Pattern Recognition 172 (2026) 112519

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

On the robustness of adversarial training against uncertainty attacks

Emanuele Ledda [a,b,*], Giovanni Scodeller [c], Daniele Angioni [a], Giorgio Piras [a], Antonio Emanuele Cinà [c], Giorgio Fumera [a,b], Battista Biggio [a,b], Fabio Roli [a,b,c]

[a] Department of Electric and Electronic Engineering, University of Cagliari, Via Marengo 3, Cagliari, 09100, Italy
[b] CINI, Consorzio Interuniversitario Nazionale per l'Informatica, Via Ariosto 25, Rome, 00185, Italy
[c] Department of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genova, Via Dodecaneso 35, Genova, 16146, Italy

ARTICLE INFO

Keywords:
Uncertainty quantification
Adversarial machine learning
Neural networks

ABSTRACT

In learning problems, the noise inherent to the task at hand hinders the possibility to infer without a certain degree of uncertainty. Quantifying this uncertainty, regardless of its wide use, assumes high relevance for security-sensitive applications. Within these scenarios, it becomes fundamental to guarantee good (i.e., trustworthy) uncertainty measures, which downstream modules can securely employ to drive the final decision-making process. However, an attacker may be interested in forcing the system to produce either (i) highly uncertain outputs jeopardizing the system's availability or (ii) low uncertainty estimates, making the system accept uncertain samples that would instead require a careful inspection (e.g., human intervention). Therefore, it becomes fundamental to understand how to obtain robust uncertainty estimates against these kinds of attacks. In this work, we reveal both empirically and theoretically that defending against adversarial examples, i.e., carefully perturbed samples that cause misclassification, additionally guarantees a more secure, trustworthy uncertainty estimate under common attack scenarios without the need for an ad-hoc defense strategy. To support our claims, we evaluate multiple adversarial-robust classification models from the publicly available benchmark RobustBench on the CIFAR-10 and ImageNet datasets, and on a robust semantic segmentation model evaluated on Pascal-VOC. The code for the reproducibility of the experiments is available at the following link: https://github.com/pralab/UncertaintyAdversarialRobustness.

1. Introduction

In recent years, the presence of Machine Learning (ML) has become increasingly widespread, mainly owing to its powerful predictive abilities. However, the use of ML models, such as any system learning from data, is tied to uncertainty. The presence of variability in the data observations, the limit on the number of observations, and, more generally, the presence of noise on the task at hand make models uncertain [1]. Thus, it has become fundamental to represent uncertainty in a reliable and informative way through the use of Uncertainty Quantification (UQ) methods, enabling an aware decision-making process [2].

The role of UQ techniques assumes a highly pivotal role in security-sensitive applications: in such scenarios, in fact (e.g., medical diagnostics [3,4] and autonomous driving [2]), the tolerance on the error margin must be, by definition, minimized and strictly confined. Leveraging UQ techniques, however, ML-based systems can also base their output on the uncertainty estimated on the given input, helping minimize errors and enabling an accurate and reliable decision-making process throughout the ML systems' pipelines.

Nevertheless, according to a recent promising line of research, the uncertainty estimates can also be subject to attacks aiming to maliciously tamper with the uncertainty [5–8]. Specifically, an attacker can be interested in jeopardizing the availability of the ML-based system, typically by increasing the uncertainty of input samples to "flood" the decision-making system. Conversely, by targeting the system's integrity, an attacker can be interested in decreasing the uncertainty to make the system accept inputs that would otherwise need to undergo the decision-making system, e.g., a human in the loop.

Within the same security-related contexts, however, offspring of an instead now mature line of research, ML models are also often required to be resistant against adversarial inputs, i.e., carefully crafted input samples aiming to deceive the classification performed by the model [9,10]. Following an arms race in the security of ML, the litera-

* Corresponding author.
E-mail addresses: emanuele.ledda@unica.it (E. Ledda), giovanni.scodeller@edu.unige.it (G. Scodeller), daniele.angioni@unica.it (D. Angioni), giorgio.piras@unica.it (G. Piras), antonio.cina@unige.it (A.E. Cinà), giorgio.fumera@unica.it (G. Fumera), battista.biggio@unica.it (B. Biggio), fabio.roli@unige.it (F. Roli).

https://doi.org/10.1016/j.patcog.2025.112519
Received 14 August 2024; Received in revised form 25 September 2025; Accepted 26 September 2025
Available online 3 October 2025

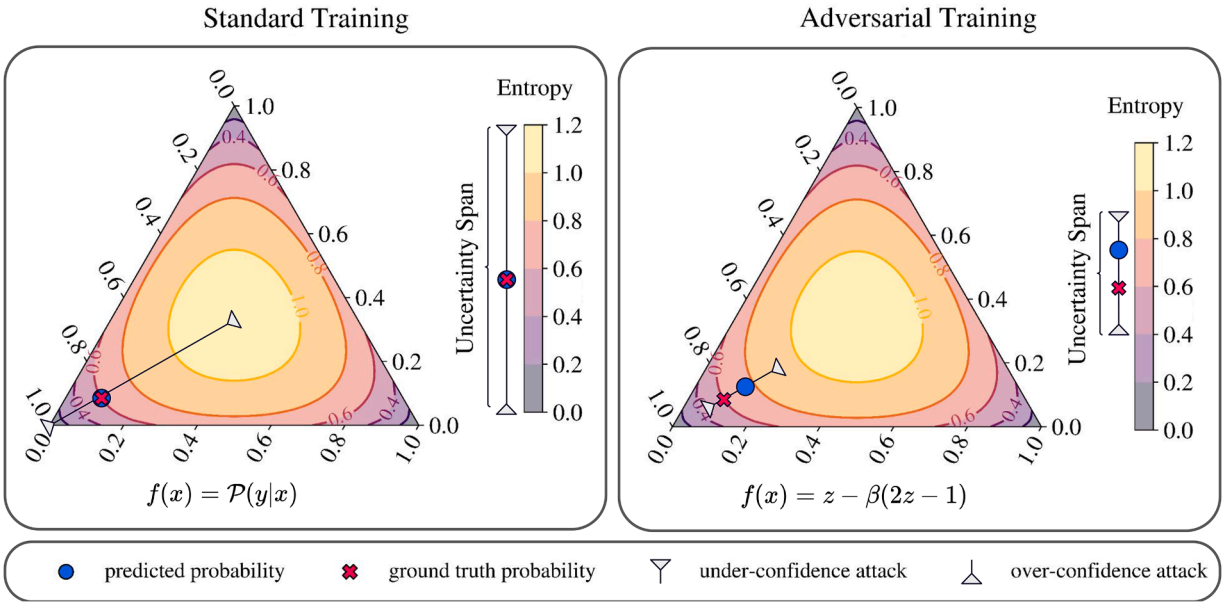


Fig. 1. An illustration of the effect of standard (left) and adversarial (right) training on the Uncertainty Span (i.e., the range of uncertainty values one can obtain by perturbing a sample with a given budget ϵ) for a classification problem visualized through a standard 3-dimensional simplex. While for standard training, the predicted probability (●) converges on the ground truth probability (✖), for adversarial training, it results in less confidence, i.e., has higher entropy than the ground truth probability. As a result, adversarial training reduces the uncertainty span, whose lower and upper bounds are, respectively, an over-confidence ⅄ and under-confidence Y attack. For further details, see Sections 3 and 4.

ture has flourished, proposing multiple defense approaches and attack algorithms [11]. Among the State-Of-the-Art (SoA) defense approaches, Adversarial Training (AT) [12] has been shown to be the reference technique by also being the pillar of more recent approaches, and its effectiveness is deeply and continuously studied against the SoA adversarial attacks [13].

In contrast, for attacks against uncertainty, little to no work has advanced the study of defensive approaches, rather than focusing on different ways to disrupt the UQ techniques. We thus questioned, starting from the principle of evaluating existing approaches first, the effectiveness of AT against attacks on uncertainty. In our work, we reveal, through theoretical analysis and empirical investigation, that models trained to be robust against adversarial attacks can be likewise robust against attacks to uncertainty; we summarize the main theoretical findings of our analysis in Fig. 1.

In Section 2, we display the background concepts of uncertainty quantification and adversarial ML; then, in Section 3, starting from the state of the art, we exemplify a modular framework for dealing with adversarial attacks targeting uncertainty, which is functional to the theoretical analysis on the robustness of adversarial trained models of Section 4. Successively, in Section 5, we empirically demonstrate our results by considering, on multiple datasets, highly robust SoA models for both classification and semantic segmentation. Finally, in Section 6, we outline our findings and discuss the potential limitations of the study at hand.

2. Background

2.1. Uncertainty quantification

Uncertainty Quantification (UQ) is an emerging ML field devoted to measuring uncertainty (usually, the uncertainty associated with models' predictions) [1]. This uncertainty, commonly known as *predictive uncertainty*, can be decomposed into two distinct sources [1]: (i) **aleatoric uncertainty**: which refers to the inherent randomness of the decision process; (ii) **epistemic uncertainty**: caused by a lack of knowledge. An example of aleatoric uncertainty is the ambiguity when trying to discriminate the hand-written digit "5" from the letter "S". In some extreme cases, the similarities between the strokes composing these two symbols can make them even indistinguishable. Unlike the previous example, epistemic uncertainty has no relation to any inherent randomness of the process: for example, imagine determining if a mushroom is poisonous by its appearance. Without proper knowledge of the target domain, one would be uncertain in determining if a given mushroom would be poisonous; however, knowledge acquisition reduces the uncertainty of the predictions (the reason for which we also refer to this source as "reducible uncertainty").

Bayesian approach to uncertainty quantification – One solid approach for quantifying these two sources of uncertainty consists of estimating a posterior distribution on the set of the model's parameters θ given the data $\mathcal{D}$:

$$p(\theta|D) = \frac{p(D|\theta) \cdot p(\theta)}{p(D)}. \tag{1}$$

From this distribution, one can obtain a prediction by Bayesian Model Averaging (BMA), i.e., by computing the expected value of the prediction's posterior $p(y|\boldsymbol{x}, \theta)$ given the parameters θ with respect to the parameter's posterior $p(\theta|D)$:

$$f^{BMA}(\boldsymbol{x}) = p(y|\boldsymbol{x}, D) = \int_{\Theta} p(y|\boldsymbol{x}, \theta) \cdot p(\theta|D)\, \partial\theta, \tag{2}$$

where Θ is the set of all the possible parameters. For obtaining uncertainties associated directly with the model's prediction, the standard approach consists of using a second-order probability over the prediction's Bayesian posterior: by convention, the first-order moment of this distribution is "believed" to embed information about the aleatoric uncertainty, while the second order moment is *directly* associated to epistemic uncertainty.

