AGREE: A SIMPLE AGGREGATOR OF DETECTORS' DECISIONS.

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a simple yet effective method to aggregate the decisions based on the soft-probability outputs of multiple trained detectors, possibly provided by a third party. We formally derive a mathematically sound theoretical framework, which is straightforward as it does not require further training of the given detectors, and modular, allowing existing (and future) detectors to be merged into a single one. As an application, we evaluate our framework by tackling the recently proposed problem of simultaneous adversarial examples detection, i.e. when the attacks at the evaluation time can be simultaneously crafted according to a variety of algorithms and objective loss functions. While each single detector tends to underperform or fail in the aforementioned attack scenario, our framework successfully aggregates the knowledge of the available detectors to guarantee a more reliable decision. We validate our AGgregatoR of dEtectors' dEcisions (AGREE) on popular datasets (e.g., CIFAR10 and SVHN) and we show that it consistently outperforms the state-of-the-art when simultaneous adversarial attacks are present at evaluation time.

1 INTRODUCTION

In recent years, the need for reliable deep models has sparked interest in the field of trustworthy AI across several research areas, such as misclassification detection (Granese et al., 2021; Geifman & El-Yaniv, 2019; Gangrade et al., 2021), out-of-distribution (OOD) detection (Gomes et al., 2022; Vyas et al., 2018; Sastry & Oore, 2020; Ovadia et al., 2019; Liu et al., 2020; Hendrycks & Gimpel, 2017; Zhang et al., 2021; Lin et al., 2021), robustness (Madry et al., 2018; Zhang et al., 2019; Alayrac et al., 2019; Picot et al., 2022; Robey et al., 2021; Engstrom et al., 2019), and adversarial attacks detection (Aldahdooh et al., 2022; Kherchouche et al., 2020; Ma et al., 2018; Lee et al., 2018; Meng & Chen, 2017; Xu et al., 2018; Feinman et al., 2017). One main difficulty all the aforementioned areas come across is proposing a solution that is generally better then the previously proposed ones in all scenarios. In particular, the community interested in adversarial attacks detection has invested a lot of effort in designing sophisticated defense strategies, which new attacks have systematically circumvented. However, as stated in Tramèr et al. (2020), there exists an informal "no-free-lunch-theorem": *For any proposed attack, it is possible to build a non-robust defence that prevents that attack.* Indeed, the question arises as to whether existing methods can be aggregated so to collect their individual expertise and achieve improved detection. Recently, Granese et al. (2022) pointed out a novel multi-armed framework, called MEAD, to assess the performances of the detectors when several attack strategies are perpetrated at the same time. Interestingly, the authors show that even without giving to the attacker the full knowledge on the underlying defence mechanism, the state-of-the-art (SOTA) detectors tragically fail under this setting.

To address this problem, we suggest a simple but still effective way to aggregate multiple detection methods building a "team of experts". The requirements for using one method in combination with others within our proposed framework are very flexible: in principle, as long as its output is interpretable as a probability distribution over the two categories natural/adversarial, any existing (or future) method, either supervised or unsupervised, can be combined with others using the framework we propose in this work. Crucially, a modular aggregator allows pre-trained detectors to be reused, virtually free of charge, without additional training or data.

1.1 SUMMARY OF CONTRIBUTIONS

Our contributions are threefold:

- To the best of our knowledge, this work is the first one that proposes an aggregation framework to combine the expertise of different adversarial examples detectors and address the problem raised in Granese et al. (2022). Our proposed method can aggregate detectors off the shelf, with no further training required.
- From a theoretical perspective, we revisit the *simultaneous* attack detection problem in Granese et al. (2022) and we formalize it as a minimax cross-entropy risk. Based on this formulation, we derive a surrogate loss from which we characterize our optimal soft-detector in Eq. (10) that leads to AGREE (cf. Definition 1). (Sec. 3).
- We empirically evaluate AGREE on popular datasets (e.g., CIFAR10 and SVHN). In particular we test it over simple detectors which individually perform much worse than SOTA. We show that AGREE, through their aggregation, leads to higher and more consistent performance w.r.t. SOTA (cf. Sec. 4) over the simultaneous attack setup.

1.2 RELATED WORKS

Detection mechanisms. Methods to defend deep models against adversarial attacks can be grouped into two main families: methods that are designed to increase the targeted model's robustness by re-training it Goodfellow et al. (2015); Madry et al. (2018); Picot et al. (2022); Xie et al. (2020); Tramèr et al. (2018), and methods engineered to detect adversarial examples at evaluation time Kherchouche et al. (2020); Ma et al. (2018); Feinman et al. (2017); Xu et al. (2018); Meng & Chen (2017); Lee et al. (2018). The work in Aldahdooh et al. (2022) provides a recent and thorough survey about the state-of-the-art detection methods, which fall under two main categories: *supervised* and *unsupervised*. Detectors within the former category extract features either directly from the targeted network's layer Kherchouche et al. (2020); Feinman et al. (2017) or by using statistical tools Ma et al. (2018); Lee et al. (2018). To do so, both natural and adversarial examples are necessary. Generally, the adversarial samples are created according to a single fixed algorithm and a given loss function, which are then also used to create the examples at evaluation time. Methods falling under the unsupervised category only rely on the features of natural samples that can be extracted using different techniques (e.g., *feature squeezing* Xu et al. (2018)) or can be based on autoencoders training procedures with the scope of minimizing the reconstruction error Meng & Chen (2017).

Attack algorithms. Since Szegedy et al. (2014) first shed light on the problem, several machine learning models, including state-of-the-art neural networks, have been found to be vulnerable to adversarial examples. Over the years, a plethora of algorithms to generate adversarial samples has been proposed and, overall, we can group them into two main categories: *whitebox* and *blackbox* attacks. We talk about *white-box* attacks when the adversary knows everything about the target model (its architecture and weights). *Gradient-based* attacks belong to this category. They rely on finding the perturbation direction, i.e., the sign of gradient at each pixel of the input, that maximizes the attacker's objective value. Examples of gradient-based attacks are the *Fast Gradient Sign Method* (FGSM) Goodfellow et al. (2015), the *Basic Iterative Method* (BIM) Kurakin et al. and the *Projected Gradient Descent* method (PGD) Madry et al. (2018). BIM and PGD can be seen as iterative versions of FGSM (one-step perturbation). Unlike BIM, PGD attacks start from a random perturbation in L_p-ball around the input sample. Another powerful attack is the *Carlini-Wagner* attack (CW) Carlini & Wagner (2017c), which directly minimizes the additive noise constrained by a function which assure the misclassification of the perturbed sample. We conclude the list of white-box attacks by mentioning the *DeepFool attack* (DF) Moosavi-Dezfooli et al. (2016), which is an iterative method based on a local linearization of the targeted classifier, and the resolution of the resulting simplified adversarial problem. In the case of *black-box* attacks, the adversary has no access to the internals of the target model, hence it creates attacks by querying the model and monitoring outputs of the model to attack. Examples of black-box attacks are the *Square Attack* (SA) Andriushchenko et al. (2020), which iteratively searches for a random perturbation, and checks if it increases the attacker's objective at each step; the *Hop Skip Jump* attack (HOP) Chen et al. (2020) which estimates the gradient direction to perturb, and the *Spatial Transformation Attack* (STA) Engstrom et al. (2019) which transforms the original samples by applying small translations and rotations to them. It is

worth to mention that there also exists *gray-box* attacks, i.e. when the adversary knows the training data but not the internals of the model. These attacks rely on the transferability property of the adversarial examples: to create attacks these methods build a substitute model that performs the same task as the target model. A special class of attacks are the so-called *adaptive attacks* (Athalye et al., 2018; Tramèr et al., 2020; Carlini & Wagner, 2017c; Yao et al., 2021) where attacks are specifically designed to target a given defence. In this scenario, the attacker is supposed to have full knowledge of both the targeted classifier and the underlying defence.

We refer to the survey in Aldahdooh et al. (2022) and references therein for a comprehensive discussion of these topics.

2 Main Definitions and Preliminaries

2.1 Target classifier

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space and let $\mathcal{Y} = \{1, \dots, C\}$ be the label space related to a classification task. We denote by P_{XY} the unknown data distribution over $\mathcal{X} \times \mathcal{Y}$. Throughout the paper, we refer to the *classifier* with $p_{\widehat{Y}|X}(y|\mathbf{x};\theta)$, i.e. the parametric soft-probability model, where $y \in \mathcal{Y}$, and $\theta \in \Theta$ are the learnt parameters. The function $h_\theta : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ outputs the logits vector of the classifier given an input sample. The induced hard decision of the classifier is defined as $g_\theta : \mathcal{X} \to \mathcal{Y}$ s.t. $g_\theta(\mathbf{x}) \triangleq \arg\max_{y \in \mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta)$.

2.2 Adversarial problem

Let us consider a natural sample $\mathbf{x} \in \mathcal{X}$ together with its true label $y \in \mathcal{Y}$. An attacker targets the model g_θ by crafting a sample $\mathbf{x}'_\ell \in \mathbb{R}^d$ according to an objective loss function $\ell(\mathbf{x}, \mathbf{x}'_\ell; \theta)$ which is denoted by ℓ, perturbation magnitude ε, and norm constraint L_p, $p \in \{1, 2, \infty\}$. The goal of the attack is to obtain an $\mathbf{x}'_\ell$ such that $g_\theta(\mathbf{x}'_\ell) \neq g_\theta(\mathbf{x})$, in order to force the target model to make a prediction error. As thoroughly investigated in Szegedy et al. (2014), the adversarial generation problem is difficult to tackle and it is commonly relaxed as follows

$$\mathbf{x}_\ell{}' \equiv \mathbf{x}_\ell{}'(\mathbf{x}) = \underset{\mathbf{x}_\ell{}' \in \mathbb{R}^d \,:\, \|\mathbf{x}_\ell{}' - \mathbf{x}\|_p < \varepsilon}{\arg\max} \ell(\mathbf{x}, \mathbf{x}_\ell{}'; \theta), \tag{1}$$

where $\mathbf{x}'_\ell$ is updated iteration by iteration starting from an initial given value. The objective function ℓ traditionally used is the Adversarial Cross-Entropy (ACE) Szegedy et al. (2014); Madry et al. (2018):

$$\ell_{\text{ACE}}(\mathbf{x}, \mathbf{x}_\ell{}'; \theta) = \mathbb{E}_{Y|\mathbf{x}}\big[-\log p_{\widehat{Y}|X}(Y|\mathbf{x}'_\ell;\theta)\big], \tag{2}$$

where the expectation is understood to be over the ground true conditional distribution of Y given $\mathbf{x}$. Inspired by recent development in the fields of robustness and misclassification detection (Granese et al., 2021; Picot et al., 2022; Zhang et al., 2019), Granese et al. (2022) have included in their study recently proposed objective functions which generate diversified adversarial examples and that we briefly recall below.

