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 target-selection, stopping criteria, and objective function behaviors that mask true vulnerabilities. We formalize these issues in a taxonomy of evaluation pitfalls, adapt eight 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 (NLP) (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 an 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.

Our analysis reveals that progress on these 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 adapts attacks to a given defense. Without a standardized, defense-adapted attack suite, adversarial machine learning 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. Adapted attacks have been proposed for many defenses, but attacks adapted to defeat NOTA defenses have not been well-studied. By providing a taxonomy of attack failures against NOTA defenses and a drop-in set of NOTA-adapted attack variants, our work aims to raise the default evaluation bar for this genre of defenses. This will enable better understanding of the robustness provided by current and future NOTA defenses by giving practitioners a higher-confidence starting point and allowing the community to focus on genuinely new vulnerabilities rather than repeating past mistakes. Specifically, this paper has the following main contributions:

- Taxonomy of three recurring evaluation pitfalls for NOTA or open-set approaches to adversarial defense.
- Evaluation attack suite of eight NOTA-aware attack variants (code released).
- Case Studies, from both the computer vision and natural language processing domains, demonstrating that existing defenses–including two novel open-set NOTA defenses–fail under proper evaluation.

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 $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}^c$. Where the input is $x \in \mathbb{R}^d$, and the output is in $\mathbb{R}^c$, 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 classes. In the text classification domain, an input text is a sequence of n tokens, where each token is an element of $\{0, ..., v - 1\}$, where v is the vocabulary size. Each token is mapped to an embedding vector $\mathbf{e_i} \in \mathbb{R}^{\mathbf{m}}$, where i is the token number and m is the dimensionality of the embedding. The input, x, to the classification function, f, is then a sequence of n embedding vectors. 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 $x + \delta_x = x'$, where δ_x corresponds to only a small change in x (Goodfellow et al., 2015). This small change can be defined differently for different input modalities and methods (Madry et al., 2017; Guo et al., 2021). When attacking a NOTA-defended DNN classifier f with an input x', an attack is successful only if the prediction is not the original class y_x or the NOTA class y_{NOTA}. In other words, $f(x') \neq y_x \wedge f(x') \neq y_{NOTA}$ 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). In our threat model, the modifications to an input needed to change the predicted class of a model can be found using complete information of the classifier being attacked (i.e., a whitebox scenario).

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 $\mathcal{L}(x, y_x; f)$ 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:

$$x' = x + \alpha \text{ sign}(\nabla_x \mathcal{L}(x, y_x; f)), \tag{1}$$

where the gradient vector $\nabla_x \mathcal{L}(x, y_x, f)$ 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.

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

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 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:

$$\mathcal{L}_{\mathrm{DLR}}(x, y, f) = -\frac{z_y - \max\limits_{i \neq y}(z_i)}{z_{\pi_1} - z_{\pi_3}}, \tag{2}$$

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 loss of

$$\mathcal{L}_{CW}(x, x', y, f) = D(x, x') + \lambda_c \cdot \ell_{\mathrm{CW}}(x', y, f) \tag{3}$$

The CW constraint, ℓ_{CW}, term often takes the form of the margin loss

$$\ell_{\mathrm{CW}}(x', y, f) = \mathrm{ReLU}\big(z_y - \max_{i \neq y}(z_i) + \gamma\big), \tag{4}$$

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

The DeepFool attack (Moosavi-Dezfooli et al., 2016) is a greedy algorithm that attempts to approximate the minimum distance to the nearest, usually per an L_2 metric, 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

$$\min_{x' \in [0,1]^d} \mathcal{L}(x', y, f), \text{s.t. } ||x' - x||_p \leq \epsilon, \tag{5}$$

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 $\text{argmax}(f(x')) \neq y_{\text{true}}$. 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 & Hein (2020a), 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).

2.2.8 Gradient-Based Distributional Attack

Gradient-Based Distributional Attack (GBDA) is an adversarial evasion attack that targets transformers used in LLM-based text classification, seeking to produce fluent and semantically similar adversarial examples (Guo et al., 2021). A major challenge in this approach is overcoming the non-differentiable nature of the mapping from the input text token numbers to embeddings vectors. GBDA seeks to address this challenge by using the Gumbel softmax to create a differentiable method for linking token numbers with an embedding.