A standard approach consists of using the entropy of the predictive vector as a measure of aleatoric uncertainty. Practically, for classification problems, such a solution acts directly on the predicted softmax vector encoding the prediction $f(\boldsymbol{x})$:

$$H(f(\boldsymbol{x})) \coloneqq -\sum_{c=1}^{|\mathcal{Y}|} f_c(\boldsymbol{x}) \log f_c(\boldsymbol{x}), \tag{3}$$

where $f_c(\boldsymbol{x})$ denotes the probability value the model assigns to the class c, with $f(\boldsymbol{x})$ which can be obtained either by using a Bayesian model $f(\boldsymbol{x}) \coloneqq f^{BMA}(\boldsymbol{x})$ or a deterministic one $f(\boldsymbol{x}) \coloneqq f^{\theta}(\boldsymbol{x})$ parametrized with θ. Correctly modeled entropies should convey calibration properties, i.e., when the model assigns a probability value of p to an event (e.g., belonging to a specific class), it actually occurs with a probability of p. The standard metric for quantifying this behavior takes the name of Expected Calibration Error (ECE) [14], which is a discrete expectation of the absolute difference between the expected $ex(\mathbf{B}_s)$ and the observed $ob(\mathbf{B}_s)$ probability computed on a set of S buckets $\mathbf{B}_s$:

$$\mathrm{ECE} = \sum_{s=1}^{S} \frac{|\mathbf{B}_s|}{|D|} |ob(\mathbf{B}_s) - ex(\mathbf{B}_s)|, \tag{4}$$

where each $\mathbf{B}_s \in D$ contains all the samples having confidence (i.e., $\arg\max f(\boldsymbol{x})$) within the interval $[\frac{s}{S}, \frac{s+1}{S}]$, and $ex(\mathbf{B}_s) = \frac{2s+1}{2S}$ is the average over the interval. For measuring epistemic uncertainty, a common approach employs the variance of the predictions: $\mathbb{V}(f^{BMA}(\boldsymbol{x})) \coloneqq \mathbb{E}[(f^{BMA}(\boldsymbol{x}))^2] - \mathbb{E}[(f^{BMA}(\boldsymbol{x}))]^2$.

2.2. Adversarial attacks

In a security-sensitive application, another priority, together with reliable uncertainty estimates, is to guarantee a certain level of robustness in a model that can be the target of attackers aiming at disrupting its normal functioning. This necessity brought to light what is known as *adversarial machine learning*, an ML field that keeps exploring the attack surfaces of an ML system to propose new countermeasures to achieve robustness [11]. In this regard, most of the research community efforts focused on attacks jeopardizing the integrity of a model, where an attacker is interested in changing a model's prediction at test time [9,10]. In such an attack scenario, the goal of the attacker is to craft a small perturbation $\boldsymbol{\delta}'$ such that, once added to an input sample $\boldsymbol{x}$, it induces the target model f^{θ} to output a wrong prediction, i.e., $f^{\theta}(\boldsymbol{x} + \boldsymbol{\delta}') \neq f^{\theta}(\boldsymbol{x})$. Here, the perturbation is typically constrained to a limited domain $\mathcal{B}_\epsilon = \{\forall \boldsymbol{\delta} \ : ||\boldsymbol{\delta}||_p \leq \epsilon\}$, where $||.||_p$ is an arbitrary L_p norm (e.g., L_1, L_2 or L_∞) and ϵ is the perturbation budget indicating the limits of the attacker's power.

Optimizing the adversarial perturbation – To find the optimal perturbation $\boldsymbol{\delta}'$ that maximizes the probability of success for the attacker, one can solve the following optimization problem:

$$\boldsymbol{\delta}' = \underset{\boldsymbol{\delta} \in \mathcal{B}_\epsilon}{\operatorname{argmin}} \mathcal{L}'(\boldsymbol{t}, f^{\theta}(\boldsymbol{x} + \boldsymbol{\delta})), \tag{5}$$

where $\boldsymbol{t}$ is the one-hot encoding of the target label, which can be either any label different than the ground truth or a specific target class chosen by the attacker, and $\mathcal{L}'$ is the attacker's objective, e.g., the negative cross-entropy between the prediction and the target or the logit $f_k(\boldsymbol{x} + \boldsymbol{\delta}')$ corresponding to the correct class.

Defending against adversarial attacks – Currently, the most promising approach to protect a model against adversarial attacks is to proactively anticipate the attacker by including them during training. Such a procedure, which takes the name of *adversarial training* (AT) [12], aims at finding the optimal parametrization θ' that ideally protects from all possible perturbations $\boldsymbol{\delta} \in \mathcal{B}_\epsilon$. This strategy can be formalized as the following minimax problem:

$$\min_{\theta \in \Theta} \mathbb{E}_{\boldsymbol{x}, y \sim D} \left[\max_{\boldsymbol{\delta} \in \mathcal{B}_\epsilon} \mathcal{L}(f^{\theta}(\boldsymbol{x} + \boldsymbol{\delta}), y) \right], \tag{6}$$

where $\mathcal{L}$ is the objective that minimize classification error (e.g., the cross-entropy loss). Here, Eq. (6) is composed of (i) an *inner maximization* problem to find, for each sample, the perturbation $\boldsymbol{\delta}$ that achieves high loss, and (ii) an *outer minimization* problem to find the parameters θ that minimize this worst-case loss.

3. Adversarial attacks against uncertainty quantification

From the perspective of the most typical test-time evasion attacks, adversarial machine learning characterizes an attacker interested in making the model misclassify the prediction on the input adversarial example. In the context of Uncertainty Quantification, instead, the focus shifts from "deteriorating predictions" to "deteriorating the uncertainty measure". The correct functioning of an uncertainty measure implies that the most likely correct predictions have a low uncertainty, which, in contrast, becomes high when the predictions are plausibly incorrect. Therefore, while predictions and uncertainty represent two different measures, they are, however, affine: the uncertainty estimate measures the likelihood that the given prediction is wrong. As we will see in the following subsection, state-of-the-art work conceived uncertainty adversarial attacks specifically to spoil this statistical correlation.

3.1. Previous work

Galil and El-Yaniv [5] are the first to consider attacks to disrupt the uncertainty measure. Specifically, they consider using such a measure for building a classifier with a rejection option. They aim to subvert the rejection decision (i.e., to increase the probability of rejecting correctly classified samples and decrease the probability of rejecting the misclassified ones) without changing the predicted labels so that a user would not notice the attack. To do so, they selectively increase or decrease the confidence assigned to each sample *depending on the model's prediction before the attack* by using the Maximum Softmax Probability (MSP) as loss function $\ell(\boldsymbol{x}) = f_{\hat{y}}(\boldsymbol{x})$, where $\hat{y}$ is the predicted class:

$$\begin{aligned} \underset{\boldsymbol{\delta} \in \mathcal{B}_\epsilon}{\operatorname{argmin}} \quad & \gamma \cdot f_{\hat{y}}(\boldsymbol{x} + \boldsymbol{\delta}) \\ \text{s.t.} \quad & f_{\hat{y}}(\boldsymbol{x} + \boldsymbol{\delta}) \geq f_c(\boldsymbol{x} + \boldsymbol{\delta}), \quad \forall c \in \mathcal{Y}, \end{aligned} \tag{7}$$

where $\gamma = -1$ is used for misclassified samples and $\gamma = 1$ for correctly the classified ones. Zeng et al. [6] model an attacker interested in making Out-of-Distribution (OOD) samples appear as in-distribution samples. To do so, they propose to minimize the cross-entropy between the adversarial prediction $f(\boldsymbol{x}_{out} + \boldsymbol{\delta})$ for an OOD sample $\boldsymbol{x}_{out}$ and the target $\boldsymbol{t}$ representing the one-hot encoding of the predicted label $\hat{y}$:

$$\underset{\boldsymbol{\delta} \in \mathcal{B}_\epsilon}{\operatorname{argmin}} \quad H(f(\boldsymbol{x}_{out} + \boldsymbol{\delta}), \boldsymbol{t}). \tag{8}$$

It is worth noting that both the techniques mentioned above [5,6] design an attacker who has access to prior knowledge about input annotations. In the former, the ground truth is crucial to determine when to increase or decrease the confidence, while, in the latter, the attacker needs to know which samples are OOD (i.e., implicitly not belonging to any known class) to perform the attack.

In our previous work [7], we proposed an alternative scenario in which the attacker is interested in increasing or decreasing the predictive uncertainty of all the input samples. The ability to damage predictive uncertainty without having specific information about the labels is an attractive feature, especially if such a measure is the input of a downstream module or is given to a human. For instance, imagine a system employing the uncertainty measure of an ML model for determining if the clinical data of a patient is sufficient to make a diagnosis: without knowledge of this specific domain (thus without the labels), an attacker may craft a perturbation for increase indiscriminately the uncertainty associated to the patient's data. Such an attack may overload the processing pipeline of a hospital, causing the system to require more analysis, even when unnecessary. The proposed attack scenario takes the name of *over-confidence* (when decreasing the uncertainty) or *under-confidence* attack (when increasing the uncertainty). While for the over-confidence attack, we kept the same optimization process of Eq. (8), the under-confidence case has been investigated only theoretically, without an actual loss function implementation.

Obadinma et al.[8] propose to harm the model's calibration, disrupting the functioning of any downstream module or human using the uncertainty measure. For doing so, they present four different attacks, two of which coincide with the formulation we proposed in [7], i.e., (i) over-confidence and (ii) under-confidence attack. They further propose (iii) *maximum miscalibration attack* (MMA), where they increase or

decrease the confidence of each sample following the same criteria of [5], and (iv) *random confidence attack* (RCA), where they extend the latter but with a randomized confidence goal for each input, producing a more natural confidence distribution that, although being a less effective attack, results in being more stealthy. Based on these types of attacks against model's calibration, they propose two ad-hoc defenses: (i) *Calibration Attack Adversarial Training* (*CAAT)*, in which they perform adversarial training using samples generated with maximum miscalibration attacks, and (ii) *Compression Scaling* (*CS*), a post-processing technique where they heuristically distribute low scores to higher range. To the best of our knowledge, this is the only work that proposes a defense strategy against uncertainty attacks.

To the best of our knowledge, all state-of-the-art approaches rely exclusively on aleatoric uncertainty estimates derived from deterministic models, whereas epistemic uncertainty in the context of adversarial robustness has been explored solely in our prior work [7]. This common practice is not a deliberate choice, but rather a consequence of the unpopularity of Bayesian models among the adversarial training community, which makes robust models capable of providing principled epistemic measures hard to find.