- The Kullback-Leibler divergence (KL):

$$\ell_{\text{KL}}\left(\mathbf{x}, \mathbf{x}_\ell{}'; \theta\right) = \mathbb{E}_{\widehat{Y}|\mathbf{x};\theta}\left[\log\left(\frac{p_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x};\theta)}{p_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x}'_\ell;\theta)}\right)\right]. \tag{3}$$

- The Fisher-Rao objective (FR) Picot et al. (2022):

$$\ell_{\text{FR}}(\mathbf{x}, \mathbf{x}_\ell{}'; \theta) = 2\arccos\left(\sum_{y \in \mathcal{Y}} \sqrt{p_{\widehat{Y}|X}(y|\mathbf{x};\theta) p_{\widehat{Y}|X}(y|\mathbf{x}'_\ell;\theta)}\right). \tag{4}$$

- The Gini Impurity score (Gini) Granese et al. (2021):

$$\ell_{\text{Gini}}(\cdot, \mathbf{x}_\ell{}'; \theta) = 1 - \sqrt{\sum_{y \in \mathcal{Y}} p^2_{\widehat{Y}|X}(y|\mathbf{x}'_\ell;\theta)}. \tag{5}$$

3 FORMALIZATION OF THE PROBLEM OF DETECTING SIMULTANEOUS ATTACKS

3.1 STATISTICAL MODEL

Let $\mathcal{K}$ be the countable set of indexes corresponding to each possible attack, e.g., based on various attack algorithms and loss functions, as described in Sec. 2.2. Let $\mathcal{M} = \left\{P_{XZ}^{(k)} : k \in \mathcal{K}\right\}$ be the set of joint probability distributions on $\mathcal{X} \times \mathcal{Z}$ which are indexed with k, $\forall k \in \mathcal{K}$, where $\mathcal{X}$ is the input (feature) space and $\mathcal{Z} = \{0, 1\}$ indicates a binary space label for the adversarial example detection task. At the evaluation time, the attacker selects an arbitrary strategy $k \in \mathcal{K}$ and then samples an input according to $p_{X|Z}^{(k)}(\mathbf{x}|z = 1)$ which corresponds to the probability density function induced by the chosen attack k where $p_{X|Z}^{(k)}(\mathbf{x}|z = 0) = p_X(\mathbf{x})$ *almost surely* corresponding to the probability distribution of the natural samples. The learner is given a set of *soft-detectors* models:

$$\mathcal{Q} = \left\{q_{\widehat{Z}|\mathbf{u}}^{(k)} : \mathcal{U} \mapsto [0, 1]^2\right\}_{k \in \mathcal{K}},$$

which have possibly been trained to detect attacks according to each strategy $k \in \mathcal{K}$, e.g., $q_{\widehat{Z}|\mathbf{u}}^{(k)} \equiv p_{\widehat{Z}|U}(z|\mathbf{u}; \psi_k)$ with parameters ψ_k and $\mathbf{u} \in \mathcal{U} = \{h_\theta(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d\}$ denotes the space of logits. The set of possible detectors $\mathcal{Q}$ is available to the defender. However, the specific attack chosen by the attacker at the test time is unknown. In the remainder of this section, we formally devise an optimal detector that exploits full knowledge of the set $\mathcal{Q}$.

3.2 A NOVEL OBJECTIVE FOR DETECTION UNDER SIMULTANEOUS ATTACKS

Consider a fixed input sample $\mathbf{x_0}$ and let $\mathbf{u_0} = h_\theta(\mathbf{x_0})$. Clearly, the problem at hand consists in finding an optimal soft-detector $q_{\widehat{Z}|\mathbf{u_0}}^\star$ that performs well simultaneously over all possible attacks in $\mathcal{K}$. This can be formalized as the solution to the following minimax problem:

$$\mathcal{L}(\mathcal{Q}, \mathbf{x_0}) = \min_{q_{\widehat{Z}|\mathbf{u_0}}} \max_{k \in \mathcal{K}} \mathbb{E}_{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}} \left[-\log q_{\widehat{Z}|\mathbf{u_0}}\right], \tag{6}$$

which requires to solve equation 6 for $\mathcal{Q}$ and for each given input sample $\mathbf{x_0}$. Unfortunately, this objective is not tractable computationally. To overcome this issue, we derive a surrogate (an upper bound) that can be computationally optimized. For any arbitrary choice of $q_{\widehat{Z}|\mathbf{u_0}}$, we have

$$\max_{k \in \mathcal{K}} \mathbb{E}_{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}} \left[-\log q_{\widehat{Z}|\mathbf{u_0}}\right] \leq \underbrace{\max_{k \in \mathcal{K}} \mathbb{E}_{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}} \left[-\log q_{\widehat{Z}|\mathbf{u_0}}^{(k)}\right]}_{=\text{constant term}} + \max_{k \in \mathcal{K}} \mathbb{E}_{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}} \left[\log\left(\frac{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right]. \tag{7}$$

Observe that the first term in equation 7 of the upper bound is constant w.r.t. the choice of $q_{\widehat{Z}|\mathbf{u_0}}$ and the second term is well-known as being equivalent to the *average worst-case regret* Barron et al. (1998). This upper bound provides a surrogate to our intractable objective in equation 6 that can be minimized over all $q_{\widehat{Z}|\mathbf{u_0}}$. We can formally state our problem as follows:

$$\tilde{\mathcal{L}}(\mathcal{Q}, \mathbf{x_0}) = \min_{q_{\widehat{Z}|\mathbf{u_0}}} \max_{k \in \mathcal{K}} \mathbb{E}_{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}} \left[\log\left(\frac{q_{\widehat{Z}|\mathbf{u_0}}^{(k)}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right] = \min_{q_{\widehat{Z}|\mathbf{u_0}}} \max_{P_\Omega} \mathbb{E}_\Omega \left[D_{\mathrm{KL}}\left(q_{\widehat{Z}|\mathbf{u_0}}^{(\Omega)} \middle\| q_{\widehat{Z}|\mathbf{u_0}}\right)\right], \tag{8}$$

where the $\min$ is taken over all the possible distributions $q_{\widehat{Z}|\mathbf{u_0}}$; and $\Omega \in \mathcal{K}$ is a discrete random variable with P_Ω denoting a generic probability distribution whose probabilities are $(\omega_1, \ldots, \omega_{|\mathcal{K}|})$, i.e., $P_\Omega(k) = \omega_k$; and $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback–Leibler divergence, representing the expected value of regret of $q_{\widehat{Z}|U}$ w.r.t. the worst-case distribution in the class $\mathcal{Q}$. The convexity of the KL-divergence allows us to rewrite Eq. (8) as follows:

$$\min_{q_{\widehat{Z}|\mathbf{u_0}}} \max_{P_\Omega} \mathbb{E}_\Omega \left[D_{\mathrm{KL}}\left(q_{\widehat{Z}|\mathbf{u_0}}^{(\Omega)} \middle\| q_{\widehat{Z}|\mathbf{u_0}}\right)\right] = \max_{P_\Omega} \min_{\widehat{q}_{\widehat{Z}|\mathbf{u_0}}} \mathbb{E}_\Omega \left[D_{\mathrm{KL}}\left(q_{\widehat{Z}|\mathbf{u_0}}^{(\Omega)} \middle\| q_{\widehat{Z}|\mathbf{u_0}}\right)\right]. \tag{9}$$

The solution Eq. (9) provides the optimal distribution $P^{\star}_{\Omega}$, i.e. the collection of weights $\{w^{\star}_k\}$, which leads to our soft-detector Barron et al. (1998):

$$\widehat{q}^{\star}_{\widehat{Z}|\mathbf{u_0}} = \sum_{k \in \mathcal{K}} w^{\star}_k \cdot q^{(k)}_{\widehat{Z}|\mathbf{u_0}}, \quad \text{with} \quad P^{\star}_{\Omega} = \arg\max_{\{\omega_k\}} I_{\mathbf{u_0}}(\Omega; \widehat{Z}), \tag{10}$$

where $I_{\mathbf{u_0}}(\cdot;\cdot)$ denotes the Shannon mutual information between the random variable Ω, distributed according to $\{\omega_k\}$, and the binary soft-prediction variable $\widehat{Z}$, distributed according to $q^{(k)}_{\widehat{Z}|\mathbf{u_0}}$ and conditioned on the particular test example $\mathbf{u_0}$. Further details are provided in Appendix A.

From theory to our practical detector. According to our derivation in Eq. (10), the optimal detector turns out to be given by a mixture of the $|\mathcal{K}|$ detectors belonging to the class $\mathcal{Q}$, with weights carefully optimized to maximize the mutual information between K and the predicted variable $\widehat{Z}$ for each detector in the class $\mathcal{Q}$. Using this key ingredient, it is straightforward to devise our optimal detector.

Definition 1 (AGREE) *For any $0 \leq \gamma \leq 1$ and a given $\mathbf{x_0} \in \mathcal{X}$, let us define the following detector:*

$$\text{AGREE}(\mathbf{x}_0) \triangleq \mathbb{1}\left[q^{\star}_{\widehat{Z}|\mathbf{u_0}}(\hat{z} = 1 | h_\theta(\mathbf{x_0})) > \gamma\right], \tag{11}$$

where $\mathbb{1}[\cdot]$ is the indicator function.

4 EXPERIMENTAL RESULTS

We evaluate our proposed framework AGREE deploying it against the attacks in MEAD, i.e., the simultaneous attacks scenario introduced in Granese et al. (2022). The goal of AGREE is to detect at evaluation time the attacks that can be simultaneously crafted according to a variety of algorithms and objective loss functions. To reproduce our results, we provide our source code in the Supplementary Material.

In our empirical evaluation, we suppose a third party provides us four simple supervised detectors. Each of them is trained to detect only a specific kind of attack. Note that this is a reasonable assumption, as many methods in the literature are good at detecting one type of attacks and fail at detecting others. In addition, to emphasize the role played by the proposed method, these detectors are shallow networks (fully-connected, 3 layers of 256 nodes each), which are only allowed to observe the logits of the target classifier to distinguish between natural and adversarial samples. Each of these detectors is designed to successfully recognize only one specific kind of attack. Therefore, they are bound to perform very poorly (much worse than the SOTA) against attacks they have not been trained on.

Table 1: MEAD. Each cell corresponds to attacks simultaneously executed on the targeted classifier. Attacks created using all the losses in Sec. 2.2 are marked with *. Attacks such as SA and DF are not dependent on the choice for the loss. Empty cells correspond to combinations of perturbation magnitude and norm constraint that have not been considered.

	L_1	L_2	L_∞	No norm
$\varepsilon = 0.01$	-	CW2	-	-
$\varepsilon = 0.03125$	-	-	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.0625$	-	-	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.1$	-	HOP	-	-
$\varepsilon = 0.125$	-	PGD2*	PGDi*,FGSM*,BIM*,SA	-
$\varepsilon = 0.25$	-	PGD2*	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.3125$	-	PGD2*	PGDi*,FGSM*,BIM*,CWi	-
$\varepsilon = 0.5$	-	PGD2*	PGDi*,FGSM*,BIM*	-
$\varepsilon = 1$	-	PGD2*	-	-
$\varepsilon = 1.5$	-	PGD2*	-	-
$\varepsilon = 2$	-	PGD2*	-	-
$\varepsilon = 5$	PGD1*	-	-	-
$\varepsilon = 10$	PGD1*	-	-	-
$\varepsilon = 15$	PGD1*	-	-	-
$\varepsilon = 20$	PGD1*	-	-	-
$\varepsilon = 25$	PGD1*	-	-	-
$\varepsilon = 30$	PGD1*	-	-	-
$\varepsilon = 40$	PGD1*	-	-	-
No ε	-	DF	-	-
max. rotation = 30 max. translation = 8	-	-	-	STA

4.1 EVALUATION FRAMEWORK

Evaluation setup: MEAD. According to the simultaneous attacks framework of MEAD (Granese et al., 2022), we consider all the attack algorithms mentioned in Sec. 1.2, and we group them w.r.t. the corresponding norm and the perturbation magnitude. For each natural sample and for each gradient-based attack algorithm (i.e., FGSM, PGD or BIM), we create four adversarial counterparts, each corresponding to one of the loss functions described in Sec. 2.2. In order to clearly explain how the evaluation is carried on, let us take a look at Tab. 1. Each cell corresponds to a group of attacks: each of them has been created according to the algorithm within the cell, the associated norm (i.e., the column label) and perturbation magnitude (i.e., the row label) and one of the four loss functions. Thus for example, when we consider L_∞ norm and $\varepsilon = 0.125$, the detector is evaluated on $4 + 4 + 4 + 1 = 13$ simultaneous attacks. Note that we discard the perturbed examples that do not fool the classifier as, by definition, they are neither natural nor adversarial.

