks APGD-DLR L_2 (AT: 49%, ANE: 55%), APGD-DLR L_∞ (AT: 88%, ANE: 91%), SquareAttack L_2 (AT: 12%, ANE: 5%), SquareAttack L_∞ (AT: 36%, ANE: 37%), AutoAttack L_2 (AT: 53%, ANE: 53%), and AutoAttack L_∞ (AT: 97%, ANE: 95%). One would need to compare ANE's results for NOTA-Aware APGD-AN to AT's results for APGD-CE for a fair comparison of the anti-NOTA loss version of APGD, in which case ANE shows modest improvement (where significant) over AT, L_2 (AT: 53%, ANE: 47%), and L_∞ (AT: 97%, ANE: 74%).

Turning to CIFAR-100 for a comparison of robustness between adversarial training and adversarial NOTA envelopment, we have very similar results. AT is less robust when compared to ANE in C&W L_2 (AT: 96%, ANE: 49%), C&W L_∞ (AT: 100%, ANE: 30.3%), APGD-CE L_2 (AT: 79%, ANE: 31%), APGD-CE L_∞ (AT:97%, ANE: 31%), APGD-DLR L_∞ (AT: 96%, ANE: 90%), and DeepFool (AT: 89%, ANE: 21%). However, AT shows greater robustness compared to ANE in APGD-DLR L_2 (AT: 77%, ANE: 90%) and

[3] The WRN-12-6 C100 models for each of undefended, Adversarial Training, BP, and ANE, all have natural error rates that hover at 30% to 33%, see Table 2 for specific model accuracies (natural error = 1 - accuracy).

AutoAttack L_2 (AT: 79%, ANE: 89%). There are no statistically significant differences noted for SquareAttack L_2 and L_∞, or AutoAttack L_∞. Again, one needs to compare ANE's results for NOTA-Aware APGD-AN to AT's results for APGD-CE for a fair comparison of the anti-NOTA loss version of APGD, in which case for CIFAR-100, there is no statistically significant difference in performance between the two, L_2 (AT: 79%, ANE: 83%), and L_∞ (AT: 97%, ANE: 97%).

All results considered, ANE would appear to confer greater and more general robustness to a model than adversarial training, however, considering the results from the adapted NOTA-Aware attacks, both collapse and fail to defend models from evasion attacks, with many attack options available to get 90% or greater attack success rates.

5 Conclusion

In this paper we begin by discussing and evaluating a group of open-set adversarial defense approaches which employ a none-of-the-above class to defend against evasion attacks on deep neural networks. In investigating why this genre of defense, on its surface, is effective against many attacks, we discover and provide a simple taxonomy for several common attack failure modes. Finally, we modify seven prominent and highly effective benchmark attacks, eliminating the identified failure modes and allowing us to largely recover attack potency against NOTA defenses. We then evaluate the effectiveness of the adapted attacks against NOTA defenses and compare the results to the effectiveness of standard attacks against adversarial training.

We observe that although our adapted attacks clearly show that present NOTA defenses are not sufficient to defend against attacks adapted as outlined in this paper, NOTA defenses in general do appear to confer some small residual resilience to even these adapted attacks that at least rivals adversarial training. The attack adaptations identified in this paper can be applied to any existing or future attack to increase its effectiveness against open-set approaches to adversarial defense.

With several adapted attacks recovering ASRs back to 90% and above in both datasets, we advise that future evaluations of NOTA-type or open-set enabled defenses must begin by testing with NOTA-aware attacks. To this end we make the NOTA-Aware adaptations to attacks created in this paper available as a library for public use. Finally, if a NOTA-aware version of an attack is not available, the practitioner will find the taxonomy in this paper instructive in how to adapt any new attack to eliminate the three identified pitfalls.

References

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv preprint arXiv:1802.00420*, 2018.

A. Barton, E. Jatho, and V. Berzins. Defending Against Adversarial Examples in Deep Neural Network Classifiers. Technical Report NPS-CS-21-002, Naval Postgraduate School, Monterey, CA 93941, 2021. URL `https://calhoun.nps.edu/handle/10945/68624`.

Armon Barton. *Defending Neural Networks Against Adversarial Examples.* PhD thesis, 2018. URL `https://rc.library.uta.edu/uta-ir/handle/10106/27743`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

Ewen Callaway. 'it will change everything': Deepmind's ai makes gigantic leap in solving protein structures. *Nature*, 588:203–204, 12 2020. doi: 10.1038/d41586-020-03348-4.

Nicholas Carlini. (yet another) broken adversarial example defense at ieee sp 2024. Nicholas Carlini, May 2024. URL `https://nicholas.carlini.com/writing/2024/yet-another-broken-defense.html`.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On Evaluating Adversarial Robustness, February 2019. URL `http://arxiv.org/abs/1902.06705`. arXiv:1902.06705 [cs, stat].