3.2. A modular unified framework for uncertainty attacks

Interestingly, all the state-of-the-art works build an optimization process with a proper loss formulation for accomplishing an attacker's goal. This means that proving the model's robustness against a specific goal does not directly translate to the same robustness guarantees on other goals. For example, analyzing the effect of an attack on OOD uncertainty minimization does not provide any information on the effectiveness against indiscriminate uncertainty maximization, making it difficult to compare models in terms of uncertainty robustness. Accordingly, we propose a modular approach for unifying the diverse attacks proposed in the literature, which, as a result, also unifies the notion of robustness. In general, all the possible instances of the attacker's goal have a shared characteristic: they all need to modify the uncertainty measure associated with the model's predictions; more sensitive uncertainty variations generally lead to less robustness against uncertainty attacks. As a result, any possible uncertainty attack discussed in literature (e.g., Maximum Miscalibration [8], Random Confidence [8], and OOD camouflage [5]) can be traced back to a combination of over- and under-confidence attacks. Let us take random confidence as an example: knowing which is the maximum and minimum attainable uncertainty value under attack for a specific set of samples directly influences the success rate of the random confidence attack (the higher the possible excursion, the higher the space of action of the attack). As another example, given that for succeeding in a maximum miscalibration attack it is necessary to selectively increase or decrease the uncertainty of different samples, the attack success rate depends upon the robustness of the models against under- and over-confidence attacks.

Therefore, instead of focusing on specific instances of the attacker's goal, we *estimate the range of uncertainty values an attacker can reach* when perturbing each input x on the dataset $\mathcal{D}$. This approach provides a comprehensive overview of the model's vulnerability under uncertainty manipulations, encompassing all possible attack scenarios. Concretely, given an attacker capable of perturbing a sample $\boldsymbol{x}$ inside a domain $\mathcal{B}_\epsilon$, we can distinguish between two different cases: (i) decreasing the uncertainty (i.e., over-confidence attack) and (ii) increasing the uncertainty (i.e., under-confidence attack). These objectives can be achieved, respectively, by minimising or maximising the uncertainty measure $\mathcal{U}(f^{\boldsymbol{\theta}}(\boldsymbol{x}))$:

$$\underbrace{\boldsymbol{\delta}^{\downarrow} = \underset{\boldsymbol{\delta}\in\mathcal{B}_\epsilon}{\operatorname{argmin}}\ \mathcal{U}(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}))}_{\textbf{Over-confidence Attack}} \qquad \underbrace{\boldsymbol{\delta}^{\uparrow} = \underset{\boldsymbol{\delta}\in\mathcal{B}_\epsilon}{\operatorname{argmax}}\ \mathcal{U}(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}))}_{\textbf{Under-confidence Attack}} \tag{9}$$

Here, $\boldsymbol{\delta}^{\downarrow}$ represents the perturbation for which the attacker minimizes the uncertainty (over-confidence attack) while $\boldsymbol{\delta}^{\uparrow}$ represents the perturbation for which it is maximized (under-confidence attack). Starting from them, we can compute the lower $\mathcal{U}^{\downarrow} \coloneqq \mathcal{U}(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow}))$ and upper $\mathcal{U}^{\uparrow} \coloneqq \mathcal{U}(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\uparrow}))$ bound of what we call the **Uncertainty Span** (US), i.e., all the uncertainty values an attacker can obtain when perturbing the sample $\boldsymbol{x}$ for fooling a model parameterized by $\boldsymbol{\theta}$ given a perturbation budget ϵ. With such an approach, one can easily derive the robustness to specific uncertainty attacks, e.g., by selectively choosing the upper/lower bound of the uncertainty range [5,6] or even intermediate values [8].

In accordance with the state of the art, we use the entropy of the prediction vector $H(f^{\boldsymbol{\theta}}(\boldsymbol{x}))$ [1] as our uncertainty measure $\mathcal{U}(\cdot)$. To instantiate an over-confidence attack, the objective is to minimize the entropy of the output probability. To do so, it is sufficient to force the model to output $f^{\boldsymbol{\theta}}_k(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow}) = 1$ for an arbitrary $k \in \mathcal{Y}$. We know from [7] that the most straightforward strategy for the attacker is to maximize the probability of the class corresponding to the predicted class, as it already has the highest probability among all the classes, as in Eq. (8); this attack takes the name of Stabilizing Attack (STAB) since it stabilizes the model's prediction by enforcing the most likely class [7]. To achieve this objective, one can employ the cross-entropy loss, setting as target $\boldsymbol{t}^{\downarrow}$ the one-hot-encoding of the predicted label $\hat{y}$:

$$\underset{\boldsymbol{\delta}^{\downarrow}\in\mathcal{B}_\epsilon}{\operatorname{argmin}} H(\boldsymbol{t}^{\downarrow}, f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow})) = \underset{\boldsymbol{\delta}^{\downarrow}\in\mathcal{B}_\epsilon}{\operatorname{argmin}} -\log(f^{\boldsymbol{\theta}}_{\hat{y}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow})), \tag{10}$$

which finds its minimum when $f_{\hat{y}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow}) = 1$. In the case of under-confidence attacks, we carry out a similar approach by using the cross-entropy: to this end - since the entropy reaches its maximum when the output is a uniform vector - we set the target $\boldsymbol{t}^{\uparrow} = \frac{1}{c}\cdot\mathbb{1}$ to be a c-dimensional vector with all elements equal to $\frac{1}{c}$, where c is the total number of classes:

$$\underset{\boldsymbol{\delta}^{\downarrow}\in\mathcal{B}_\epsilon}{\operatorname{argmin}} H(\boldsymbol{t}^{\uparrow}, f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\uparrow})) = \underset{\boldsymbol{\delta}^{\downarrow}\in\mathcal{B}_\epsilon}{\operatorname{argmin}} -\frac{1}{c}\sum_{k=1}^{c}\log(f^{\boldsymbol{\theta}}_k(\boldsymbol{x}+\boldsymbol{\delta}^{\uparrow})). \tag{11}$$

Finally, for a thorough evaluation, we must extend the notion of the Uncertainty Span (a sample-wise measure) onto an entire data set. To this aim, we propose to compute general statistics by taking the mean of the uncertainty spans[1] (MUS):

$$\text{MUS} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[H(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\uparrow})) - H(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow}))\right]. \tag{12}$$

We can also penalize larger gaps by employing a *squared* mean (MSUS):

$$\text{MSUS} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\left(H(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\uparrow})) - H(f^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}^{\downarrow}))\right)^2\right]. \tag{13}$$

4. Theoretical analysis on adversarial uncertainty robustness

Despite much literature on the attacker side, only one work dealt with adversarial robustness against uncertainty attacks, proposing two ad hoc strategies. In contrast to them, driven by the flourishing literature on defenses against prediction attacks (i.e., traditional adversarial examples), before thinking of ad hoc strategies, we ask ourselves if and to what extent such defenses, particularly adversarial training, may protect even against uncertainty attacks. Accordingly, our main research question is:

> Do **adversarial trained** models disclose robust characteristics also **against attacks targeting uncertainty**?

A recent work [8] likewise raised this observation, but it is still limited to empirically analyzing a small set of models and only against

[1] By abuse of notation, we refer to Uncertainty Span for indicating both the interval and the interval size.

specific attack instances. To this aim, instead of considering specific attack instances, we answer our research question by analyzing the Uncertainty Span of adversarial trained models, i.e., consider both the under- and over-confidence cases, in accordance with Section 3, drawing *general* conclusions on these models' robustness against *any* attack instance. Since the predictive entropy of a sample increases when approaching the decision boundary, it is reasonable to think that adversarial training may also protect against under-confidence attacks: in fact, adversarial training forces the model to be robust to label-flips (i.e., to samples crossing the boundary), hindering any potential attacker interested in increasing uncertainty. For over-confidence attacks instead, we cannot draw similar conclusions: indeed, being robust to label flips does not guarantee an entropy reduction since the regions where the entropy is minimized do not lie on the decision boundary.

Until now, we have provided a qualitative and intuitive discussion; now, let us move on to a formal, theoretical analysis.

4.1. On the convergence of adversarial trained models

For simplicity, we consider a binary classification problem: let $\mathcal{P}(\mathcal{X}, \mathcal{Y})$ be a generative distribution; sampling from this distribution results in obtaining a pair $(\boldsymbol{x}, y)$ characterized by an observation $\boldsymbol{x} \in \mathbb{R}^d$ and a target binary label $y \in \{A, B\}$. Let then D be a collection of i.i.d. pairs $(\boldsymbol{x}, y)$ obtained sampling the generative distribution $\mathcal{P}(\mathcal{X}, \mathcal{Y})$:

$$D = \{(\boldsymbol{x}_i, y_i) \sim \mathcal{P}(\mathcal{X}, \mathcal{Y}),\ i = [1, \dots, N]\}, \tag{14}$$

Note that for the law of large numbers, when the cardinality of the data set $|D|$ increases, the probability $p(y|\boldsymbol{x}, D)$ conditioned to the data set tends to $p(y|\boldsymbol{x})$:

$$\lim_{|D|\to\infty} p(y|\boldsymbol{x}, D) = p(y|\boldsymbol{x}). \tag{15}$$

Eq. (15) represents the ideal condition for our theoretical analysis: considering the edge case of a large data set, we can filter out any source of epistemic uncertainty and, thus, study the effect of aleatoric uncertainty in isolation.

Uncertainty convergence of standard training – In a binary classification problem, the standard approach consists of using the cross-entropy loss $\ell(\boldsymbol{x}, y) = H(\boldsymbol{t}, f^{\boldsymbol{\theta}}(\boldsymbol{x}))$, corresponding to the entropy between the target $\boldsymbol{t}$ (representing the one-hot encoding of y) and the output of the classifier $f^{\boldsymbol{\theta}}(\boldsymbol{x})$ parameterized with $\boldsymbol{\theta}$ evaluated on $\boldsymbol{x}$. The empirical loss corresponds to the sum of the losses for each data point $\mathcal{L}(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} \ell(\boldsymbol{x}_i, y_i)$. Therefore, to minimize $\mathcal{L}(D)$, one have to find a suitable parametrization $\theta^* = \arg\min_\theta \mathcal{L}(D)$ minimizing the loss.