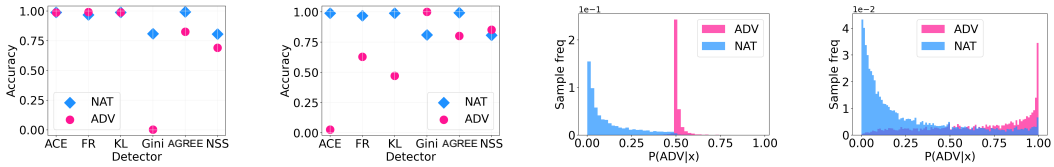
Evaluation metrics. Following the evaluation setup described above, and w.r.t. each cell of Tab. 1, throughout the paper we consider a detection successful if and only if all the attacks within the considered cell are detected. Hence, we say that a group of attacks corresponding to a cell counts as a true positive if and only if *all* the corresponding adversarial examples are detected as adversarial, as a false negative otherwise. We stress the fact that, to the best of our knowledge, this evaluation is stricter and more realistic than the ones considered so far in the literature. We use the classical definitions of *true negative* and *false positive* for the detection of natural samples. This means that a true negative is a natural sample detected as natural, a false positive is a natural sample detected as adversarial. We measure the performance of the detectors in terms of: i) AUROC$\uparrow$% (Davis & Goadrich, 2006) (the *Area Under the Receiver Operating Characteristic curve*) which represents the ability of the detector to discriminate between adversarial and natural examples (higher is better); ii) FPR at 95 % TPR (FPR$\downarrow_{95\%}$%), i.e., the percentage of natural examples detected as adversarial when 95 % of the adversarial examples are detected (lower is better).

Datasets and pre-trained classifiers. We run our experiments on CIFAR10 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011) image datasets. For both of them, we consider as pre-trained classifier ResNet-18 models that have been trained for 100 epochs, using SGD optimizer with a learning rate of 0.1, weight decay of 10^{-5}, and momentum of 0.9. The accuracy achieved by the classifiers on the original clean data are 99% for CIFAR10 and 100% for SVHN at training time; 93.3% for CIFAR10 and 95.5% for SVHN at testing time.

Detectors. AGREE aggregates four simple pre-trained detectors. The detectors are four fully-connected neural networks, composed of 3 layers of 256 nodes each. All the detectors are trained for 100 epochs, using SGD optimizer with learning rate of 0.01 and weight decay 0.0005. They are trained to distinguish between natural and adversarial examples created according to the PGD algorithm, under L_∞ norm constraint and perturbation magnitude $\varepsilon = 0.125$ for CIFAR10 and $\varepsilon = 0.25$ for SVHN. Each detector is trained on natural and adversarial examples generated using one of the loss functions mentioned in Sec. 2.2 (i.e., ACE Eq. (2), KL Eq. (3), FR Eq. (4), or Gini Eq. (5)) to craft its training adversarial samples. We would like to point out that the purpose of this paper is not creating a new supervised detector, but rather showing a method to aggregate a set of detectors, either supervised or unsupervised, and already trained. We further expand on the selection of the ε parameter of the adversarial examples used at training time in Appendix B.3 (c.f. Tabs. 5 and 7).

NSS (Kherchouche et al., 2020). We compare AGREE with NSS, which turns out to be the best among the SOTA methods under simultaneous attacks (c.f. (Granese et al., 2022)). NSS characterizes the adversarial perturbations through the use of *natural scene statistics*, i.e. statistical properties that can be altered by the presence of adversarial perturbations. NSS is trained once by using PGD algorithm, L_∞ norm constraint and perturbation magnitude $\varepsilon = 0.03125$ for CIFAR10 and $\varepsilon = 0.0625$ for SVHN. We further expand on the selection of the ε parameter of the adversarial examples used at training time in Appendix B.3 (c.f. Tabs. 4 and 6).

On the optimization of Eq. (10) For the optimization of Eq. (10), we rely on the `SciPy` (Virtanen et al., 2020) library, package `optimize`, function `minimize` which uses the *Sequential*



(a) Detectors on PGD-L_1-40-FR (b) Detectors on FGSM-L_∞-0.5-FR (c) AGREE on PGD-L_1-40-FR (d) NSS on PGD-L_1-40-FR

Figure 1: Discrimination performances. In Fig. 1a and Fig. 1b, the accuracies of the detectors on natural and adversarial examples; in Fig. 1c and Fig. 1d we show how AGREE and NSS split the data samples. In pink the results for the adversarial examples and in blue the ones for the naturals. Under each plot the tested attack configuration parameters: algorithm-L_p-ε-loss.

Least Squares Programming (SLSQP) algorithm to find the optimum. Further details can be found in Appendix A.2.

4.2 NUMERICAL RESULTS

We graphically present the intuition behind the way AGREE works, then we move on to present and discuss a collection of experimental results. We relegate to Appendix B the further discussions on the experiments that for space constraints have not been included in this section.

4.2.1 THE *shallow* DETECTORS IN AGREE

Figure 1 provides a graphical interpretation of the detection performance when ResNet18, trained on CIFAR10, is the target classifier (c.f. Sec. 4.1). In Figs. 1a and 1b the single detectors are referred to as loss used to create the adversarial examples on which each detector was trained (along with the natural samples), i.e ACE, FR, KL, Gini along the horizontal axis.

In Figure 1a, and Figure 1b, we report the accuracy for the adversarial detection task on the natural examples in blue, and on the adversarial examples in pink. We consider each individual detector, along with NSS and AGREE. In Figure 1a, the attacks have been created according to the PGD algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_1. In Figure 1b the attacks have been created according to the FGSM algorithm, FR loss, $\varepsilon = 0.5$, and L_∞ norm. As we can observe the individual detectors, i.e. ACE, FR, KL and Gini exhibit different behaviours according to the current attack. In fact, in Fig. 1a the Gini detector drastically fails at detecting the attack as its accuracy plummets to 0% on the adversarial examples. In the same way, FR, KL but mostly ACE perform poorly against FGSM (c.f. Fig. 1b). On the contrary, AGREE, benefiting from the aggregation, obtains favorable results in both cases.

One main takeaway of this paper is that, if we are provided with generally mediocre detectors, whose performance is good only on a limited amount of cases, we can successfully aggregate them through AGREE in order to obtain a more consistent detection.

Further insight into the way AGREE works is provided by figures Fig. 1c and Fig. 1d. The histograms show how AGREE and NSS separate natural (blue) and adversarial examples (pink). The values along the horizontal axis represent the probability of being classified as adversarial. In each plot, and according to the corresponding discrimination method, the bins' heights represent the frequency of the samples whose associated probability of being adversarial falls within that bin. The detection error is proportional to the area of overlap between the blue and the pink histograms. Fig. 1c and Fig. 1d suggest that AGREE achieves lower detection error on the considered attack, as it is confirmed by Tab. 2 where we found NSS with 76.1 AUROC↑% and AGREE with 92.1 AUROC↑%. Additional plots are provided in Appendix B.5.

Indeed, AGREE's performance over the simultaneous attacks are consistent over the majority of the considered attacks (c.f. Tab. 2).

Table 2: Comparison between AGREE and NSS on CIFAR10 and SVHN. The * symbol means the perturbation mechanism is executed in parallel four times starting from the same original clean sample, each time using one of the objective losses between ACE Eq. (2), KL Eq. (3), FR Eq. (4), Gini Eq. (5).

	CIFAR10				SVHN			
	NSS		AGREE		NSS		AGREE	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1								
PGD1*								
$\varepsilon = 5$	48.5	94.2	**62.1**	**87.1**	40.2	91.3	**76.9**	**79.0**
$\varepsilon = 10$	54.0	**90.3**	**56.8**	90.6	36.9	91.3	**73.0**	**82.5**
$\varepsilon = 15$	58.8	86.8	**69.3**	**84.4**	35.6	91.3	**78.9**	**72.5**
$\varepsilon = 20$	63.5	82.3	**78.7**	**73.1**	36.1	91.3	**83.6**	**60.7**
$\varepsilon = 25$	67.7	77.2	**87.1**	**50.8**	37.8	91.3	**87.0**	**48.6**
$\varepsilon = 30$	71.4	73.4	**90.3**	**35.4**	39.8	91.3	**89.3**	**37.2**
$\varepsilon = 40$	76.1	67.3	**92.1**	**26.4**	43.1	91.3	**92.6**	**20.0**
Norm L_2								
PGD2*								
$\varepsilon = 0.125$	48.3	94.3	**63.9**	**85.4**	40.8	91.3	**80.2**	**74.5**
$\varepsilon = 0.25$	53.2	91.2	**57.1**	**90.5**	37.2	91.3	**74.0**	**81.7**
$\varepsilon = 0.3125$	55.8	89.2	**61.0**	**88.9**	36.1	91.3	**75.2**	**79.4**
$\varepsilon = 0.5$	63.3	82.6	**79.4**	**73.2**	35.9	91.3	**82.5**	**64.4**
$\varepsilon = 1$	76.4	67.5	**91.4**	**26.4**	42.5	91.3	**92.3**	**24.7**
$\varepsilon = 1.5$	81.0	63.0	**91.9**	**24.2**	46.3	91.3	**94.1**	**7.5**
$\varepsilon = 2$	82.6	62.3	**91.9**	**24.1**	49.8	91.3	**94.9**	**5.3**
DeepFool								
No ε	57.0	91.7	**81.9**	**54.8**	41.3	91.3	**94.9**	**12.0**
CW2								
$\varepsilon = 0.01$	**56.4**	**90.8**	53.4	92.2	41.0	**91.3**	**54.2**	92.0
HOP								
$\varepsilon = 0.1$	66.1	87.0	**86.1**	**49.1**	67.6	84.2	**96.0**	**10.2**
Norm L_∞								
PGDi*, FGSM*, BIM*								
$\varepsilon = 0.03125$	**83.0**	**55.3**	82.3	59.7	**86.3**	**46.9**	81.4	64.9
$\varepsilon = 0.0625$	**96.0**	**17.2**	92.0	29.6	88.9	**0.7**	**89.1**	33.3
$\varepsilon = 0.25$	**97.3**	**0.6**	95.9	8.8	51.6	88.9	**92.3**	**16.4**
$\varepsilon = 0.5$	82.5	100.0	**94.6**	**9.7**	46.7	86.7	**92.9**	**14.4**
PGDi*, FGSM*, BIM*, SA								
$\varepsilon = 0.125$	9.4	99.9	**88.9**	**40.8**	32.9	91.3	**89.2**	**29.1**
PGDi*, FGSM*, BIM*, CWi								
$\varepsilon = 0.3125$	63.2	99.1	**80.0**	**61.1**	41.3	91.3	**88.2**	**33.1**
No norm								
STA								
No ε	**88.5**	**38.8**	82.7	52.4	**91.2**	**0.2**	90.2	23.2

Table 3: Comparison between AGREE and AGREE+*competitor* (NSS (a) and FS (b)) on CIFAR10. The * symbol means the perturbation mechanism is executed in parallel four times starting from the same original clean sample, each time using one of the objective losses between ACE Eq. (2), KL Eq. (3), FR Eq. (4), Gini Eq. (5). We focus only in the cases in which AGREE is outperformed from the corresponding competitors.