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018.

Joana C Costa, Tiago Roxo, Hugo Proença, and Pedro RM Inácio. How deep learning sees the world: A survey on adversarial attacks & defenses. *IEEE Access*, 2024.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020b.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Philip Enevoldsen, Christian Gundersen, Nico Lang, Serge Belongie, and Christian Igel. Familiarity-based open-set recognition under adversarial attacks. In Tetiana Lutchyn, Adín Ramírez Rivera, and Benjamin Ricaud (eds.), *Proceedings of the 6th Northern Lights Deep Learning Conference (NLDL)*, volume 265 of *Proceedings of Machine Learning Research*, pp. 58–65. PMLR, 07–09 Jan 2025. URL `https://proceedings.mlr.press/v265/enevoldsen25a.html`.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2018. URL `https://arxiv.org/abs/1707.08945`.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization, April 2021. URL `http://arxiv.org/abs/2010.01412`. arXiv:2010.01412 [cs, stat].

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR'15)*, 2015.

Kathrin Grosse, David Pfaff, Michael T. Smith, and Michael Backes. The limitations of model uncertainty in adversarial settings. *CoRR*, abs/1812.02606, 2018. URL `http://arxiv.org/abs/1812.02606`.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL `http://arxiv.org/abs/1412.6980`. arXiv:1412.6980 [cs].

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

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

Patrick McClure, Nao Rho, John A. Lee, Jakub R. Kaczmarzyk, Charles Y. Zheng, Satrajit S. Ghosh, Dylan M. Nielson, Adam G. Thomas, Peter Bandettini, and Francisco Pereira. Knowing what you know in brain segmentation using bayesian deep neural networks. *Frontiers in Neuroinformatics*, 13, 2019. ISSN 1662-5196. doi: 10.3389/fninf.2019.00067. URL `https://www.frontiersin.org/articles/10.3389/fninf.2019.00067`.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Maria-Irina Nicolae, Mathieu Sinn, Tran Ngoc Minh, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Ian M. Molloy, and Benjamin Edwards. Adversarial robustness toolbox v0.2.2. *CoRR*, abs/1807.01069, 2018. URL `http://arxiv.org/abs/1807.01069`.

Neil Savage. How AI is improving cancer diagnostics. *Nature*, 579:S14–S16, 03 2020. doi: 10.1038/d41586-020-00847-2.

Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=r1lWUoA9FQ`.

Rui Shao, Pramuditha Perera, Pong C. Yuen, and Vishal M. Patel. Open-set adversarial defense. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII*, pp. 682–698, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58519-8. doi: 10.1007/978-3-030-58520-4_40. URL `https://doi.org/10.1007/978-3-030-58520-4_40`.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pp. 1528–1540, 2016.

Fnu Suya, Anshuman Suri, Tingwei Zhang, Jingtao Hong, Yuan Tian, and David Evans. Sok: Pitfalls in evaluating black-box attacks. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 387–407. IEEE Computer Society, 2024.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Hamed Valizadegan, Miguel J. S. Martinho, Laurent S. Wilkens, Jon M. Jenkins, Jeffrey C. Smith, Douglas A. Caldwell, Joseph D. Twicken, Pedro C. L. Gerum, Nikash Walia, Kaylie Hausknecht, Noa Y. Lubin, Stephen T. Bryson, and Nikunj C. Oza. ExoMiner: A highly accurate and explainable deep learning classifier that validates 301 new exoplanets. *The Astrophysical Journal*, 926(2):120, feb 2022. doi: 10.3847/1538-4357/ac4399. URL `https://doi.org/10.3847%2F1538-4357%2Fac4399`.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020. Publisher: ACM New York, NY, USA.

Algorithm 1: DeepFool: multi-class case (Moosavi-Dezfooli et al., 2016)

```
input: Image $x$, classifier $f$ ;
output: Perturbation $\hat{r}$ ;
$y_{\text{pred}}(x) = \text{argmax}(f(x))$ ;
Initialize $x_0 \leftarrow x, i \leftarrow 0$;
while $y_{pred}(x_i) = y_{pred}(x_0)$ do
    for $y \neq y_{pred}(x_0)$ do
        $w'_y \leftarrow \nabla f_y(x_i) - \nabla f_{y_{\text{pred}}(x_0)}(x_i)$;
        $f'_y \leftarrow f_y(x_i) - f_{y_{\text{pred}}(x_0)}(x_i)$;
    $\hat{l} \leftarrow \text{argmin}_{y \neq y_{\text{pred}}(x_0)} \frac{|f'_y|}{||w'_y||_2}$;
    $r_i \leftarrow \frac{|f'_{\hat{l}}|}{||w'_{\hat{l}}||_2^2} w'_{\hat{l}}$;
    $x_{i+1} \leftarrow x_i + r_i$;
    $i \leftarrow i + 1$;
return: $\hat{r} = \sum_i r_i$
```