Lemma 1. *A classifier $f^{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ minimizes the loss $\mathcal{L}(D)$ when, for all $(\boldsymbol{x}, y) \in D$, it holds $f^{\boldsymbol{\theta}}(\boldsymbol{x}) = p(y|\boldsymbol{x})$*

Proof. Let $\mathbb{P}_D = \{D_{\boldsymbol{x}_0}, \dots D_{\boldsymbol{x}_K}\}$ be a partition of D where each element $D_{\boldsymbol{x}_i} \coloneqq \{(\boldsymbol{x}, y) \in D \mid \boldsymbol{x} = \boldsymbol{x}_i\}$ represents the set of all the samples having feature vector $\boldsymbol{x}_i$. Since minimizing the loss for each $D_{\boldsymbol{x}} \in \mathbb{P}_D$ results in minimizing $\mathcal{L}(D)$, we discuss the minimization of a single generic subset $D_{\boldsymbol{x}}$. Such subset contains either elements belonging to class A or B: accordingly, let the one hot encoded targets be $\boldsymbol{t}_{\boldsymbol{A}} \coloneqq [1, 0]^\intercal$ and $\boldsymbol{t}_{\boldsymbol{B}} \coloneqq [0, 1]^\intercal$. Let then $\boldsymbol{z} = [z, 1-z]^\intercal$ be the vector associated to the categorical distribution of $p(y|\boldsymbol{x})$, i.e., a vector where z and $1-z$ equal respectively $p(A|\boldsymbol{x})$ and $p(B|\boldsymbol{x})$. If a classifier $f^{\boldsymbol{\theta}}(x)$ evaluated on x outputs a probability vector $\boldsymbol{\alpha} \coloneqq [\alpha, 1-\alpha]^\intercal$, one can expand the loss term on the subset D_x as follows:

$$\begin{aligned}\mathcal{L}(D_{\boldsymbol{x}}) &= \underbrace{p(\mathrm{A}|\boldsymbol{x})}_{z} \cdot \underbrace{H(\boldsymbol{t}_{\boldsymbol{A}}, f^{\boldsymbol{\theta}}(\boldsymbol{x}))}_{-\log(\alpha)} + \underbrace{p(\mathrm{B}|\boldsymbol{x})}_{1-z} \cdot \underbrace{H(\boldsymbol{t}_{\boldsymbol{B}}, f^{\boldsymbol{\theta}}(\boldsymbol{x}))}_{-\log(1-\alpha)} \\ &= -z \cdot \log(\alpha) - (1-z)\log(1-\alpha) = H(\boldsymbol{z}, \boldsymbol{\alpha}).\end{aligned}$$

Finally, since $H(\boldsymbol{z}, \boldsymbol{\alpha})$ is minimized when $\boldsymbol{z} = \boldsymbol{\alpha}$, one can conclude that the loss $\mathcal{L}(D)$ on the entire data set is minimized when this condition holds for all $(\boldsymbol{x}, y) \in D$ in the data set. □

Lemma. 1 represents a well-known result of probabilistic machine learning [15], stating that the optimal classifier should be perfectly calibrated. This phenomenon should be considered when dealing with aleatoric uncertainty because it asserts that lower entropy does not always go along with more accurate classifiers. From this starting point, we will now show that such conditions slightly change in the presence of an attacker.

Uncertainty convergence of adversarial training – Unlike standard training, the data we use during the minimization problem of adversarial training changes when the weight configuration changes: this happens because the perturbations applied to the input samples used during training depend directly upon the weights $\boldsymbol{\theta}^{(t)}$ at each iteration t. From this perspective, one can extend the notation used with clean data to construct a dataset containing adversarial examples for training the model. Let D be the clean data set, and let D_{adv} be the adversarial data set built upon D:

$$D_{\text{adv}}^{(t)} = \{(\boldsymbol{x} + \delta^{(t)}, y) \mid (\boldsymbol{x}, y) \sim \mathcal{P}(\mathcal{X}, \mathcal{Y})\}, \tag{16}$$

where $\delta^{(t)}$ is an adversarial perturbation applied to $\boldsymbol{x}$, obtained by optimizing Eq. (6) with the parameters $\boldsymbol{\theta}^{(t)}$ at iteration t. Since we are considering a binary classification problem, one has to distinguish two cases for the same $\boldsymbol{x}$: if $y = A$, then Eq. (6) is minimized when $f^{\boldsymbol{\theta}}(x + \delta^{(t)})$ equals $\boldsymbol{t}_{\boldsymbol{B}} \coloneqq [0, 1]^\intercal$, whereas if $y = B$, it is minimized when $f^{\boldsymbol{\theta}}(x + \delta^{(t)})$ equals $\boldsymbol{t}_{\boldsymbol{A}} \coloneqq [1, 0]^\intercal$.

The minimization-maximization process leads to an equilibrium where the classifier would not improve its robustness by updating its parameters $\boldsymbol{\theta}^{(t)}$, because even though any update may eventually fix some vulnerabilities, it would at the same time expose the model to new ones. Therefore, for each $(\boldsymbol{x}, y)$, this equilibrium theoretically bounds the loss an attacker can achieve when minimizing $p(y = c|\boldsymbol{x})$ for a given target $c \in \mathcal{Y}$. Specifically, if we let the attack strength for the target class c be $\beta_c = \frac{\|f_c^{\boldsymbol{\theta}(\boldsymbol{x})} - f_c^{\boldsymbol{\theta}(\boldsymbol{x}+\delta)}\|_2}{\sqrt{2}}$, the equilibrium would be described by the following lemma:

Lemma 2. *An adversarial trained classifier $f^{\boldsymbol{\theta}_R}$ with robust parameters $\boldsymbol{\theta}_R$, minimizes the loss $\mathcal{L}^{\boldsymbol{\theta}_R}(D_{adv})$ when, for all $(\boldsymbol{x} + \delta, y) \in D_{adv}$ it holds $\alpha = z - \beta(2z - 1)$, where $\beta = \max(\beta_A, \beta_B)$.*

Proof. Let $\mathbb{P}_{D_{\text{adv}}} = \{D_{\text{adv}}^{\boldsymbol{x}_0}, \dots, D_{\text{adv}}^{\boldsymbol{x}_K}\}$ be a partition of D_{adv} where $D_{\text{adv}}^{\boldsymbol{x}} \coloneqq \{(\boldsymbol{x} + \delta, y) \in D_{\text{adv}} \mid (\boldsymbol{x}, y) \in D_{\boldsymbol{x}}\}$. One can compute the loss on this partition as follows:

$$\begin{aligned}\mathcal{L}^{\boldsymbol{\theta}_R}(D_{\text{adv}}^{\boldsymbol{x}}) &= \underbrace{p(\mathrm{A}|\boldsymbol{x} + \delta_{\mathrm{A}})}_{z} \cdot \underbrace{H(\boldsymbol{t}_{\boldsymbol{A}}, f^{\boldsymbol{\theta}}(\boldsymbol{x} + \delta_{\mathrm{A}}))}_{-\log(\alpha-\beta_A)} + \underbrace{p(\mathrm{B}|\boldsymbol{x} + \delta_{\mathrm{B}})}_{1-z} \cdot \underbrace{H(\boldsymbol{t}_{\boldsymbol{B}}, f^{\boldsymbol{\theta}}(\boldsymbol{x} + \delta_{\mathrm{B}}))}_{-\log(1-\alpha-\beta_B)} \\ &= -z \cdot \log(\alpha - \beta_A) - (1-z) \cdot \log(1 - \alpha - \beta_B).\end{aligned}$$

Note that, for simplicity, one can consider the worst case by taking as attack strength $\beta = \max(\beta_A, \beta_B)$; in this configuration, we can write the loss as follows:

$$\mathcal{L}^{\boldsymbol{\theta}_R}(D_{\text{adv}}^{\boldsymbol{x}}) = -z \cdot \log(\alpha - \beta) - (1-z) \cdot \log(1 - \alpha - \beta). \tag{17}$$

Contrary to Lemma. 1, when using adversarial training, the loss does not equal a cross-entropy between two distinct components. Therefore, we need to compute the values for which the partial derivative with respect to α equals zero:

$$\implies \frac{\partial \mathcal{L}}{\partial \alpha} = 0 \implies \frac{1-z}{1-\alpha-\beta} - \frac{z}{\alpha-\beta} = 0 \implies \alpha = z - \beta \cdot (2z - 1).$$

□

4.2. Demystifying the underconfidence of robust models

From Lemma. 1 and Lemma. 2, it is possible to derive that, in the presence of an attacker, a classifier inevitably deviates from the trajectory of the optimization process that it would have followed in a standard, clean optimization process. Interestingly, the entropy of an adversarial-trained classifier is higher compared to standard training. We can easily prove this inequality starting from the two lemmas:

Theorem 1. *Let f^{θ_S} and f^{θ_R} be two classifiers trained respectively with standard and adversarial training on the data set $\mathcal{D}$. In the presence of an attacker with attack strength $\beta \geq 0$, for all $(x, y) \in \mathcal{D}$ holds that:*

$$H(f^{\theta_S}(x)) \leq H(f^{\theta_R}(x)) \tag{18}$$

Proof. From Lemma. 1, we know $f^{\theta_S}(x) = [z, 1-z]^\intercal$, while from Lemma. 2 we know that in the presence of an attacker, the optimum deviates from an offset amount of $o = \beta(2z-1)$: $f^{\theta_R}(x) = [z-o, 1-z+o]^\intercal$. Within this interval, the entropy $H(f^{\theta_R}(x))$ reaches its minimum when the offset equals zero and, thus, when $f^{\theta_S}(x) = f^{\theta_R}(x)$; therefore, $H(f^{\theta_S}(x)) \leq H(f^{\theta_R}(x))$. □

From Theorem. 1, we can draw some conclusions on the robustness against uncertainty attacks by studying the effect of adversarial training on the Uncertainty Span. Albeit Theorem. 1 proves the tendency of adversarial training to produce underconfident models, it remains factual the robustness of such models against *traditional* adversarial examples, i.e., crafted to induce a misclassification on a sample [12]. In order to succeed, the adversarial example requires crossing the decision boundary; in a binary classification problem, such conditions correspond precisely to reaching the maximum possible entropy. Consequently, reducing the number of samples for which a traditional adversarial example exists (i.e., increasing the robust accuracy) directly implies reducing the upper bound of the uncertainty span. Concerning the robustness against over-confidence attacks, findings from Theorem. 1 suggest greater robustness against such attacks as well. This theorem shows that adversarial-trained models output predictions with higher entropies than undefended models. Suppose we aim to decrease it until reaching a desired value of v: in that case, the initially higher entropy of the adversarial trained model makes it more challenging to lower it to v, as the starting values are already higher and, therefore, need a significant reduction.

Discussion – In safe environments (without any attacker), a model whose purpose is to solve a classification problem takes advantage of correctly modeling the true posterior $p(y|\boldsymbol{x})$ for any given input sample. Nevertheless, in the presence of an attacker, the classifier tends to balance good generalization capabilities and security guarantees. Such a balance compromises not only the model's accuracy (as we already know from previous literature) but also comes at the price of sacrificing a certain amount of confidence in the predictions. As a result, adversarial-trained models disclose robust characteristics against uncertainty attacks, providing theoretical support for the claim that adversarial training makes models more resilient to over- and under-confidence attacks.

5. Experimental analysis

We now present the experimental setup employed to empirically validate our theoretical findings by analysing the Uncertainty Span on a selection of 23 state-of-the-art models from the well-known Robustbench repository [13]. In addition, we also consider a concrete case study where an attacker is interested in tampering with the model's calibration, proving the robustness of adversarial trained models for a real scenario.

5.1. Experimental setup

Datasets – We consider two well-known datasets used for classification tasks, CIFAR-10[2], and ImageNet[3].

We generate 8000 adversarial examples from each dataset by randomly sampling from their respective test sets.