	CIFAR10			
	AGREE		AGREE+NSS	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_2				
CW2				
$\varepsilon = 0.01$	53.4	92.2	**54.1**	**91.3**
Norm L_∞				
PGDi*, FGSM*, BIM*				
$\varepsilon = 0.03125$	82.3	59.7	**89.9**	**34.4**
$\varepsilon = 0.0625$	92.0	29.6	**96.4**	**9.0**
$\varepsilon = 0.25$	95.9	8.8	**96.7**	**3.5**
No norm				
STA				
No ε	82.7	52.4	**87.3**	**35.4**

(a) AGREE+NSS (supervised)

	CIFAR10			
	AGREE		AGREE+FS	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1				
PGD1*				
$\varepsilon = 5$	62.1	87.1	**69.4**	**74.5**
$\varepsilon = 10$	56.8	90.6	**76.8**	**64.5**
$\varepsilon = 15$	69.3	84.4	**77.6**	**60.3**
Norm L_2				
PGD2*				
$\varepsilon = 0.125$	63.9	85.4	**67.9**	**76.4**
$\varepsilon = 0.25$	57.1	90.5	**76.0**	**64.7**
$\varepsilon = 0.3125$	61.0	88.9	**77.2**	**62.9**
CW2				
$\varepsilon = 0.01$	53.4	92.2	**86.4**	**46.8**

(b) AGREE+FS (unsupervised)

4.2.2 AGREE IN THE MEAD SCENARIO

AGREE turns out to be more general than NSS at recognizing simultaneous attacks on all the considered datasets. On CIFAR10, AGREE's maximum AUROC improvement w.r.t. NSS is 79.5 percentage points and happens for attacks under L_∞-norm constraint, $\varepsilon = 0.125$ and PGD*, FGSM*, BIM*, SA, i.e., when as many as 13 different simultaneous attacks are mounted. Similarly, AGREE's maximum FPR at 95% of TPR improvement w.r.t. NSS is 90.3 percentage points and happens for

attacks under L_∞-norm constraint, $\varepsilon = 0.5$ and PGD*, FGSM*, BIM*, i.e., when as many as 12 different simultaneous attacks are mounted. AGREE outperforms NSS in the case of the attacks with L_1 and L_2 norm, regardless of the algorithm or the perturbation magnitude, and in the case of L_∞ norm with large perturbations. However, for the attacks with L_∞ norm and small ε, although AGREE's performance is comparable to that of NSS, we notice a slight degradation. To shed a light on this, we remind that individual detectors aggregated by AGREE are based on the classifier's logits; NSS, on the other hand, extracts natural scene statistics from the inputs (see Sec. 4.1). This more sophisticated technique leads NSS to perform well when tested on attacks made with similar ε and same norm as the ones seen at training time. Similar conclusions can be drawn for the results on SVHN (c.f. Tab. 2).

Tab. 3 shows the modularity of AGREE when NSS (a) and FS (b) are plugged in as a fifth detector. We test AGREE+*competitor* on the attacks on which AGREE was outperformed by the competitors. In all the cases, AGREE+*competitor* outperforms AGREE either in terms of AUROC and FPR. Also, in most of the cases, AGREE+*competitor* is also better than the individual competitor. In particular, remarkable is the case of CW2 attack, where the performance of AGREE+FS improves of 33 percentage points in terms of AUROC and 45.4 percentage points in terms of FPR.

4.3 ADDITIONAL RESULTS CONSIDERING ADAPTIVE ATTACKS AND THE NON-SIMULTANEOUS SETTING

Due to the relevance of adaptive attacks in the field of adversarial attacks detection, we have tested out aggregation method against such threats. An adversary which is able to adapt its attack strategy to the very same detectors aggregated by AGREE is also able to lower the performance of our proposed method. **It is important to note that AGREE aggregates the provided detectors, and that making them more robust is not part of its design**. Keeping this in mind, we reference Appendix B.3.1 for further details and the related experimental results. Due to space constraint we relegate to Appendix B.3.2 the results showing that AGREE is able to improve the detection performance against specific individual attacks, and is valuable in this scenario as well as in the MEAD scenario.

5 CONCLUDING REMARKS AND LIMITATIONS

We introduced AGREE, a framework to tackle adversarial detection in the presence of simultaneous attacks. The proposed method goes beyond current SOTA methods and provides a formal way to aggregate multiple existent (or future) detectors, possibly provided by a third party.

We formalized the simultaneous attacks detection setup as a minimax cross-entropy risk, and we derived a surrogate loss from which we formally characterized our optimal soft-detector leading to AGREE. Overall, we have empirically shown that, in the simultaneous attacks scenario, aggregating simple detectors through AGREE provides better results than using the best SOTA method individually. AGREE presents two key ingredients: it is *modular*, as it allows existing (and possible future) methods to be merged into a single one; and it is *general*, as it can simultaneously recognize adversarial examples created according to various attacks algorithms and loss functions.

While recent work has attempted to bridge the gap between the field of robustness and the field of adversarial detection Tramèr (2021), via a reduction that is computationally inefficient, our work is motivated by the idea of reusing already published methods within a sound aggregation framework.

Moreover, we draw the attention to the fact that AGREE can potentially be extended to aggregate both SOTA supervised and unsupervised adversarial detection methods. Finally, from a theoretical perspective a limitation of AGREE derives from Eq. (10), since it requires to solve one optimization problem with $|\mathcal{K}|$ unknowns for each sample at evaluation time.

AGREE relies on a collection of detectors whose expertise is combined to obtain a more robust adversarial detection. Such models could be potentially poisoned by a malicious actor, drastically reducing AGREE's reliability. We think this constitutes a limitation with potential severe societal impact if AGREE happened to be deployed with no additional checks on the quality of the available detectors.

REFERENCES

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems*, pp. 12214–12223, 2019.

Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Deforges. Adversarial example detection for dnn models: A review and experimental comparison. *Artificial Intelligence Review*, 2022.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture Notes in Computer Science*, 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. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Andrew R. Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Trans. Inf. Theory*, 44(6):2743–2760, 1998.

Oliver Bryniarski, Nabeel Hingun, Pedro Pachuca, Vincent Wang, and Nicholas Carlini. Evading adversarial example detection defenses with orthogonal projected gradient descent. *CoRR*, abs/2106.15023, 2021. URL https://arxiv.org/abs/2106.15023.

Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Bhavani Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha (eds.), *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pp. 3–14. ACM, 2017a.

Nicholas Carlini and David A. Wagner. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *CoRR*, abs/1711.08478, 2017b. URL http://arxiv.org/abs/1711.08478.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57. IEEE Computer Society, 2017c.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pp. 1277–1294. IEEE, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2206–2216. PMLR, 2020.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1802–1811. PMLR, 2019.

Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting adversarial samples from artifacts. *CoRR*, abs/1703.00410, 2017.

Aditya Gangrade, Anil Kag, and Venkatesh Saligrama. Selective classification via one-sided prediction. In Arindam Banerjee and Kenji Fukumizu (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2179–2187. PMLR, 2021.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019.

Eduardo Dadalto Câmara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection. *CoRR*, abs/2203.07798, 2022.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. DOCTOR: A simple method for detecting misclassification errors. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5669–5681, 2021.

Federica Granese, Marine Picot, Marco Romanelli, Francisco Messina, and Pablo Piantanida. MEAD: A multi-armed approach for evaluation of adversarial examples detectors. *CoRR*, abs/2206.15415, 2022. URL https://doi.org/10.48550/arXiv.2206.15415.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Anouar Kherchouche, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. Detection of adversarial examples in deep neural networks with natural scene statistics. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pp. 1–7. IEEE, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 7167–7177, 2018.

Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. MOOD: multi-level out-of-distribution detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 15313–15323. Computer Vision Foundation / IEEE, 2021.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (eds.), *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pp. 135–147. ACM, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2574–2582. IEEE Computer Society, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

M. Picot, F. Messina, M. Boudiaf, F. Labeau, I. Ben Ayed, and P. Piantanida. Adversarial robustness via fisher-rao regularization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022. ISSN 1939-3539.

Alexander Robey, Luiz Chamon, George J Pappas, Hamed Hassani, and Alejandro Ribeiro. Adversarial robustness with semi-infinite constrained learning. *Advances in Neural Information Processing Systems*, 34:6198–6215, 2021.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with Gram matrices. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8491–8501. PMLR, 13–18 Jul 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Florian Tramèr. Detecting adversarial examples is (nearly) as hard as classifying them. *CoRR*, abs/2107.11630, 2021. URL `https://arxiv.org/abs/2107.11630`.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero,

Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV (8)*, pp. 560–574, 2018.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan L. Yuille, and Quoc V. Le. Smooth adversarial training. *CoRR*, abs/2006.14536, 2020.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018.

Chengyuan Yao, Pavol Bielik, Petar Tsankov, and Martin Vechev. Automated discovery of adaptive attacks on adversarial defenses. *Advances in Neural Information Processing Systems*, 34:26858–26870, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 1–11, 2019.

Lily H. Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12427–12436. PMLR, 2021.

Roland S. Zimmermann, Wieland Brendel, Florian Tramèr, and Nicholas Carlini. Increasing confidence in adversarial robustness evaluations. *CoRR*, abs/2206.13991, 2022. doi: 10.48550/arXiv.2206.13991. URL https://doi.org/10.48550/arXiv.2206.13991.

Supplementary Material

A Supplementary Details on Sec. 3

A.1 Proofs

Proof of Eq. (7)

$$\max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[-\log q_{\widehat{Z}|\mathbf{u_0}}\right] = \max_{k\in\mathcal{K}}\left[\mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[-\log q^{(k)}_{\widehat{Z}|\mathbf{u_0}}\right] + \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[\log\left(\frac{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right]\right] \tag{12}$$

$$\leq \max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[-\log q^{(k)}_{\widehat{Z}|\mathbf{u_0}}\right] + \max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[\log\left(\frac{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right]. \tag{13}$$

Proof of Eq. (8) The equality hold by noticing that

$$\max_{P_\Omega} \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right] \leq \max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[\log\left(\frac{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right], \tag{14}$$

and moreover,

$$\max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[\log\left(\frac{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right] = \mathbb{E}_{\bar{\Omega}}\left[D_{\mathrm{KL}}\left(q^{(\bar{\Omega})}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right], \tag{15}$$

by choosing the random variable $\bar{\Omega}$ with uniform probability over the set of maximizers: $\overline{\mathcal{K}} = \arg\max_{k\in\mathcal{K}} \mathbb{E}_{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}\left[\log\left(\frac{q^{(k)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right)\right]$ and zero otherwise. The above inequalities show the equality.