The GBDA objective function then balances three components in a weighted sum (i) a CW margin loss (see Section 2.2.4), (ii) a fluency constraint that preserves natural language flow, and (iii) a semantic similarity constraint that preserves meaning. This results in GBDA minimizing

$$\mathcal{L}(\Theta, y; f) = \mathbb{E}_{\bar{\pi} \sim P_\Theta} \Big[\underbrace{\ell_{\text{CW}}(e(\bar{\pi}), y; f)}_{\text{margin loss}} + \underbrace{\lambda_{lm} \text{NLL}_g(\bar{\pi})}_{\text{fluency}} + \underbrace{\lambda_{sim} \rho_g(x, \bar{\pi})}_{\text{semantic similarity}} \Big] . \tag{6}$$

In GBDA's objective function, P_Θ is the learned adversarial Gumbel-softmax distribution for a particular input. $\bar{\pi}$ is a sampled probability distribution from P_Θ over the vocabulary for each token in the input. e is a function that computes the expected embedding vectors according to $\bar{\pi}$. $\text{NLL}_g(\bar{\pi})$ is the perplexity (an analog of fluency) of the expected embeddings according to the language model g, which in this case is GPT-2 (Radford et al., 2019). ρ_g is the BERTScore (Zhang et al., 2019) (semantic similarity term) computed using g. $\lambda_{\text{lm}}, \lambda_{\text{sim}} > 0$, are hyper-parameters that adjust fluency and semantic similarity with attack strength. At each gradient-based update step, the attack draws multiple soft samples, $\bar{\pi}$ from P_Θ, calculates the loss $\mathcal{L}$ and then calculates the gradients to update Θ. After Θ has been learned, the attack samples text inputs using P_Θ until a successful adversarial text is generated or the maximum number of samples has been reached.

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, particularly for images, uses PGD, which, in many cases, leads to significantly improved adversarial robustness (Madry et al., 2017). Related gradient-based adversarial attacks have likewise been successfully employed to defend against adversarial attacks on NLP tasks.

Attacking to Training (A2T) is an adversarial training regime designed to defend LLM-based tools from adversarial attacks (Yoo & Qi, 2021). A2T introduces a gradient-based word-importance ranking derived from the gradient of the loss with respect to the embedding produced for each token in the input. It uses this basis to select the most impactful words to swap out and replace in order to achieve a successful attack (a change in classification). Once the attack is successful, the new example is added to the training dataset with a label that matches the original input's ground truth label. These examples are created on the fly during training as in other forms of adversarial training.

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. (2019) 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 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.

(a) No Defense (b) NOTA

Figure 1: A conceptual illustration of a linear classifier for a binary classification task without (a) and with (b) a NOTA defense.

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. This can lead to NOTA becoming a 'padding' class that separates data from other classes (Figure 1).

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, and a variant of A2T for LLM-based NOTA defense.

3.1 Taxonomy of Attack Failure Modes for NOTA Defenses

During our evaluations, we found that attacking NOTA-defenses with unadapted attacks often resulted in overwhelming defense successes against a broad range of attacks. On close inspection of attack code, we realized that most attacks share common failure modes when attacking an open-set defense incorporating a large, all-encompassing class like NOTA.[2] We found that one, or a combination, of three common failure modes often resulted in a less-than-thorough search of the input space for a potentially successful adversarial example. These three failure modes are an attack's *target selection*, its *stopping criteria*, and/or its *objective function*. These failure modes are described in Table 1. By addressing these failure modes, we expect attack evaluations to give a more accurate assessment of the adversarial robustness of NOTA and/or open-set defensed models.

3.1.1 Target Selection (F1)

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. 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 adversarial example out of that class. If NOTA is the initial predicted true label, the attack may perturb the adversarial example into the true class. Again, the solution is to exclude NOTA as a potential true label class as well.

3.1.2 Stopping Criteria (F2)

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

[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.

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:

$$(\text{argmax}(f(x')) \neq y_{NOTA}) \wedge (\text{argmax}(f(x')) \neq y_{True}) \tag{7}$$

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

3.1.3 Objective Function (F3)

If changes in target selection and stopping criteria do not restore attack effectiveness, the objective function should be examined. In some cases, an objective function which does not account for the 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 failure mode is present 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 they do not already exist. This leads to an attack loss of

$$\mathcal{L}(x, y_{\text{true}}; f) = \begin{cases} \text{CE}(y_{\text{true}}, \text{softmax}(f(x))) & \text{when } \beta \leq 0.5 \\ \text{CE}(y_{NOTA}, \text{softmax}(f(x))) & \text{otherwise,} \end{cases} \tag{8}$$

where $\beta \sim U(0, 1)$. β is used to randomly choose between pushing an adversarial example away from the true class and pushing the adversarial example away from the NOTA class.