Models – For a comprehensive investigation, we consider 15 CIFAR-10 (denoted as C1-C15) and 8 ImageNet (denoted as I1-I8) standard robust classifiers from the Robustbench repository [13], encompassing many different model architectures and robust accuracies, which are summarized in Table 1. We consider a wide range of different network architectures, including Convolutional Neural Networks (CNN) and Visual Tranformers (ViT); specifically, for analysing the uncertainty robustness when the model's size changes, we selected a wide range of model capacity, including ResNet-18, ResNet-50, ResNet-152, WideResNet-28-10, WideResNet-70-10, RaWideResNet-28-10, Swin-B, Swin-L, ConvNeXt-B and ConvNeXt-L. Many of them rely on data augmentation (C2 [16], C3, C4 [17], C5,C6 [17], C9 [18], I5, I6, I7, I8 [19]) or transfer learning with synthetic data (C7, C8 [20], I2, I3 [21]), while only two classifiers perform adversarial training using solely the original set (C1, I1 [22]). Some work proposes more sophisticated strategies such as: (i) weight averaging (C9 [18], I5, I6, I7, I8 [19]); (ii) ODE networks (C10 [23]); (iii) robust residual blocks (C11 [24]); (iv) modifications on the optimization process (C12 [25], C13 [26], C14 [27], C15 [28]).

Table 1
The RobustBench models used for our analysis. For more details on the specific hyperparameters used during the adversarial training phase, we refer the reader to the original papers.

ID	RobustBench Name	Architecture
CIFAR-10		
C1 [22]	Engstrom2019Robustness	ResNet-50
C2 [16]	Addepalli2022Efficient_RN18	ResNet-18
C3 [17]	Gowal2021Improving_28_10_ddpm_100m	WideResNet-28-10
C4 [17]	Gowal2021Improving_70_16_ddpm_100m	WideResNet-70-16
C5 [29]	Wang2023Better_WRN-28-10	WideResNet-28-10
C6 [29]	Wang2023Better_WRN-70-16	WideResNet-70-16
C7 [20]	Sehwag2021Proxy_R18	ResNet-18
C8 [20]	Sehwag2021Proxy_ResNest152	ResNet-152
C9 [18]	Rebuffi2021Fixing_70_16_cutmix_extra	WideResNet-70-16
C10 [23]	Kang2021Stable	WideResNet-70-16
C11 [24]	Peng2023Robust	RaWideResNet-70-16
C12 [25]	Addepalli2021Towards_RN18	ResNet-18
C13 [26]	Cui2023Decoupled_WRN-28-10	WideResNet-28-10
C14 [27]	Xu2023Exploring_WRN-28-10	WideResNet-28-10
C15 [28]	Pang2022Robustness_WRN70_16	WideResNet-70-16
ImageNet		
I1 [22]	Engstrom2019Robustness	ResNet-50
I2 [21]	Salman2020Do_R18	ResNet-18
I3 [21]	Salman2020Do_R50	ResNet-50
I4 [30]	Wong2020Fast	ResNet-50
I5 [19]	Liu2023Comprehensive_Swin-B	Swin-B
I6 [19]	Liu2023Comprehensive_Swin-L	Swin-L
I7 [19]	Liu2023Comprehensive_ConvNeXt-B	ConvNeXt-B
I8 [19]	Liu2023Comprehensive_ConvNeXt-L	ConvNeXt-L

Adversarial setup – We implement the under-confidence and over-confidence attacks by employing the cross-entropy loss between the perturbed input and the targets specified in Section 3. We then leverage the Projected Gradient Descent (PGD) method [12] for crafting the adversarial perturbation. When running the attack, we constrain the l_∞ norm of the adversarial perturbation with a budget ϵ, which we set as $8/255$ for CIFAR-10 and $4/255$ for ImageNet, accordingly to their original RobustBench evaluation. Lastly, we set the number of attack iterations to 150, which is slightly higher compared to standard approaches used in literature for traditional evasion attacks targeting the predictions [13,31]. Moreover, we empirically observed that this is suitable as all example losses reached convergence.

OOD and Open-Set setup – To demonstrate how the analysis of the uncertainty span can be applied to concrete uncertainty attack scenarios, we instantiate our framework in both out-of-distribution (OOD) and Open-Set recognition (OSR) settings. As discussed in Section 3.1 and first proposed in [6], these attacks are designed to disguise samples that are semantically outside the training distribution, making them appear indistinguishable from in-distribution data. To this end, we conduct two types of experiments:

[2] https://www.cs.toronto.edu/~kriz/cifar.HTML
[3] https://www.image-net.org/

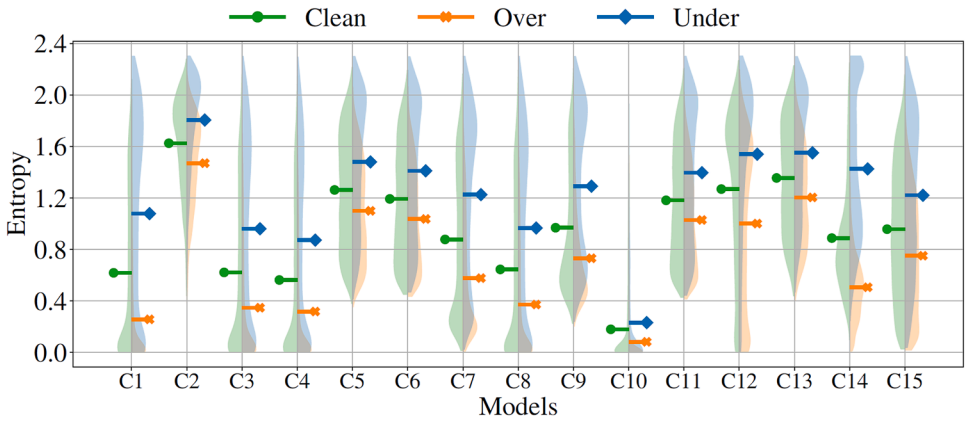


Fig. 2. Entropy distribution before (green on the left) and after the over- (orange on the right) and under-confidence (blue on the right) attacks for all 15 CIFAR-10 robust classifiers, along with a line marking the mean of each distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- OOD scenario: we use 800 samples from the MNIST [32] dataset, which are clearly outside the training distribution;
- OSR scenario: we select 800 samples from 10 novel classes in the CIFAR-100 dataset. These classes are visually similar to the original 10 CIFAR-10 classes but are not part of the training set, thus simulating an open-set scenario [33].

For each scenario, we construct two datasets composed of 8800 samples (8000 in-distribution and 800 outliers): (i) the unperturbed dataset, and (ii) the dataset in which the outliers are perturbed with an over-confidence attack. Finally, we compute the entropy of all the samples in the dataset, which is used to discern in-distribution samples from the outliers.

Evaluation metrics – In our empirical evaluation, we use the MUS and MSUS measures discussed in Section 3. For the calibration case study, however, a standard calibration measure such as the ECE would capture the miscalibration error but not tell if the model is under- or over-confident. Since our study deals with increasing and decreasing the confidence scores of the adversarial samples, it is crucial to use a measure that captures such a phenomenon. Accordingly, we propose a *signed* version of the Expected Calibration Error, which we refer to as s-ECE Eq. (19), by taking the signed subtraction between the expected and the observed probabilities (thus, without the absolute value used for the standard ECE).

$$\text{s-ECE} = \sum_{s=1}^{S} \frac{|\mathbf{B}_s|}{|D|} (ex(\mathbf{B}_s) - ob(\mathbf{B}_s)) \,. \tag{19}$$

By doing so, negative s-ECE values correspond to over-confidence in the models' predictions, whereas positive values correspond to under-confidence.

For the OOD and OSR experiments, we follow common evaluation practices [34,35] by measuring the AUROC, AUPR (IN/OUT), and FPR95TPR.

5.2. Experimental results

This section presents the experimental results of our evaluation, which we first show through our entropy distribution plots and reliability diagrams and then analyze and discuss via the proposed metrics.

Entropy distribution shift – To evaluate the entropy of the models before and after the attacks, we show here the entropy distribution plots, which reveal substantial differences in how the different robust models react to the adversarial attacks. Through Fig. 2, we first depict the entropy distributions of the different models before and after the attacks, highlighting, interestingly, a similar behavior across models.

In C1, C3, C4, and C8 (which we denote with group ◇), the mass of the entropy distribution before the attack is primarily concentrated in low-entropy values, with a mode near ~ 0.1 and a mean of ~ 0.6. The over-confidence attack primarily influences the samples having higher entropies, moving the mass toward the low-entropy mode. The distribution after the under-confidence attack, instead, reveals a bimodal trend: apparently, some of the samples keep lower entropy values, while some others drastically move toward higher entropy regions; such a phenomenon generates two modes on the new distribution, one around ~ 0.1 and one near ~ 1.6. C5, C6, C11, and C13 (group ♣) are characterized by a near-uniform distribution before the attack confined between 0.6 and 1.8. After the attacks, the distributions gain a soft positive (for the over-confidence attack) and negative (for the under-confidence attack) skewness, shifting the mean of about 0.4 from its clean value. Even though we cannot group the remaining methods, some present the behaviors reported above, such as the bimodal distribution for the under-confidence distribution (C7, C9, C14). C2 and C10 are the best-performing models: in C2, the distribution softly shifts when under attack, while C10 remains almost unchanged. Interestingly, while C10 presents very low clean entropy, C2 has one of the highest clean entropies among the models: such behavior reflects their low and high clean accuracies, respectively, reported on the RobustBench leaderboard [13].

Just like CIFAR-10, in ImageNet, we report similarities across models, which allows us to divide them into groups (Fig. 3). The clean entropy distributions of I1, I3, and I4 (group ♡) present a mode of around ~ 0.2; however, a significant part of their mass is also distributed into a considerable interval of medium entropy levels, mainly between 1.0 and 5.0. The over-confidence attacks successfully shift a significant part of the mass onto the mode around ~ 0.2. After the under-confidence attack, similarly, a substantial mass shift, generating a mode around ~ 5.2. The transformer architectures (I5, I6, I7, and I8) (group ♠) seem less sensitive to both attacks, particularly to over-confidence attacks. Indeed, only I5 presents a mode around higher entropy values, so we expect only I6, I7, and I8 to be generally more robust than the remaining classifiers.

Impact on the model's calibration – We now show the effect of the over- and under-confidence attacks on the calibration of adversarial trained models. Figs. 4 and 5 show the reliability diagrams of the 23 robust models for both the analysed data sets, while Tables 2 and 3 report their s-ECE scores.

The majority of the clean reliability curves are positioned above the identity line,[4] which empirically supports the findings of Theorem. 1, i.e., adversarial trained models tend to be under-confident. The "under-confident nature" of adversarial trained models makes them naturally

[4] Note that the spikes in models C1, C3, C10, and C14 do not significantly affect the s-ECE, due to the negligible impact of their small mass on the weighted sum.