Proof of Eq. (9) We consider a zero-sum game with a concave-convex mapping defined on a product of convex sets. The sets of all probability distributions $q_{\widehat{Z}|\mathbf{u_0}}$ and P_Ω are two nonempty convex sets, bounded and finite dimensional. On the other hand, $\left(P_\Omega, q_{\widehat{Z}|\mathbf{u_0}}\right) \to \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right]$ is a concave-convex mapping, i.e., $P_\Omega \to \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right]$ is concave and $q_{\widehat{Z}|\mathbf{u_0}} \to \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right]$ is convex for every $\left(P_\Omega, q_{\widehat{Z}|\mathbf{u_0}}\right)$. Then, by classical min-max theorem von Neumann (1928) we have that

$$\min_{q_{\widehat{Z}|\mathbf{u_0}}}\max_{P_\Omega} \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right] = \max_{P_\Omega}\min_{q_{\widehat{Z}|\mathbf{u_0}}} \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right]. \tag{16}$$

Proof of Eq. (10) It is enough to show that

$$\min_{\widehat{q}_{\widehat{Z}|\mathbf{u_0}}} \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right] = I_{\mathbf{u_0}}(\Omega;\widehat{Z}), \tag{17}$$

for every random variable Ω distributed according to an arbitrary probability distribution P_Ω and each distribution $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$. We begin by showing that

$$\mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right] \geq I_{\mathbf{u_0}}(\Omega;\widehat{Z}), \tag{18}$$

for any arbitrary distributions P_Ω and $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$. To this end, we use the following identities:

$$\mathbb{E}_\Omega\left[D_{\text{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}\middle\|q_{\widehat{Z}|\mathbf{u_0}}\right)\right] = \mathbb{E}_\Omega\mathbb{E}_{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}\left(\log\frac{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right) \tag{19}$$

$$= \mathbb{E}_\Omega\mathbb{E}_{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}\left(\log\frac{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}{P_{\widehat{Z}}}\right) + D_{\text{KL}}\left(P_{\widehat{Z}}\|q_{\widehat{Z}|\mathbf{u_0}}\right) \tag{20}$$

$$= I_{\mathbf{u_0}}(\Omega;\widehat{Z}) + D_{\text{KL}}\left(P_{\widehat{Z}}\|q_{\widehat{Z}|\mathbf{u_0}}\right) \tag{21}$$

$$\geq I_{\mathbf{u_0}}(\Omega;\widehat{Z}), \tag{22}$$

where $P_{\widehat{Z}}$ denotes the marginal distribution of $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$ w.r.t. P_Ω and the last inequality follows since the KL divergence is positive. Finally, it is easy to check that by selecting $q_{\widehat{Z}|\mathbf{u_0}} = P_{\widehat{Z}}$ the lower bound in equation 22 is achieved which proves the identity in expression equation 17. By taking the maximum over all probability distributions P_Ω at both sides of expression equation 17 the claim follows.

A.2 On the optimization of Eq. (10)

The maximization problem in Eq. (10) is well-posed given that the mutual information is a concave function of $\omega \in \Omega$. Although, from the theoretical point of view, Eq. (10) guarantees the optimal solution for the average regret minimization problem, in practice, we have to deal with some technical limitations. For the optimization of Eq. (10), we rely on the `SciPy` Virtanen et al. (2020) library, package `optimize`, function `minimize`[1] which uses the *Sequential Least Squares Programming* (SLSQP) algorithm to find the optimum. This algorithm relies on local optimization, and is particularly straightforward when dealing with non-linear equations and both equality and inequality constraints, as in our case. Overall, we obtained the satisfactory results provided in the paper by assigning default values to all the parameters, and by setting a uniform distribution $[\omega_1, \omega_2, \omega_3, \omega_4] = [.25, .25, .25, .25]$ as initial point in the space of the solutions.

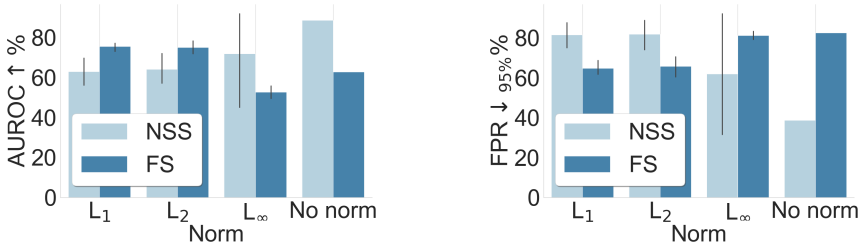
Although these results are satisfactory and confirm the mathematical intuition behind our proposed framework, we are aware that, in some cases, as in Fig. 1a, the aggregation of AGREE, slightly underperforms in terms of accuracy w.r.t. the best detector in the set of allowed detectors. In this regard we would like to raise a couple of points which are interesting for practitioners and possible future research:

1. For each input sample we solve one different optimization problem: although the algorithm above always reaches the end with a success state, given the finite amount of iterations and the tolerance which decides the stopping criterion, further sample-by-sample parameter optimization may be required. At this time we have not delved into the problem, and we leave this for future research.
2. The hard decisions made by the single detectors only depend on the $\arg\max$ of their soft-probabilities. On the contrary, the optimization in Eq. (10) considers the complete soft-probability distributions output by each single detector. Indeed, although the hard decision on two randomly considered samples can be right for both, often the confidence on these decisions can be very different (i.e. two correctly classified samples may have utterly different associated soft probabilities). Further research on how differently accurate detectors influence the optimization in Eq. (10) is left for future work.

B Supplementary Results of Section 4

In the following, we provide further discussions on the experiments in Sec. 4 that for space constraints have not been included in the main paper.

[1] Therefore we invert the sign of the objective function.



(a) Analysis AUROC↑%

(b) Analysis FPR$\downarrow_{95\%}$%

Figure 2: SOTA's performances under MEAD grouped by norm. The plots reflects the results of Tabs. 5-8 in (Granese et al., 2022). We focus on the best supervised method (i.e., NSS (Kherchouche et al., 2020)) and the best unsupervised method (i.e., FS (Xu et al., 2018)).

B.1 EXPERIMENTAL ENVIRONMENT

We run each experiment on a machine equipped with an Intel(R) Xeon(R) Gold 6226 CPU, 2.70GHz clock frequency, and a Tesla V100-SXM2-32GB GPU.

B.1.1 TIME MEASUREMENTS

Training 1 single detector in AGREE	`1h45m10s`
Evaluating AGREE optimization	`1m35s` (for one attack)
Training NSS	`3m30s`
Evaluating NSS	`20s` (for one attack)
On the largest set of simultaneous attacks (13 attacks):	
AGREE	`1m35s * 13 ~ 21m`
NSS	`20s * 13 ~ 4m`

B.2 ON THE MEAD FRAMEWORK

B.2.1 STATE-OF-THE-ART (SOTA) DETECTORS

Granese et al. (2022) suggests NSS (Kherchouche et al., 2020) and FS (Xu et al., 2018) as the most robust methods in the simultaneous attacks detection scheme (i.e., MEAD). We remind that NSS is a supervised method which extracts the *natural scene statistics* of the natural and adversarial examples to train a SVM. On the contrary, FS is an unsupervised method that uses *feature squeezing* to compare the model's predictions.

In particular, we choose NSS as method to compare with for multiple reasons:

1. NSS achieves the best overall score in terms of AUROC↑% and FPR$\downarrow_{95\%}$% among the SOTA against simultaneous attacks (c.f. Tab. 3 (Granese et al., 2022)).
2. NSS achieves the best score in terms of AUROC↑% and FPR$\downarrow_{95\%}$% under the L_∞ norm where the biggest group of simultaneous attack are evaluated (see Tab. 1). This is stressed in the plots in Fig. 2. Moreover, FS turns out to reach better performance w.r.t. AGREE only with PGD1 and PGD2 when the perturbation magnitude is small and in CW2.
3. The case study for AGREE in the experimental section is based on supervised detectors as consequence the comparison with a supervised detector was a natural choice.

For the sake of completeness, the performances of NSS and FS under MEAD are given in Fig. 2.

B.2.2 ATTACKS

We believe that it is important to stress that, differently from literature, we are the first to consider a defence mechanism against the simultaneous attack setting in which we detect attacks based on four different losses. More specifically, for each 'clean dataset' (in our case CIFAR10 and SVHN):

- No. of adversarial examples generated with:
  - L_1 norm: 7 (no. of ε) * 1 (PGD algorithm) * 4 (no. of losses) = 28 ('adversarial datasets')
  - L_2 norm: 7 (no. of ε) * 1 (PGD algorithm) * 4 (no. of losses) + 3 (CW2, HOP, DeepFool) = 31 ('adversarial datasets')
  - L_∞ norm: 6 (no. of ε) * 3 (PGD, FGSM, BIM algorithms) * 4 (no. of losses) + 2 = 74 ('adversarial datasets')
  - No norm: 1 ('adversarial dataset')

$=>$ For a total of `28 + 31 + 74 + 1 =` **`134`** 'adversarial datasets' for each 'clean dataset'.

Moreover, it is interesting to notice that the experiments on CIFAR10 and SVHN represent a satisfying choice to show that state-of-the-art detection mechanisms struggle to maintain good performance when they are faced with the framework of simultaneous attacks. That said, we leave the evaluation of larger datasets as future work.

B.3 SIMULATIONS ACCORDING TO DIFFERENT ε

As previously discussed in Sec. 4, both NSS and AGREE are trained on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. We show in Tabs. 4 to 7 the results of the two methods according to $\varepsilon \in \{.03125, .0625, .125, .25, .3125, .5\}$.

Table 4: Simultaneous attacks detection: NSS on CIFAR10. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	NSS											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**48.5**	**94.2**	47.7	94.7	46.6	95.6	46.8	95.5	47.0	95.4	46.5	95.6
$\varepsilon = 10$	**54.0**	**90.3**	53.4	90.8	51.6	94.3	50.4	94.9	50.4	94.9	50.9	94.7
$\varepsilon = 15$	**58.8**	**86.8**	58.1	87.4	55.8	92.8	53.8	94.2	53.2	94.4	54.5	93.7
$\varepsilon = 20$	**63.5**	**82.3**	62.7	82.7	60.1	90.7	57.4	93.2	56.7	93.6	58.2	92.3
$\varepsilon = 25$	**67.7**	**77.2**	66.8	78.4	64.0	87.8	61.0	92.0	60.1	92.6	61.9	90.6
$\varepsilon = 30$	**71.4**	**73.4**	70.5	73.5	67.6	83.7	64.4	90.4	63.4	91.4	65.4	88.2
$\varepsilon = 40$	**76.1**	**67.3**	75.3	68.0	72.6	75.4	69.4	87.2	68.5	88.9	70.4	83.4
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**48.3**	**94.3**	47.5	94.8	46.6	95.6	46.7	95.5	47.1	95.4	46.5	95.6
$\varepsilon = 0.25$	**53.2**	**91.2**	52.6	91.6	50.9	94.6	50.0	95.0	50.0	95.0	50.3	94.8
$\varepsilon = 0.3125$	**55.8**	**89.2**	55.2	89.9	53.3	93.7	51.7	94.6	51.5	94.7	52.3	94.3
$\varepsilon = 0.5$	**63.3**	**82.6**	62.6	83.0	60.0	90.7	57.4	93.2	56.7	93.5	58.2	92.4
$\varepsilon = 1$	**76.4**	**67.5**	75.7	67.8	73.1	75.0	70.1	86.7	69.2	88.5	71.0	83.0
$\varepsilon = 1.5$	**81.0**	63.0	80.5	**62.7**	78.5	63.5	76.2	80.7	75.6	83.2	76.9	74.4
$\varepsilon = 2$	**82.6**	62.3	82.1	**61.6**	80.6	62.5	78.6	78.5	78.1	81.2	79.1	72.1
DeepFool												
No ε	**57.0**	**91.7**	56.7	**91.7**	55.6	93.6	54.6	94.1	54.2	94.3	54.7	94.0
CW2												
$\varepsilon = 0.01$	**56.4**	**90.8**	55.9	90.9	54.5	93.7	53.4	94.3	53.0	94.5	53.6	94.1
HOP												
$\varepsilon = 0.1$	**66.1**	**87.0**	65.1	88.2	63.0	91.3	61.2	92.6	60.8	92.9	61.6	92.1
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	**83.0**	55.3	82.1	**55.2**	80.3	57.8	77.4	77.0	76.8	81.3	78.3	65.4
$\varepsilon = 0.0625$	**96.0**	**17.2**	94.6	17.4	94.9	19.2	94.3	21.6	94.4	21.1	94.4	21.1
$\varepsilon = 0.25$	**97.3**	**0.6**	94.7	5.9	96.5	2.5	96.9	1.7	97.2	1.1	96.7	2.1
$\varepsilon = 0.5$	**82.5**	**100.0**	80.4	**100.0**	81.9	**100.0**	82.2	**100.0**	82.4	**100.0**	82.0	**100.0**
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	9.4	**99.9**	10.4	100.0	26.2	**99.9**	30.9	100.0	**33.8**	100.0	27.3	100.0
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	**63.2**	99.1	62.7	**99.0**	61.9	99.3	60.9	99.5	60.5	99.5	61.2	99.4
No norm												
STA												
No ε	88.5	38.8	92.0	25.1	92.1	22.4	**93.3**	**18.3**	92.7	19.6	92.7	19.7