3.3 NOTA A2T

To create NOTA examples for LLM-based models, we used a modified version of A2T (Yoo & Qi, 2021). Specifically, we labeled successful adversarial examples created by A2T (see Section 2.3.1) as the NOTA class, instead of the ground truth label. As in standard A2T, these examples are then used to augment the original training dataset during training.

3.4 Applying the Taxonomy to Adapt Attacks for NOTA

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.

3.4.1 NOTA Carlini-Wagner L_∞ and L_2

For a description of the original Carlini and Wagner's attacks, see Section 2.2.4. 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.

3.4.2 NOTA Deep Fool

The original DeepFool attack is described in detail in Section 2.2.5. 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.

3.4.3 NOTA Square Attack

The original Square Attack is described in Section 2.2.6. We adapted the attack by first preventing 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.

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

Both the CE and DLR versions of the APGD attack 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).

3.4.5 NOTA 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.

3.4.6 NOTA 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.

3.4.7 NOTA GBDA

The original GBDA attack is described in Section 2.2.8. For GBDA, it is necessary to modify both target selection (F1) and the stopping criterion (F2). As in CW attacks, the NOTA class must be eliminated from consideration when selecting a target for the margin loss. After learning the adversarial Gumbel softmax distribution for an input example, GBDA samples from this distribution until a successful adversarial example is found or the maximum number of samples is reached. The stopping criterion for this procedure must be modified so that a successful adversarial example is defined to be one that is both not classified as the ground truth class and not classified as NOTA.

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 3.4, 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 perform both computer vision and NLP experiments to evaluate NOTA defenses and NOTA-adapted attacks.

4.1.1 CIFAR-10 and CIFAR-100

We perform computer vision 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. 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.

For CIFAR-10 and CIFAR-100, we use standard Wide Residual Networks (WRNs) with dropout and batch normalization, configured as suggested by Zagoruyko & Komodakis (2016). 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.

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.

ASR is calculated on the selected 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 3.4.1—3.4.6. 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 $\epsilon = 0.5$.

4.1.2 IMDB

We perform NLP experiments using the IMDB binary sentiment classification dataset (Maas et al., 2011). We use the same training and test sets as in Guo et al. (2021). We train and evaluate a model that consists of BERT (Devlin et al., 2019) to create text embeddings, followed by a dropout layer with $p = 0.1$, a linear layer, and, finally, a softmax operation, as done in Morris et al. (2020); Guo et al. (2021). We fine-tune this model for 5 epochs uisng AdamW (Loshchilov & Hutter, 2017) with a batch size of 16, a learning rate of 2e-05, a weight decay of 0.01, and a maximum sequence length of 512.

The NOTA defense and the attack evaluated also required several hyperparameters to be set. When creating NOTA examples during training, we used similar hyperparameters to those used for adversarial training by Geisler et al. (2024). Namely, we started generating adversarial examples for the NOTA class after the model had been trained for one epoch and created a number of successful adversarial examples equal to 20% of the training dataset each epoch afterwards. For original and NOTA-adapted GBDA, we closely followed the hyperparameters used by Guo et al. (2021). Specifically, we used a learning rate of 0.3, a CW confidence of 5, a perplexity coefficient of 1, a similarity coefficient of 20, and 100 Gumbel samples.

4.2 Results

4.2.1 CIFAR-10 and CIFAR-100

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), with 95% binomial confidence intervals, for Original and 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 inherently adapted for NOTA.

4.2.2 Unadapted Attacks vs. Defenses

With unadapted attacks, as can be seen in Tables 3 and 4, we find that both BP and ANE appear highly robust. Even strong baselines like AutoAttack show low ASR at or below natural error, and several other attacks show substantial reductions as well. 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. However, these results stem from unadapted attacks treating NOTA as a regular class, meaning that they often stop once they cross the true class boundary–even if they result in a NOTA classification. This also explains the apparent anomaly of how Square Attack appears more effective on its own than AutoAttack (which includes Square Attack as a component). Once an earlier component attack reaches NOTA, the attack pipeline stops rather than continuing to a non-NOTA misclassification. To sum up, these unadapted results overstate robustness, each falling prey to some combination of the failures identified in our taxonomy.

Table 4: Attack Success Rates (ASRs), with 95% binomial confidence intervals, for Original and 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 inherently adapted for NOTA.