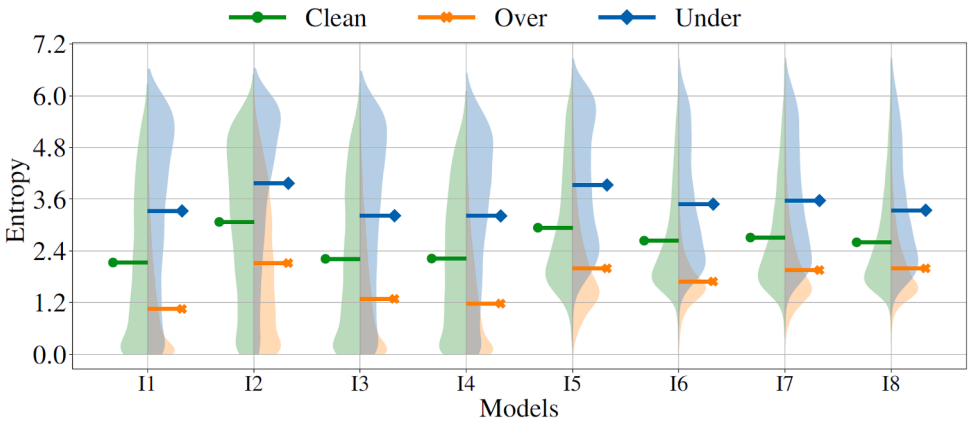


Fig. 3. Entropy distribution before (green on the left) and after the over- (orange on the right) and under-confidence (blue on the right) attacks for all 8 ImageNet robust classifiers, along with a line marking the mean of each distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

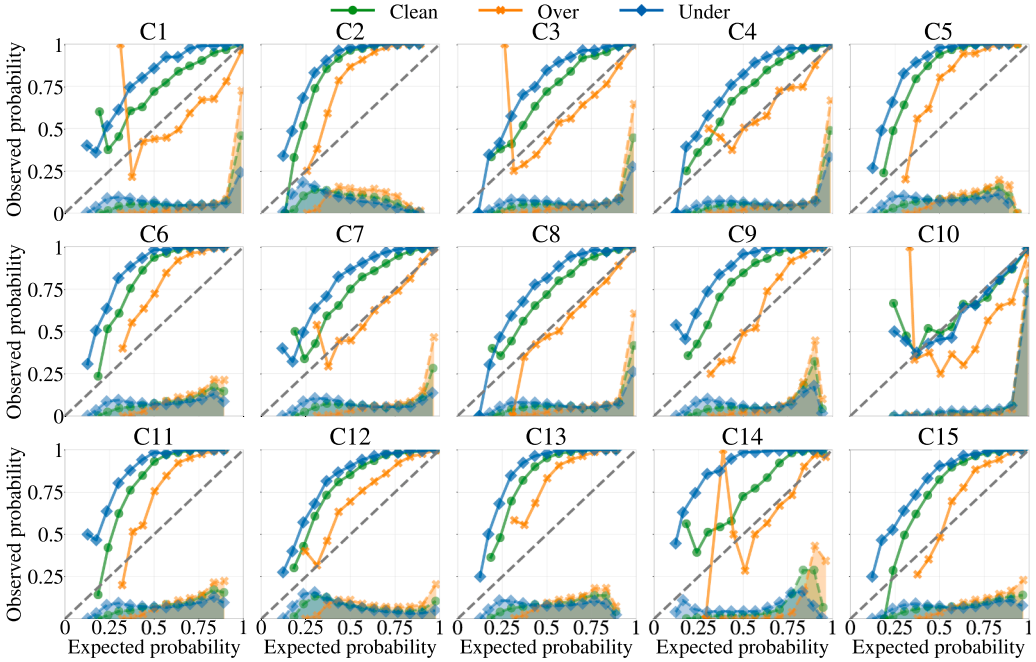


Fig. 4. Calibration curves (above) and confidence histograms (below) for each CIFAR-10 model before (green) and after the over- (orange) and under-confidence (blue) attacks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

robust against over-confidence attacks (in line with our conjecture in Section 4): indeed, as we can see from Tables 2 and 3, even after the attack only a few CIFAR-10 models and half of the ImageNet models reach negative s-ECE values (i.e., become over-confident).

Discussion – In Table 4 we present a leaderboard ranking models based on their robustness against uncertainty attacks, sorted by MUS. For each of the reported values we also show the standard deviation computed on the MUS and MSUS distributions, for strengthen the statistical significance of the results.

Here, the results reflect well our analysis: the group ♢ comes after group ♣; the same applies for ImageNet, where almost all models from group ♡ come after, except I5, the transformers (group ♠). Generally speaking - except for some outliers - we observe a slight correlation between robust accuracy and robust uncertainty: disregarding C10 and C2, group ♣ has (on average) lower MUS and MSUS than group ♢, while the uncategorized models perform similarly to ♣; at the same time - without taking I5 into account - the Transformers outperform the ResNet architectures. Nevertheless, the presence of C2 in the second place reveals that high entropic models may eventually disclose general robust properties against uncertainty attacks.

It is also important to note that Table 4 reports the rankings obtained by RobustBench ordered by the best known robust accuracy. Most of these robust accuracies are obtained by using AutoAttack, which is an ensemble of gradient-based white- and black-box attacks. However, for certain models, even AutoAttack reveals to be unreliable: this is the case of C10, for which the best known robust accuracy was obtained by running a transfer attack. In fact, methods that make use of Neural ODE in their architecture like C10 are known to be affected by obfuscated gradients [23,36]. This phenomenon hinders the effectiveness of gradient-based attacks, leading to unreliable robustness evaluations [37]. Accordingly, since our evaluation consists of gradient-based attacks, we mark the results obtained with C10 as unreliable, noting that our result may change by employing different attack strategies, e.g., attack transferability, as suggested in [23,36].

OOD and Open-Set results – Table 5 reports the results of the experiments for the OOD and OSR scenarios.

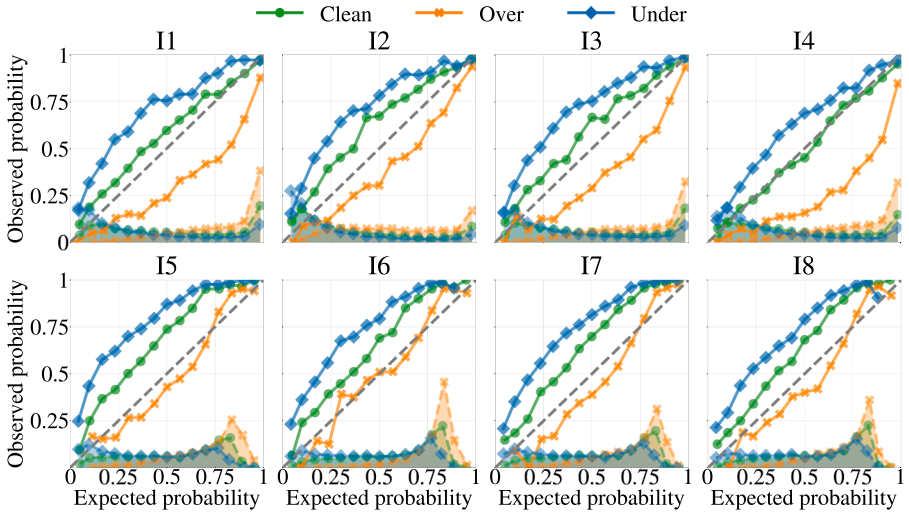


Fig. 5. Calibration curves (above) and confidence histograms (below) for each ImageNet model before (green) and after the over- (orange) and under-confidence (blue) attacks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
s-ECE before and after the over- and under-confidence attacks for each CIFAR-10 model.

s-ECE	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
Under	0.22	0.44	0.19	0.17	0.37	0.39	0.25	0.19	0.30	-0.01	0.34	0.33	0.39	0.33	0.26
Clean	0.09	0.39	0.10	0.09	0.29	0.27	0.14	0.11	0.19	-0.01	0.27	0.26	0.32	0.15	0.20
Over	-0.06	0.29	-0.02	-0.01	0.21	0.20	0.00	-0.01	0.09	-0.05	0.19	0.13	0.24	0.04	0.10

Table 3
s-ECE before and after the over- and under-confidence attacks for each ImageNet model.

s-ECE	I1	I2	I3	I4	I5	I6	I7	I8
Under	0.20	0.20	0.20	0.11	0.31	0.27	0.27	0.24
Clean	0.06	0.11	0.08	-0.01	0.19	0.16	0.17	0.16
Over	-0.20	-0.12	-0.15	-0.28	0.01	0.07	0.01	0.02

As we can see from the results, all the considered metrics deteriorate in the presence of an attacker. Nevertheless, even though such deterioration can vary on the base of the considered model, we notice that it does not lead to a complete break, as it happens for undefended models, where even for small perturbation budget it is possible to make the uncertainty measure on the OOD lower than any ID sample [7]; this is a clear indicator of robustness for adversarially trained models against OOD and Open Set uncertainty attacks. Quantitatively, we also notice that in almost all the cases, the adversarially trained models are able to discriminate better between open set samples compared to OOD samples. The aleatoric nature of the considered uncertainty measure can be a possible explanation of this phenomenon. Indeed, we remark that the entropy is an aleatoric measure, which is based upon the output prediction vector; notably, OOD represents an emblematic case of epistemic uncertainty, because, by definition, the considered model has no prior knowledge of the samples which are far from the training distribution. Instead, for the open set case, the lack of knowledge is less evident: even though the selected classes from CIFAR-100 were not present in the training data, some visual features may possibly be shared across classes (e.g., features from the “helicopter” class may be present in some CIFAR-10 veichle classes su as the “airplane” class).

Interestingly, we notice that C4 and C14 are more sensitive to the open set attacks compared to the remaining ones. In particular, C14 suffers from an extreme deterioration under attack in the open set case (e.g., AUROC from 0.840 ot 0.348), which is in line with the ID leaderboard results in Table 4.

5.3. Robustness of uncertainty quantification for semantic segmentation

We now demonstrate that our analysis is not limited to classification tasks or architectures designed explicitly for classification, but also extends to different and more complex scenarios. To this end, we analyze the robustness of adversarially trained segmentation models against uncertainty-based attacks. This type of task plays a pivotal role in numerous safety-critical applications, ranging from organ segmentation in CT scans [3] to visual perception in autonomous driving [2].

Semantic Segmentation, Uncertainty and Attacks – Semantic segmentation can be viewed as a multivariate version of classification, where the goal is to assign a class label to each pixel of an input image. In this setting, uncertainty is modeled at the pixel level, producing an uncertainty map that reflects the confidence of the segmentation at each location. The pixel-wise nature of the task reveals properties that can be used to produce more effective uncertainty attacks compared to the standard formulation in Eq. (9). Specifically, as shown in previous literature [7], one can leverage the spatial consistency between neighboring pixels to generate adversarial examples that minimize total uncertainty. This attack, known as Uniform Segmentation Target (UST), consists of performing a targeted attack using a uniform segmentation map, i.e., one where all pixels are assigned the same class as the target. The choice of the target class typically depends on how prominently a given class is represented in the image.