Table 5: Simultaneous attacks detection: AGREE on CIFAR10. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

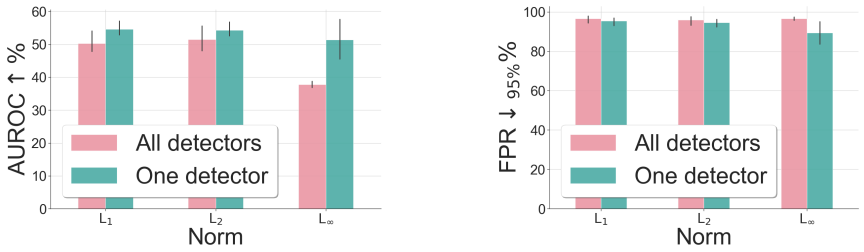
	AGREE											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**69.7**	82.5	65.5	**81.5**	62.1	87.1	56.3	93.8	53.2	94.8	48.5	95.5
$\varepsilon = 10$	62.3	**83.3**	**62.7**	86.3	56.8	90.6	52.1	94.7	52.9	94.6	50.9	95.0
$\varepsilon = 15$	66.6	**72.7**	**73.9**	77.9	69.3	84.4	65.5	89.0	64.3	91.0	60.4	93.1
$\varepsilon = 20$	72.8	**58.0**	**83.7**	59.3	78.7	73.1	73.8	82.5	73.5	85.4	69.2	90.3
$\varepsilon = 25$	76.8	42.4	**89.4**	**35.9**	87.1	50.8	81.3	68.6	79.3	78.0	74.8	87.2
$\varepsilon = 30$	79.1	31.1	**91.7**	**21.4**	90.3	35.4	84.3	61.2	81.9	73.5	77.5	85.3
$\varepsilon = 40$	80.8	22.2	**93.0**	**15.0**	92.1	26.4	85.9	56.8	83.1	71.4	78.8	84.5
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**71.3**	80.8	67.0	**80.2**	63.9	85.4	56.2	93.8	53.8	94.7	48.6	95.5
$\varepsilon = 0.25$	**63.1**	**83.4**	62.8	86.7	57.1	90.5	52.3	94.6	52.6	94.7	49.9	95.2
$\varepsilon = 0.3125$	64.1	**79.3**	**67.3**	83.1	61.0	88.9	58.0	92.8	57.7	93.3	54.5	94.4
$\varepsilon = 0.5$	72.9	**58.9**	**83.7**	60.7	79.4	73.2	74.6	81.4	73.4	85.4	68.8	90.5
$\varepsilon = 1$	81.0	21.7	**92.9**	**15.5**	91.4	26.4	85.5	57.2	82.9	72.2	78.7	84.7
$\varepsilon = 1.5$	81.5	19.2	**93.2**	**14.2**	91.9	24.2	85.9	56.3	83.2	71.9	79.2	84.4
$\varepsilon = 2$	81.6	19.0	**93.2**	**14.1**	91.9	24.1	85.9	56.3	83.3	71.8	79.2	84.4
DeepFool												
No ε	**91.1**	**22.0**	87.4	33.9	81.9	54.8	70.0	84.4	64.2	91.5	56.3	94.4
CW2												
$\varepsilon = 0.01$	52.9	**90.5**	50.7	90.6	**53.4**	92.2	53.1	94.4	52.0	94.8	50.9	95.0
HOP												
$\varepsilon = 0.1$	**91.3**	**20.9**	89.0	31.0	86.1	49.1	77.0	80.7	72.4	88.1	64.3	92.8
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	67.2	77.3	77.8	65.2	**82.3**	**59.7**	78.0	72.1	73.7	83.8	64.1	92.2
$\varepsilon = 0.0625$	69.0	83.6	85.3	47.4	**92.0**	**29.6**	90.7	35.7	88.0	45.6	81.3	78.3
$\varepsilon = 0.25$	72.0	67.4	91.8	23.2	**95.9**	**8.8**	94.1	15.4	92.6	19.5	91.6	26.5
$\varepsilon = 0.5$	58.3	84.8	84.2	44.1	**94.6**	**9.7**	91.2	16.5	90.5	18.8	91.3	22.3
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	69.0	79.1	84.1	41.9	**88.9**	**40.8**	86.6	52.3	85.4	60.4	80.7	79.0
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	66.6	75.0	**80.6**	**51.5**	80.0	61.1	72.0	84.0	67.2	90.0	60.0	93.6
No norm												
STA												
No ε	84.8	**33.8**	**85.0**	41.5	82.7	52.4	72.9	77.7	70.2	81.7	63.1	92.1

Table 6: Simultaneous attacks detection: NSS on SVHN. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	NSS											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	37.9	89.3	**40.2**	91.3	37.2	89.2	4.9	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 10$	33.7	89.3	**36.9**	91.3	34.6	89.2	6.0	35.5	0.4	8.5	0.0	**3.1**
$\varepsilon = 15$	31.9	89.3	**35.6**	91.3	34.4	89.2	7.6	35.5	0.5	8.5	0.1	**3.1**
$\varepsilon = 20$	31.5	89.3	**36.1**	91.3	35.7	89.2	9.5	35.5	0.6	8.5	0.1	**3.1**
$\varepsilon = 25$	32.8	89.3	37.8	91.3	**38.2**	89.2	11.7	35.5	0.9	8.5	0.1	**3.1**
$\varepsilon = 30$	34.5	89.3	39.8	91.3	**40.6**	89.2	14.1	35.5	1.2	8.5	0.1	**3.1**
$\varepsilon = 40$	37.9	89.3	43.1	91.3	**43.4**	89.0	16.4	35.5	2.2	8.5	0.3	**3.1**
Norm L_2												
PGD2												
$\varepsilon = 0.125$	38.7	89.3	**40.8**	91.3	37.6	89.2	4.7	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 0.25$	34.0	89.3	**37.2**	91.3	34.6	89.2	5.4	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 0.3125$	32.6	89.3	**36.1**	91.3	34.1	89.2	6.1	35.5	0.4	8.5	0.0	**3.1**
$\varepsilon = 0.5$	31.4	89.3	**35.9**	91.3	35.4	89.2	8.9	35.5	0.5	8.5	0.1	**3.1**
$\varepsilon = 1$	37.4	89.3	42.5	91.3	**42.9**	89.2	16.0	35.5	2.1	8.5	0.3	**3.1**
$\varepsilon = 1.5$	40.0	89.3	46.3	91.3	**46.5**	88.4	17.2	35.5	2.8	8.5	0.6	**3.1**
$\varepsilon = 2$	42.1	89.3	49.8	91.3	**50.5**	88.0	18.7	35.5	3.2	8.5	0.8	**3.1**
DeepFool												
No ε	38.1	89.3	**41.3**	91.3	39.7	89.2	9.2	35.5	0.8	8.5	0.1	**3.1**
CW2												
$\varepsilon = 0.01$	37.9	89.3	**41.0**	91.3	39.5	89.2	9.3	35.5	0.8	8.5	0.1	**3.1**
HOP												
$\varepsilon = 0.1$	66.8	82.3	**67.6**	84.2	60.3	84.6	16.4	35.5	2.7	8.5	0.7	**3.1**
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	84.1	49.7	**86.3**	46.9	77.5	72.1	22.2	33.2	4.3	8.5	1.2	**3.1**
$\varepsilon = 0.0625$	87.4	**0.2**	**88.9**	0.7	87.5	0.6	33.7	16.8	7.4	6.8	2.5	2.7
$\varepsilon = 0.25$	16.7	89.3	51.6	88.9	**52.0**	85.1	35.4	**0.1**	8.4	**0.1**	3.0	**0.1**
$\varepsilon = 0.5$	4.1	89.3	**46.7**	86.7	46.0	84.6	35.4	**0.1**	8.4	**0.1**	3.0	**0.1**
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	22.8	89.3	32.9	91.3	**43.6**	89.2	30.3	32.7	7.1	8.5	2.5	**3.1**
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	4.7	89.3	**41.3**	91.3	40.8	89.2	12.7	35.5	1.7	8.5	0.4	**3.1**
No norm												
STA												
No ε	89.3	**0.0**	**91.2**	0.2	85.9	23.4	19.9	33.5	4.2	8.3	1.4	3.1

Table 7: Simultaneous attacks detection: AGREE on SVHN. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	AGREE											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**79.3**	**65.2**	77.4	73.4	76.9	78.9	76.9	79.0	76.7	79.5	74.0	84.4
$\varepsilon = 10$	**74.4**	**65.1**	72.8	73.1	71.9	81.6	73.0	82.5	71.9	84.2	66.9	89.4
$\varepsilon = 15$	76.0	**57.0**	75.7	64.6	75.8	73.1	**78.9**	72.5	77.3	74.7	71.9	84.9
$\varepsilon = 20$	77.3	**48.1**	77.9	54.9	79.2	61.9	**83.6**	60.7	82.2	64.3	77.4	76.9
$\varepsilon = 25$	78.2	**40.9**	79.4	44.4	81.4	49.4	**87.0**	48.6	85.7	52.5	81.4	66.7
$\varepsilon = 30$	78.8	**34.4**	80.4	35.3	83.0	36.6	**89.3**	37.2	88.1	41.6	84.4	53.8
$\varepsilon = 40$	79.7	23.4	81.6	22.4	84.7	20.2	**92.6**	**20.0**	91.1	23.0	87.8	30.5
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**82.2**	**61.7**	80.6	68.4	80.3	72.4	80.2	74.5	80.1	73.5	79.7	75.5
$\varepsilon = 0.25$	**75.7**	**63.6**	74.0	71.7	73.3	80.3	74.0	81.7	72.6	82.8	67.8	89.0
$\varepsilon = 0.3125$	**75.5**	**61.6**	74.3	70.1	73.9	78.4	75.2	79.4	73.9	81.7	70.6	86.7
$\varepsilon = 0.5$	77.2	**50.6**	77.6	57.4	78.6	64.1	**82.5**	64.4	81.2	67.1	76.3	79.5
$\varepsilon = 1$	79.5	25.8	81.3	24.8	84.3	**24.1**	**92.3**	24.7	90.7	27.7	87.1	36.4
$\varepsilon = 1.5$	80.2	19.5	82.2	17.6	85.6	14.3	**94.1**	**7.5**	92.9	8.6	89.9	11.8
$\varepsilon = 2$	80.5	19.4	82.5	17.5	85.9	14.1	**94.9**	**5.3**	94.5	6.8	90.7	9.5
DeepFool												
No ε	**96.3**	**8.6**	95.9	10.5	95.0	12.9	94.9	12.0	95.3	12.1	95.5	12.6
CW2												
$\varepsilon = 0.01$	**59.7**	**76.3**	57.2	80.1	53.4	89.9	54.2	92.0	51.1	93.5	44.3	96.1
HOP												
$\varepsilon = 0.1$	**96.1**	**7.9**	95.6	9.8	95.9	11.7	96.0	10.2	95.9	9.9	**96.1**	10.0
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	74.3	**60.0**	75.8	60.3	77.8	62.6	**81.4**	64.9	80.1	67.1	76.7	75.5
$\varepsilon = 0.0625$	78.4	36.0	80.3	34.1	83.2	33.8	89.1	**33.3**	**87.9**	34.4	85.7	37.4
$\varepsilon = 0.25$	80.1	19.4	82.1	17.5	85.2	**15.8**	**92.3**	16.4	92.1	16.8	89.6	17.0
$\varepsilon = 0.5$	80.3	19.4	82.3	17.5	85.5	**14.1**	**92.9**	14.4	91.7	15.2	90.1	14.8
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	78.9	29.0	80.8	**28.1**	83.8	28.7	**89.2**	29.1	88.4	28.9	86.8	28.4
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	78.7	33.4	80.5	31.9	83.1	34.0	**88.2**	33.1	88.1	31.7	86.7	**31.2**
No norm												
STA												
No ε	**94.7**	**14.5**	93.3	16.8	89.9	23.1	90.2	23.2	91.0	22.4	91.1	22.4



(a) Analysis AUROC↑% (b) Analysis FPR↓$_{95\%}$%

Figure 3: AGREE against the adaptive-attacks under MEAD. We consider the worst case scenario in Tab. 11, i.e., when $\alpha = 0.1$.