4.2.3 NOTA-Aware Adaptive Attacks vs. Defenses

Adapting attacks to the NOTA paradigm largely restores high ASR against both NOTA-defended models (Tables 3 and 4). The main exceptions are APGD-CE, CW L_∞, and DeepFool, where robustness remains closer to the unadapted baseline. By contrast, APGD-DLR and our APGD-AN consistently drive ASR up, demostrating the PGD-based NOTA training may not protect against other PGD-based attacks that use a different loss function. Importantly, no attack becomes weaker after being adapted based on our taxonomy.

For BP, NOTA-aware adaptations produce large ASR jumps, especially for APGD-DLR and AutoAttack. CW L_∞ and APGD-CE show limited gains–consistent with BP's CE-driven training–while SquareAttack and DeepFool change little. BP's apparent robustness under unadapted evaluation evaporates once target selection and stopping criteria exclude the NOTA "honeypot".

ANE looks strong against unadapted attacks, but is similarly vulnerable under NOTA-aware evaluation. On both CIFAR-10 and CIFAR-100, APGD-DLR, APGD-AN, and AutoAttack regain high ASR; CW L_2 is a partial exception (weaker on CIFAR-10, stronger on CIFAR-100). As with BP, SquareAttack and DeepFool remain close to their original behavior, indicating that whatever robustness is conferred by ANE against these attacks is not due to the failure modes for stopping criteria and target selection that we identify and correct. Notably, ANE retains some CE-specific resilience. However, attacks adapted using our taxonomy that were not used to generate NOTA examples during training are effective against ANE.

4.2.4 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. All results considered, ANE would appear to confer modestly 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 overall, with many attack options available to get 90% or greater attack success rates on either.

4.2.5 IMDB

In Table 5, the clean accuracies and ASRs for the LLM-based classifier with and without the NOTA A2T defense are reported. While the NOTA defense does decrease clean test set accuracy from 93.66% to 87.75%, it even more dramatically reduces the ASR of GBDA from 94.00% to 29.00%. 29.00% demonstrates a very effective defense, considering that the natural error rate of the NOTA-defended model is 12.12%. However,

when GBDA is adapted for NOTA using our proposed taxonomy, the ASR becomes 100.00%, demonstrating that NOTA-defended, LLM-based classifiers are still vulnerable to appropriately adapted adversarial evasion attacks.

Table 5: Test Clean Accuracies and Attack Success Rates (ASRs), with 95% binomial confidence intervals, for IMDB.

IMBD			
Model	**Clean Accuracy**	**GBDA ASR**	**NOTA GBDA ASR**
No Defense	93.66%	94.00% ± 4.65%	—
NOTA A2T	87.75%	29.00% ± 8.89%	100.00% ± 0.00%

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. These identified attack adaptations can be applied to any existing or future attacks to increase their effectiveness against open-set approaches to adversarial defense. Using these, we modify eight 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 for both computer vision and NLP tasks.

We observe that our adapted attacks clearly show that present NOTA defenses are not sufficient to defend against attacks adapted as outlined in this paper. However, NOTA defenses can confer some small residual resilience to some adapted attacks that at least rivals adversarial training for the analyzed computer vision datasets. With several adapted attacks recovering ASRs back to 90% and above for both the evaluated computer vision and NLP tasks, 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.

While these findings are critical, the evaluations of NOTA defenses and attacks could be improved in several ways. In this paper, the NOTA class is populated using methods based on common adversarial training approaches (Madry et al., 2017; Yoo & Qi, 2021). However, other promising methods exist for creating NOTA examples. One set of such methods are open-set training approaches (Ge et al., 2017; Chen et al., 2021). Also, other LLM adversarial attacks, could be adapted for NOTA defense, such as Geisler et al. (2024). Additionally, evaluations could be enhanced by using higher-resolution image classification datasets and more text tasks, such as Yelp Reviews (Zhang et al., 2015), AG News (Zhang et al., 2015), and LLM jailbreaking (Geisler et al., 2024).

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.

Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2021.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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].

ZongYuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017.

Simon Geisler, Tom Wollschläger, Johannes Gasteiger, and Stephan Günnemann. Attacking large language models with projected gradient descent. *arXiv preprint arXiv:2402.09154*, 2024.

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.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5747–5757, 2021.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

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.

John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *EMNLP 2020*, pp. 119, 2020.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI*, 2019.

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.

Jin Yong Yoo and Yanjun Qi. Towards improving adversarial training of nlp models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 945–956, 2021.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019. URL http://arxiv.org/abs/1904.09675.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.