Experiemental Setup – To observe the effect of adversarial training against uncertainty attacks, we selected one robust (UperNext-ConvNext_T from Croce et al. [38]) and one non-robust (FCN [39] from the torchvision library[5]) semantic segmentation model, both trained on the Pascal-VOC dataset.[6] We replicate the setup of Section 5.1, using $\epsilon = 8/255$ in l_∞ norm, and a PGD with 100 iterations. The under-confidence attack is implemented analogously to Eq. (11), while for the

[5] https://docs.pytorch.org/vision/stable/index.HTML
[6] http://host.robots.ox.ac.uk/pascal/VOC/

Table 4
Mean (μ) and standard deviation (σ) of the **MUS** and **MSUS** of each CIFAR-10 and ImageNet model (sorted by **MUS**†), along with their accuracies and RobustBench-based rank. Denoted with * we report potentially unreliable values, as discussed in Section 5.

			Accuracy		$MUS^\dagger$		MSUS		
	Robustbench ID	Model ID	Clean	Robust	$\mu^\dagger$	σ	μ	σ	RB rank
CIFAR-10	Kang2021Stable	C10 [23]	93.73 %	64.20 %	0.149*	0.267*	0.094*	0.260*	7
	Addepalli2022Efficient_RN18	C2 [16]	85.71 %	52.48 %	0.335	0.120	0.127	0.096	13
	Cui2023Decoupled_WRN-28-10 ♣	C13 [26]	92.16 %	67.73 %	0.347	0.168	0.148	0.152	3
	Peng2023Robust ♣	C11 [24]	93.27 %	71.07 %	0.368	0.208	0.178	0.200	1
	Wang2023Better_WRN-70-16 ♣	C6 [29]	93.25 %	70.69 %	0.374	0.208	0.183	0.201	2
	Wang2023Better_WRN-28-10 ♣	C5 [29]	92.44 %	67.31 %	0.380	0.192	0.181	0.188	4
	Pang2022Robustness_WRN70_16 ◇′	C15 [28]	89.01 %	63.35 %	0.470	0.221	0.270	0.248	10
	Addepalli2021Towards_RN18	C12 [25]	80.24 %	51.06 %	0.539	0.293	0.376	0.398	14
	Gowal2021Improving_70_16_ddpm_100m ◇	C4 [17]	88.74 %	66.10 %	0.555	0.468	0.527	0.706	6
	Rebuffi2021Fixing_70_16_cutmix_extra	C9 [18]	92.23 %	66.56 %	0.560	0.339	0.428	0.441	5
	Sehwag2021Proxy_ResNest152 ◇	C8 [20]	87.30 %	62.79 %	0.594	0.443	0.550	0.660	11
	Gowal2021Improving_28_10_ddpm_100m ◇	C3 [17]	87.50 %	63.38 %	0.614	0.470	0.597	0.738	9
	Sehwag2021Proxy_R18	C7 [20]	84.59 %	55.54 %	0.649	0.394	0.576	0.601	12
	Engstrom2019Robustness ◇	C1 [22]	87.03 %	49.25 %	0.821	0.579	1.009	1.063	15
	Xu2023Exploring_WRN-28-10	C14 [27]	93.69 %	63.89 %	0.920	0.551	1.149	1.216	8
ImageNet	Liu2023Comprehensive_ConvNeXt-L ♠	I8 [19]	78.02 %	58.48 %	1.348	0.936	2.692	3.558	2
	Liu2023Comprehensive_ConvNeXt-B ♠	I7 [19]	76.02 %	55.82 %	1.613	1.020	3.640	4.311	4
	Liu2023Comprehensive_Swin-L ♠	I6 [19]	78.92 %	59.56 %	1.797	1.304	4.930	6.075	1
	Salman2020Do_R18	I2 [21]	52.92 %	25.32 %	1.850	1.070	4.568	4.910	8
	Salman2020Do_R50 ♡	I3 [21]	64.02 %	34.96 %	1.928	1.274	5.341	5.998	5
	Liu2023Comprehensive_Swin-B ♠′	I5 [19]	76.16 %	56.16 %	1.934	1.062	4.867	5.064	3
	Wong2020Fast ♡	I4 [30]	55.62 %	26.24 %	2.033	1.243	5.679	5.862	7
	Engstrom2019Robustness ♡	I1 [22]	62.56 %	29.22 %	2.272	1.477	7.346	7.700	6

Table 5
AUROC, AUPR and FPR95TPR computed on a mixture of ID-OSR (on the left) and ID-OOD (on the right) samples, for six Robust WideResNets, before and after the under-confidence attacks on the OOD and OSR subsets.

		OSR				OOD			
Model ID	ϵ	AUROC	AUPR-IN	AUPR-OUT	FPR95TPR	AUROC	AUPR-IN	AUPR-OUT	FPR95TPR
C9	clean	0.818	0.977	0.296	0.589	0.724	0.963	0.189	0.710
C9	4/255	0.741	0.966	0.175	0.695	0.681	0.956	0.151	0.770
C9	8/255	0.642	0.951	0.116	0.782	0.635	0.948	0.122	0.816
C5	clean	0.823	0.977	0.318	0.596	0.668	0.957	0.124	0.729
C5	4/255	0.750	0.967	0.191	0.687	0.635	0.951	0.112	0.772
C5	8/255	0.655	0.953	0.123	0.758	0.599	0.944	0.102	0.805
C13	clean	0.816	0.976	0.298	0.602	0.602	0.943	0.104	0.823
C13	4/255	0.744	0.966	0.182	0.699	0.572	0.937	0.096	0.858
C13	8/255	0.651	0.952	0.120	0.758	0.541	0.930	0.090	0.885
C6	clean	0.836	0.979	0.334	0.571	0.651	0.952	0.118	0.793
C6	4/255	0.767	0.969	0.204	0.666	0.621	0.946	0.109	0.828
C6	8/255	0.674	0.956	0.130	0.729	0.589	0.940	0.101	0.858
C4	clean	0.830	0.980	0.296	0.513	0.777	0.974	0.200	0.530
C4	4/255	0.764	0.971	0.186	0.607	0.762	0.972	0.184	0.549
C4	8/255	0.676	0.958	0.127	0.693	0.745	0.969	0.170	0.570
C14	clean	0.840	0.975	0.391	0.695	0.943	0.994	0.614	0.206
C14	4/255	0.587	0.915	0.116	0.989	0.930	0.993	0.542	0.244
C14	8/255	0.348	0.851	0.065	1.000	0.914	0.991	0.465	0.274

over-confidence case, we considered an ensemble of UST and STAB; this choice is crucial to ensure the success of the over-confidence attack in cases where UST alone fails, as we will discuss in the following section.
Results and Discussion – For simplicity, we conduct our analysis using the average pixel-wise uncertainty for each image. By doing so, we can follow an approach similar to that used for classification: analyzing the uncertainty distribution before and after the attacks, and measuring the models' uncertainty spans.

Fig. 6 shows the distribution of the uncertainty spans and the box plots of the uncertainty over the datasets. The US distribution plot reveals clear differences in how the two models respond to uncertainty attacks: the US distribution of the non-robust model is completely shifted toward the highest possible values (with an observed MUS of 2.64), while that of the robust model remains confined to significantly lower values (with a MUS of 0.25). The blue box plots confirm that over-confidence attacks reduce the uncertainty to zero, while under-confidence attacks push almost the entire dataset above a value of 2.5 (recall that the maximum for this dataset is approximately 3.04[7]). On the contrary, albeit not being entirely robust against these attacks, the adversarially trained model's uncertainty have been revealed to be subject to manipulation. Indeed, the red box plots indicate that such attacks can still partially affect robust models.

We finally show some qualitative examples of the uncertainty attacks in Fig. 7.

[7] The Pascal-VOC dataset comprises 21 classes, including the background class: hence, the maximum entropy is $\log(21) \approx 3.04$.

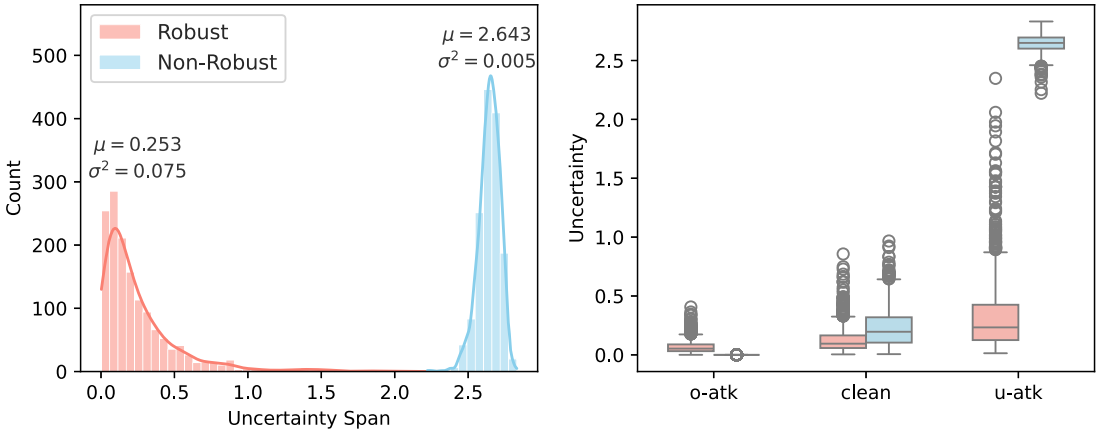


Fig. 6. On the left, the uncertainty span distribution of the robust and non-robust models, along with the mean and variance of the distributions; on the right, the box plots of the uncertainties of the robust and non-robust models.