B.3.1 AGREE AGAINST THE ADAPTIVE-ATTACKS IN THE MEAD SCENARIO

We present a new experimental setting to address the case in which also the detectors are attacked at the same time as the target classifier, taking the cue from Bryniarski et al. (2021); Carlini & Wagner (2017a); Tramèr et al. (2020); Carlini & Wagner (2017b). It is important to note that, in the spirit of the MEAD framework, we are not simply considering a scenario in which a *single* adaptive attack is perpetrated on the classifier and detectors, but rather multiple adaptive attacks are concurrently occurring. Up to our knowledge, this scenario has not yet been considered in Granese et al. (2022) and hence we are the first to deal with such a setting. We extend AGREE to include two main cases: *(i)* for attacks on the classifier and the single detectors individually; *(ii)* for attacks on the classifier and all the detectors simultaneously.

The tables with the complete results are Tabs. 11 and 12 in Appendix C.2, where α is the coefficient that controls the gradient's speed of the attack against the detectors. We try many different values $\alpha = \{.1, 1, 5, 10\}$. The case where α is equal to 0 is added for completeness and it corresponds to the case where only the target classifier is attacked. We report in Fig. 3 the comparison of the results between case *(i)* and case *(ii)* on CIFAR10 and $\alpha = 0.1$, as this corresponds to the case with the worst performances. As can be seen, the performances of AGREE improve when the detectors are attacked singularly. This is particularly interesting for the setting we are dealing with. Indeed, AGREE is not a new supervised adversarial detection method, but a framework to aggregate detectors, in this case applied to the adversarial detection problem. Hence, it does not propose to solve the problem of finding a new robust method to adaptive-attacks but rather creating a mixture of experts based on the proposed sound mathematical framework. Thus, an attacker in order to successfully fool AGREE needs to have the *complete access to all* the underlying detectors and also *an up to the date knowledge of the detectors employed* as the defender can always includes a new detection mechanism to the pool of the detectors.

In order to give more insights on AGREE under this setting, we train a *stronger* version of the four shallow detectors where the detectors at training time have seen the corresponding adaptive attacks generated through the PGD algorithm. We report the results in Tab. 8 where we focus on the group of simultaneous attacks with L_∞ norm and $\varepsilon = 0.25$ as this represents the worst result of AGREE in Tab. 12. If AGREE was only good as the best among the detectors, we should expect similar results in Tab. 8. In this case, the only solution would be to train a better detector. **However, the strength of AGREE is not just mimicking the performance of its parts but rather creating a mixture of experts based on the proposed sound mathematical framework.** Therefore we should expect better performances. Indeed, this consistently holds as AGREE performs much better than the best detector.

Table 8: Comparison between AGREE the single detectors (*stronger* version) against the adaptive-attacks. Norm L_∞ and $\varepsilon = 0.25$ (i.e., attacks PGDi*, FGSM*, BIM*).

CIFAR10	AGREE	ACE	KL	FR	Gini
AUROC↑%	**54.6**	35.7	30.6	26.3	36.2
FPR↓$_{95\%}$%	**73.0**	96.5	97.0	97.4	99.6

B.3.2 AGREE IN THE THE NON-SIMULTANEOUS SETTING

In these experiments, we move from the simultaneous attack scenario to one where the different detectors are aggregated in order to detect one single attack at a time, as usually done in the literature. We report the complete results in Appendix C.1, Tab. 10. Crucially, these experiments show that the ensemble detectors can also improve the performance for a specific attack. In particular, we would like to draw the attention on the fact that we outperform NSS in the vast majority of the cases. Moreover, we achieve a maximum gain of 82.8 percentage points in terms of AUROC↑% (c.f. SA attack) and 97.6 percentage points in terms of FPR$\downarrow_{95\%}$% (c.f. FGSM with $\varepsilon = 0.5$ attack). On the other side, the competitor outperforms our proposed method only in a few cases, achieving a maximum gain of 5.9 percentage points in terms of AUROC↑% and 27.4 percentage points in terms of FPR$\downarrow_{95\%}$% (c.f. FGSM with ε=0.03125 attack in both the cases), and these gains are much lower than those obtained through AGREE.

B.4 AUTOATTACK

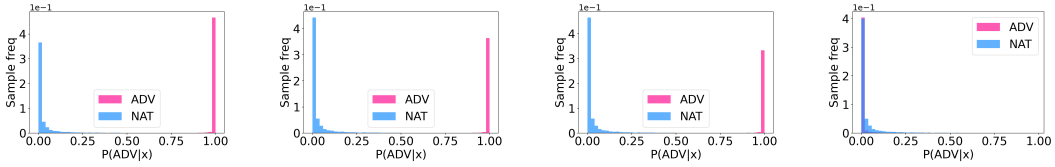
We present an application of AutoAttack Croce & Hein (2020), a state-of-the-art evaluation tool for robustness, redesigned for adversarial detection evaluation and adapted to our simultaneous attacks framework. In its original version, AutoAttack evaluates the accuracy of robust classifiers. In so doing, Croce & Hein (2020) proposes a multiple attacks framework to make sure that at least one attack succeeds in producing an adversarial example for each natural one. In their context, it does not matter which attack will succeed since any successful attack would undermine the accuracy of the target classifier in the same way. In our case the number of different successful attacks for each natural sample will affect the quality of the detection since a detector is successful only if it can detect all of them. Because of the differences underlined above, it is not possible to deploy it directly in our framework without any modifications. A modified version of AutoAttack, adapted to the evaluation of our proposed method has been implemented and the results are presented below. While AutoAttack suggests to use different attack strategies, in our case we combine different attack strategies matched with different losses, in order to make the pool of attacks more strong and diversified.

As future work, it would be interesting to use the work in Zimmermann et al. (2022) to asses the strength of the various attacks strategies before evaluating defenses on them.

Table 9: AGREE on AutoAttack (MEAD setting). The attacks are APGD-CE, APGD-DLR, FAB, SA.

	CIFAR10	
	AGREE	
	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_1		
$\varepsilon = 5$	57.1	88.4
$\varepsilon = 10$	67.1	75.7
$\varepsilon = 15$	72.2	66.7
$\varepsilon = 20$	72.7	65.2
$\varepsilon = 25$	72.8	65.6
$\varepsilon = 30$	73.4	64.0
$\varepsilon = 40$	73.6	64.0
Norm L_2		
$\varepsilon = 0.125$	67.4	81.0
$\varepsilon = 0.25$	58.0	89.0
$\varepsilon = 0.3125$	58.1	88.8
$\varepsilon = 0.5$	69.4	74.7
$\varepsilon = 1$	75.1	61.6
$\varepsilon = 1.5$	76.1	60.7
$\varepsilon = 2$	76.1	60.5
Norm L_∞		
$\varepsilon = 0.03125$	75.7	61.0
$\varepsilon = 0.0625$	76.0	60.7
$\varepsilon = 0.125$	76.8	60.3
$\varepsilon = 0.25$	76.8	60.0
$\varepsilon = 0.3125$	78.6	57.6
$\varepsilon = 0.5$	76.1	60.3

B.5 ADDITIONAL PLOTS



(a) PGD-L_1-40-ACE (b) PGD-L_1-40-KL (c) PGD-L_1-40-FR (d) PGD-L_1-40-Gini

Figure 4: In pink the results for the adversarial examples and in blue the ones for the naturals. In this simulation, we consider a subset of the available detectors (ACE, KL, FR). Under each plot, we indicate the tested attack configuration parameters: algorithm-L_p-ε-loss.

The specific shape in the histograms depends on the set of considered detectors. In order to shed a light on this fact, we include the plots in Fig. 4 in which we consider a subset of the available detectors (ACE, KL, FR). These plots should be compared with the ones in Fig. 1.

C ADDITIONAL RESULTS

C.1 AGREE UNDER THE NON-SIMULTANEOUS SETTING: TABLE OF THE RESULTS

Table 10: AGREE and NSS in the non-simultaneous setting. The column names ACE, KL, FR and Gini denote the loss function used to craft the attacks. Note that, HOP, DeepFool, CW2 and STA attacks have already been considered individually in Tab. 2.