Algorithm 2: Square Attack via random search (Andriushchenko et al., 2020)

```
input: classifier $f$, point $x \in \mathbb{R}^d$, image size $w$, number of color channels $c$, $l_p$-radius $\epsilon$, label
    $y \in [1, ..., K]$, number of iterations $N$ ;
output: approximate minimizer $x' \in \mathbb{R}^d$ of the problem stated in the equation under Square Attack ;
$x' \leftarrow init(x)$, $l^* \leftarrow \mathcal{L}(f(x), y)$, $i \leftarrow 1$;
while $i \leq N$ and $x'$ is not adversarial do
    $h^{(i)} \leftarrow$ side length of the square to modify (according to a schedule);
    $\delta$ $P(\epsilon, h^{(i)}, w, c, x', x)$ (see paper for sampling distributions.);
    $x'_{new} \leftarrow$ Project $x' + \delta$ onto $\{z \in \mathbb{R}^d : ||z - x||_p \leq \epsilon\} \cap [0, 1]^d$;
    $l_{new} \leftarrow \mathcal{L}(f(x'_{new}), y)$;
    if $l_{new} < l^*$ then
        $x' \leftarrow x'_{new}$;
        $l^* \leftarrow l_{new}$;
    $i \leftarrow i + 1$
return: $x'$
```

A Amplifying Algorithms for some attacks

A.1 DeepFool

A.2 Square Attack

B Summary of Adapted Attack Changes

Summary of Attack Changes

- NOTA Adapted Carlini-Wagner L_2 and L_∞
  - Triggers: **F1**, **F2**
  - Changes: (i) exclude y_{NOTA} from target selection; (ii) no NOTA termination, decision rule compares target logit against max of $\{z_{y_{true}}, z_{y_{NOTA}}\}$ (plus confidence margin), preventing perturbations that beat the true class but still sit below NOTA.
- NOTA Adapted Deep Fool
  - Triggers: **F1**, **F2**.
  - Changes: (i) exclude y_{NOTA} from the potential classes used to form the closest hyperplane; (ii) no NOTA termination.
- NOTA Adapted Square Attack
  - Triggers: **F1**, **F2**.
  - Changes: (i) exclude y_{NOTA} from possible clean label prediction; (ii) no NOTA termination.
  - Note: adding an explicit NOTA-repulsion term to the score criterion hurt attack success rates.
- AutoPGD-CE
  - Triggers: **F1**, **F2**.
  - Changes: (i) exclude y_{NOTA} from possible clean label prediction; (ii) no NOTA termination.
- AutoPGD-DLR
  - Triggers: **F1**, **F2**, **F3**.
  - Changes: as in CE for F1 and F2, additionally exclude y_{NOTA} from the three logits that define DLR.
- AutoPGD-AN (ours)
  - Triggers: **F1**, **F2**, **F3**.
  - Changes: as in CE for F1 and F2 (same attack call mechanisms) but change of loss to Anti-NOTA.
  - Loss: maximize CE(y_{NOTA}, $softmax(f(x))$), which minimizes $p(y_{NOTA}|x')$ and drives the perturbation away from NOTA regions within the ϵ-bound.
- AutoAttack (wrapper)
  - Triggers: **F2**, **F1/F3** (via subattacks)
  - Changes: (i) no NOTA termination after each subattack's completion; (ii) replace APGD-CE with either NOTA-aware APGD-CE or APGD-AN; (iii) substitute NOTA-aware subattacks throughout.

Future in Open-Set Enabled Closed-Set Defenses?

The authors note that our findings beg the question, "...what then can really be gained in open-set-enabled closed-set evasion defenses like NOTA that we don't already have elsewhere?" We speculate that the complement-set structure afforded in NOTA approaches can still offer an opportunity in future research. The ideas motivating these defenses did not depend on tricking attacks by circumventing their stopping criteria, target selection, or objective functions, although they have to this point benefited from them, as this work makes clear. Rather, the complement set offers an opportunity to leverage the DNN's strengths in generalization to create a buffer between all classes. Considering the mathematical arguments of Shafahi et al. (2019), and the observed persistent resistance to APGD-CE and C&W L_∞, there are good reasons to think this structure holds promise.

Some recommendations for those pursuing these defense ends would be to investigate how to bias models toward the NOTA class so that data-sparse regions of input space default to NOTA classification. In deterministic models, this could be achieved by increasing the ratio of NOTA to clean samples or perhaps logit manipulation. Alternatively (or in combination), one could look to bayesian neural networks to leverage an ensemble of related DNNs which would presumably agree on partitions of datadense input space but differ on datasparse regions.