	CIFAR10			
	AGREE AUROC↑% (FPR$\downarrow_{95\%}$%) – NSS AUROC↑% (FPR$\downarrow_{95\%}$%)			
	ACE	KL	FR	Gini
PGD1				
$\varepsilon = 5$	**66.2 (83.6)** – 49.9 (93.5)	**64.2 (85.7)** – 49.6 (93.0)	**63.0 (87.1)** – 49.9 (93.3)	**80.7 (58.4)** – 50.3 (93.2)
$\varepsilon = 10$	**62.6 (87.5)** – 56.9 (88.4)	**62.3 (88.2)** – 56.6 (88.3)	**63.1 (86.5)** – 57.0 (88.1)	**86.9 (46.0)** – 57.1 (88.8)
$\varepsilon = 15$	**74.2 (81.4)** – 63.1 (83.0)	**75.2 (80.6)** – 62.8 (83.1)	**75.3 (79.4)** – 63.2 (82.5)	**90.0 (31.1)** – 63.5 (84.0)
$\varepsilon = 20$	**86.8 (65.3)** – 68.5 (77.1)	**87.5 (63.1)** – 68.1 (77.3)	**86.9 (63.3)** – 68.7 (76.4)	**91.7 (31.2)** – 69.9 (77.6)
$\varepsilon = 25$	**93.9 (38.4)** – 73.1 (71.1)	**94.3 (36.2)** – 72.7 (71.8)	**93.7 (41.1)** – 73.4 (70.9)	**92.3 (28.9)** – 75.0 (71.4)
$\varepsilon = 30$	**97.1 (12.3)** – 77.1 (64.5)	**97.2 (12.6)** – 76.8 (65.1)	**96.8 (15.9)** – 77.4 (65.2)	**92.6 (27.9)** – 78.6 (67.3)
$\varepsilon = 40$	**98.9 (1.0)** – 83.5 (52.7)	**99.0 (1.0)** – 83.3 (53.5)	**98.8 (1.0)** – 83.6 (52.7)	**92.7 (27.4)** – 80.1 (64.9)
PGD2				
$\varepsilon = .125$	**67.9 (81.1)** – 49.5 (93.8)	**65.4 (84.3)** – 49.1 (93.5)	**63.9 (86.6)** – 49.6 (93.5)	**80.6 (58.4)** – 49.5 (94.3)
$\varepsilon = .25$	**62.3 (87.5)** – 55.9 (89.1)	**62.1 (88.0)** – 55.6 (89.2)	**62.6 (87.6)** – 55.8 (89.4)	**86.7 (46.5)** – 55.9 (89.8)
$\varepsilon = .3125$	**66.5 (86.1)** – 59.4 (86.5)	**67.0 (85.9)** – 59.0 (86.6)	**67.8 (84.8)** – 59.3 (86.6)	**88.4 (42.2)** – 59.3 (87.7)
$\varepsilon = .5$	**86.4 (67.1)** – 68.3 (77.4)	**87.2 (64.5)** – 68.0 (77.4)	**86.7 (64.0)** – 68.4 (77.2)	**91.4 (31.4)** – 69.0 (78.7)
$\varepsilon = 1$	**98.9 (0.9)** – 84.4 (50.6)	**98.9 (0.9)** – 84.3 (50.5)	**98.8 (0.9)** – 84.7 (50.7)	**92.5 (27.2)** – 79.3 (66.8)
$\varepsilon = 1.5$	**99.2 (0.9)** – 92.8 (28.7)	**99.3 (0.9)** – 92.7 (28.9)	**99.3 (0.7)** – 93.0 (27.3)	**92.5 (27.2)** – 79.5 (66.5)
$\varepsilon = 2$	**99.3 (0.8)** – 96.8 (13.9)	**99.3 (0.8)** – 96.9 (13.1)	**99.3 (0.9)** – 95.9 (17.2)	**92.5 (27.2)** – 79.5 (66.5)
PGDi				
$\varepsilon = .03125$	**99.1 (0.9)** – 92.3 (31.0)	**99.1 (0.9)** – 92.1 (31.9)	**99.0 (0.9)** – 92.2 (30.7)	**94.8 (21.5)** – 89.0 (44.0)
$\varepsilon = .0625$	**99.3 (0.8)** – 99.1 (3.3)	**99.3 (0.8)** – 99.1 (3.3)	**99.3 (0.8)** – 99.1 (3.6)	97.4 **(8.0)** – **98.1** (8.1)
$\varepsilon = .125$	99.3 (0.7) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.6 (0.6)**	97.3 (7.3) – **99.6 (0.6)**
$\varepsilon = .25$	99.3 (0.7) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.3) – **99.6 (0.6)**
$\varepsilon = .3125$	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.4) – **99.7 (0.6)**
$\varepsilon = .5$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.3) – **99.6 (0.6)**
FGSM				
$\varepsilon = .03125$	89.2 (47.5) – **94.1 (26.7)**	91.3 (40.6) – **94.0 (27.0)**	92.6 (34.1) – **96.8 (15.0)**	90.7 (42.7) – **96.6 (15.3)**
$\varepsilon = .0625$	96.4 (18.5) – **99.4 (1.3)**	96.2 (18.7) – **99.4 (1.4)**	97.6 (10.3) – **99.6 (0.6)**	97.4 (11.9) – **99.6 (0.6)**
$\varepsilon = .125$	99.3 (3.4) – **99.7 (0.6)**	99.1 (4.3) – **99.7 (0.6)**	99.3 (2.5) – **99.5 (0.6)**	99.3 (2.4) – **99.5 (0.6)**
$\varepsilon = .25$	**99.8 (0.6)** – 99.7 **(0.6)**	**99.7** (0.8) – **99.7 (0.6)**	**99.6** (1.1) – 97.9 **(0.6)**	**99.6** (1.1) – 97.7 **(0.6)**
$\varepsilon = .3125$	**99.7** (0.9) – **99.7 (0.6)**	**99.7** (0.9) – **99.7 (0.6)**	**99.5** (1.5) – 95.8 **(0.6)**	**99.5** (1.5) – 95.6 **(0.6)**
$\varepsilon = .5$	99.0 (4.9) – **99.7 (0.6)**	99.2 (2.7) – **99.7 (0.6)**	**99.2 (2.4)** – 84.9 (100.0)	**99.2 (2.4)** – 84.8 (100.0)
BIM				
$\varepsilon = .03125$	**98.3 (4.6)** – 90.3 (37.7)	**98.3 (4.4)** – 90.2 (38.1)	**97.8 (7.2)** – 90.5 (37.0)	**92.2 (32.6)** – 88.2 (45.1)
$\varepsilon = .0625$	**99.4 (0.8)** – 98.2 (7.5)	**99.4 (0.9)** – 98.2 (7.5)	**99.4 (0.8)** – 98.3 (7.3)	96.6 (13.1) – **97.3 (12.9)**
$\varepsilon = .125$	99.3 (0.9) – **99.6 (0.7)**	99.3 (0.9) – **99.7 (0.7)**	99.3 (0.8) – **99.6 (0.7)**	97.8 (6.9) – **99.3 (1.9)**
$\varepsilon = .25$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.4 (7.2) – **99.6 (0.6)**
$\varepsilon = .3125$	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	97.1 (7.4) – **99.7 (0.6)**
$\varepsilon = .5$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	96.3 (7.3) – **99.7 (0.6)**
SA				
$\varepsilon = .125$	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)
CWi				
$\varepsilon = .3125$	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)

C.2 AGREE AGAINST THE ADAPTIVE-ATTACKS UNDER MEAD: TABLE OF THE RESULTS

Table 11: AGREE against the adaptive-attacks under MEAD. In the following setting we attack each detector and the classifier once at the time. α is the parameter to control the losses.

	CIFAR10									
	$\alpha = 0$		$\alpha = .1$		$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1										
PGD1*										
$\varepsilon = 5$	**62.1**	**87.1**	61.3	88.6	61.2	89.3	63.1	89.2	62.6	91.3
$\varepsilon = 10$	**56.8**	90.6	53.1	94.5	54.4	93.9	60.0	91.0	60.6	91.9
$\varepsilon = 15$	**69.3**	**84.4**	51.5	96.5	54.7	94.6	64.1	88.1	65.7	87.7
$\varepsilon = 20$	**78.7**	**73.1**	53.4	96.8	55.9	94.9	66.7	84.1	69.4	82.7
$\varepsilon = 25$	**87.1**	**50.8**	54.0	97.2	56.7	94.6	67.8	82.7	71.1	79.0
$\varepsilon = 30$	**90.3**	**35.4**	54.5	97.1	56.6	94.4	68.9	81.1	71.9	78.4
$\varepsilon = 40$	**92.1**	**22.7**	54.4	97.0	57.7	93.6	69.4	79.7	72.9	74.2
Norm L_2										
PGD2*										
$\varepsilon = 0.125$	**63.9**	**85.4**	61.4	88.0	62.4	88.8	63.7	88.5	63.9	89.9
$\varepsilon = 0.25$	**57.1**	**90.5**	52.9	94.2	55.0	93.6	60.6	89.7	61.5	90.3
$\varepsilon = 0.3125$	**61.0**	**88.9**	51.6	95.7	54.1	94.7	62.2	87.8	63.7	87.9
$\varepsilon = 0.5$	**79.4**	**73.2**	52.8	96.8	55.3	94.3	66.2	84.6	68.8	81.5
$\varepsilon = 1$	**91.4**	**26.4**	52.7	96.8	57.3	93.4	69.0	78.3	72.1	74.4
$\varepsilon = 1.5$	**91.9**	**24.2**	53.9	96.1	57.9	91.4	70.5	73.7	74.1	68.1
$\varepsilon = 2$	**91.9**	**24.1**	54.6	94.6	59.3	88.5	72.3	67.8	75.6	62.7
Norm L_∞										
PGDi*, FGSM*, BIM*										
$\varepsilon = 0.03125$	**82.3**	**59.7**	45.3	96.2	46.0	96.4	54.5	91.4	57.4	89.3
$\varepsilon = 0.0625$	**92.0**	**29.6**	44.3	96.2	49.8	93.8	59.7	82.4	64.3	76.4
$\varepsilon = 0.5$	**94.6**	**9.7**	62.1	81.3	54.9	81.9	66.1	60.8	68.9	57.9
PGDi*, FGSM*, BIM*, SA										
$\varepsilon = 0.125$	**88.9**	**40.8**	48.6	90.7	54.9	85.0	61.9	73.1	66.3	67.5
PGDi*, FGSM*, BIM*, CWi										
$\varepsilon = 0.3125$	**80.0**	**61.1**	56.6	82.0	56.3	79.6	66.1	66.1	69.2	64.4

Table 12: AGREE against the adaptive-attacks under MEAD. In the following setting we attack all the detector and the classifier together at the time. α is the parameter to control the losses.

	CIFAR10									
	$\alpha = 0$		$\alpha = .1$		$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1										
PGD1*										
$\varepsilon = 5$	**62.1**	**87.1**	61.2	90.4	63.6	86.8	65.8	83.9	66.3	83.2
$\varepsilon = 10$	**56.8**	90.6	50.5	96.4	55.9	91.6	60.1	88.1	61.1	87.2
$\varepsilon = 15$	**69.3**	**84.4**	47.3	97.6	53.8	92.3	62.0	84.9	63.7	83.7
$\varepsilon = 20$	**78.7**	**73.1**	47.1	97.9	54.2	92.5	64.2	82.8	66.8	79.1
$\varepsilon = 25$	**87.1**	**50.8**	47.8	98.0	55.0	92.1	66.5	79.5	68.8	77.2
$\varepsilon = 30$	**90.3**	**35.4**	48.8	98.0	55.8	91.3	67.4	78.5	70.4	75.0
$\varepsilon = 40$	**92.1**	**22.7**	49.1	98.0	56.8	90.5	68.6	77.4	72.5	71.6
Norm L_2										
PGD2*										
$\varepsilon = 0.125$	**63.9**	**85.4**	62.4	88.5	65.0	86.2	66.9	82.9	67.2	81.1
$\varepsilon = 0.25$	**57.1**	**90.5**	51.2	96.0	56.3	91.7	60.6	87.2	61.6	86.8
$\varepsilon = 0.3125$	**61.0**	**88.9**	56.0	94.6	57.9	93.6	65.3	86.4	66.7	86.6
$\varepsilon = 0.5$	**79.4**	**73.2**	46.8	97.8	54.6	91.3	64.5	82.4	66.8	79.5
$\varepsilon = 1$	**91.4**	**26.4**	47.2	98.0	57.8	89.4	69.9	73.8	73.1	71.7
$\varepsilon = 1.5$	**91.9**	**24.2**	47.5	97.6	59.9	86.9	73.2	68.7	76.5	63.1
$\varepsilon = 2$	**91.9**	**24.1**	49.0	97.0	62.8	83.3	75.6	63.7	79.5	56.6
Norm L_∞										
PGDi*, FGSM*, BIM*										
$\varepsilon = 0.03125$	**82.3**	**59.7**	40.2	98.0	47.6	95.5	60.6	86.2	65.0	81.8
$\varepsilon = 0.0625$	**92.0**	**29.6**	37.9	98.0	47.0	95.9	61.9	82.1	65.8	77.1
$\varepsilon = 0.25$	**95.9**	**8.8**	36.5	96.4	47.4	97.7	62.5	92.6	65.4	90.8
$\varepsilon = 0.5$	**94.6**	**9.7**	36.7	96.2	46.0	97.7	61.6	96.1	66.0	94.8
PGDi*, FGSM*, BIM*, SA										
$\varepsilon = 0.125$	**88.9**	**40.8**	38.5	95.9	46.8	95.4	60.1	85.0	61.9	83.2
PGDi*, FGSM*, BIM*, CWi										
$\varepsilon = 0.3125$	**80.0**	**61.1**	37.2	95.3	46.7	97.4	60.9	92.4	64.1	90.1