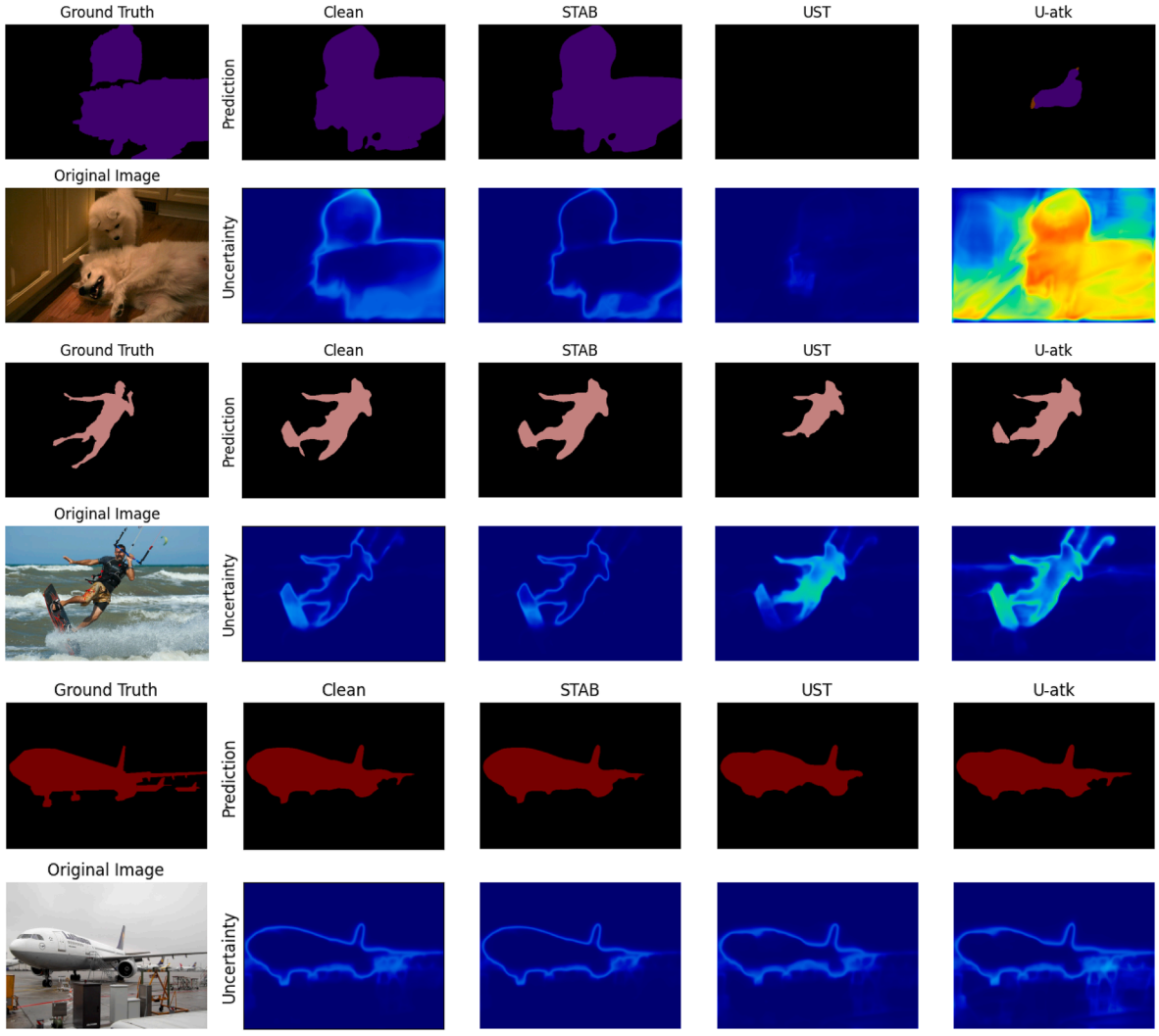


Fig. 7. Three examples of the Pascal-VOC dataset, each one composed by an image, a ground truth map and 4 predictions (clean, stab, ust and u-atk). For each prediction we show the predicted segmentation (on top) and the predicted uncertainty map (on the bottom).

Sometimes, some samples may be revealed to be sensitive to under-confidence attacks even when the prediction is not affected. Here, it is also interesting to note the differences between STAB and UST. From previous literature, it is known that STAB has the effect of minimizing the overall uncertainty in regions where the predicted segmentation is uniform, while keeping the uncertainty high on segmented class boundaries. The same effect is also observed in our tests against the robust models. Nevertheless, the same does not generally hold for the UST attack: when the attack does not manage to flip the labels, the effect obtained is the opposite of the intended one, leading to higher uncertainty. This means that UST in isolation might not guarantee a proper reduction, and to ensure an effective O-atk against adversarially trained

models, UST should be used in combination with other non-spatial attacks such as STAB. Finally, the third example also shows cases where the model successfully defends against the attacks, showing an almost completely unaltered uncertainty map for both the under- and over-confidence attacks.

6. Conclusions, limitations and future works

In this work, we provided an extensive analysis of the robustness of adversarial-trained models against uncertainty attacks. Our experimental evaluation conducted on image classification benchmarks has empirically validated our hypothesis that instead of needing an ad hoc defense strategy for uncertainty attacks, one can rely on already trained state-of-the-art robust models. Nevertheless, we point out some limitations of our work, which we consider promising starting points for future research. First, while our theoretical analysis reveals the interesting behaviour of adversarial trained models being underconfident by nature, this is still limited to 2-class cases and assumes the model finds a theoretical equilibrium. Nevertheless, as happens often in Machine Learning [12], it is important to notice that all the discussed theoretical results should be taken not as an ultimate theory, but rather as a compass that can guide the research process toward conclusions which are interesting from an engineering perpective, even if the theoretical assumptions do not perfectly reflect the complexity of the real world. Second, the attack setup that is shared across all considered models is not intended to be exhaustive from the adversarial perspective. For instance, we discussed in Section 5.2 how [23] needs further investigation using more advanced techniques to properly assess its robustness [23,36,37]. To this end, our proposed leaderboard should not be used as the ultimate benchmark, but rather as a general compass that offers valuable insights on the uncertainty robustness capabilities of adversarial trained models. Third, we prioritized our analysis to the sole aleatoric uncertainty, as previous research on adversarial robustness led to a plethora of pre-trained robust models that are deterministic by design, hence with no epistemic uncertainty available [13]. Accordingly, as future research, we intend to extend such analysis to epistemic uncertainty as well, particularly for proper Bayesian models, by employing seminal promising solutions that integrate adversarial training on top of Bayesian Neural Networks [40]. On a closing note, we believe that our study represents a first step toward understanding the robustness of standard benchmark models under adversarial perturbations targeting uncertainty quantification and their impact on uncertainty estimates. The insights obtained open up promising directions for transferring these findings to emerging areas in AI, such as generative and foundation models, as well as to safety-critical domains like autonomous driving and healthcare.

CRediT authorship contribution statement

Emanuele Ledda: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Giovanni Scodeller:** Writing – original draft, Visualization, Validation, Software, Investigation, Conceptualization; **Daniele Angioni:** Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization; **Giorgio Piras:** Writing – review & editing, Writing – original draft, Investigation, Conceptualization; **Antonio Emanuele Cinà:** Writing – review & editing, Supervision, Investigation, Conceptualization; **Giorgio Fumera:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Battista Biggio:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization; **Fabio Roli:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization.

Data availability

The code is accessible through the link on the paper

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been supported by the European Union's Horizon Europe research and innovation program under the projects ELSA (grant no. 101070617) and CoEvolution (grant no. 101168560); by Fondazione di Sardegna under the project "TrustML: Towards Machine Learning that Humans Can Trust", CUP: F73C22001320007; by EU-NGEU National Sustainable Mobility Center (CN00000023) Italian MUR Decree n. 1033-17/06/2022 (Spoke 10); and by projects SERICS (PE00000014) and FAIR (PE00000013) under the NRRP MUR program funded by the EU-NGEU. This work was conducted while Emanuele Ledda, Daniele Angioni, and Giorgio Piras were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with the University of Cagliari.

References

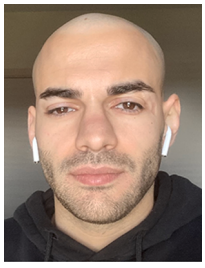
[1] E. Hüllermeier, W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, Mach. Learn. 110 (3) (2021) 457–506.
[2] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, A. Weller, Concrete problems for autonomous vehicle safety: advantages of Bayesian deep learning, in: IJCAI, 2017, pp. 4745–4753.
[3] X. Guo, X. Lin, X. Yang, L. Yu, K. Cheng, Z. Yan, UCTNet: uncertainty-guided CNN-transformer hybrid networks for medical image segmentation, Pattern Recognit. 152 (2024) 110491.
[4] J. Wei, G. Wang, S. Zhang, Fine-grained medical image out-of-distribution detection through multi-view feature uncertainty and adversarial sample generation, Pattern Recognit. 162 (2025) 111401.
[5] I. Galil, R. El-Yaniv, Disrupting deep uncertainty estimation without harming accuracy, in: Advances in Neural Information Processing Systems, 34, 2021, pp. 21285–21296.
[6] H. Zeng, Z. Yue, Y. Zhang, Z. Kou, L. Shang, D. Wang, On attacking out-domain uncertainty estimation in deep neural networks, in: IJCAI, 2022, pp. 4893–4899.
[7] E. Ledda, D. Angioni, G. Piras, G. Fumera, B. Biggio, F. Roli, Adversarial attacks against uncertainty quantification, in: IEEE/CVF Int. Conf. on Computer Vision Workshops, 2023, pp. 4599–4608.
[8] S. Obadinma, X. Zhu, H. Guo, Calibration attack: a framework for adversarial attacks targeting calibration, arXiv:2401.02718 (2024).
[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I.J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: ICLR, 2014.
[10] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: ECML/PKDD, 8190, Springer, 2013, pp. 387–402.
[11] B. Biggio, F. Roli, Wild patterns: ten years after the rise of adversarial machine learning, Pattern Recognit. 84 (2018) 317–331.
[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: ICLR (Poster), OpenReview.net, 2018.
[13] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, M. Hein, RobustBench: a standardized adversarial robustness benchmark, in: NeurIPS Datasets and Benchmarks, 1, 2021.
[14] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: Int. Conf. on Machine Learning, PMLR, 2017, pp. 1321–1330.
[15] J. Blasiok, P. Gopalan, L. Hu, P. Nakkiran, When does optimizing a proper loss yield calibration?, in: Advances in Neural Information Processing Systems, NeurIPS, 2023.
[16] S. Addepalli, S. Jain, et al., Efficient and effective augmentation strategy for adversarial training, Adv. Neural Inf. Process. Syst. 35 (2022) 1488–1501.
[17] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D.A. Calian, T.A. Mann, Improving robustness using generated data, Adv. Neural Inf. Process. Syst. 34 (2021) 4218–4233.
[18] S.-A. Rebuffi, S. Gowal, D.A. Calian, F. Stimberg, O. Wiles, T. Mann, Fixing data augmentation to improve adversarial robustness, arXiv:2103.01946 (2021).
[19] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, S. Zheng, A comprehensive study on robustness of image classification models: benchmarking and rethinking, Int. J. Comput. Vis. 133 (2) (2025) 567–589.
[20] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, P. Mittal, Robust learning meets generative models: can proxy distributions improve adversarial robustness?, in: ICLR, OpenReview.net, 2022.

[21] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, A. Madry, Do adversarially robust imagenet models transfer better?, Adv. Neural Inf. Process. Syst. 33 (2020) 3533–3545.
[22] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, D. Tsipras, Robustness (python library), 2019, https://github.com/MadryLab/robustness.
[23] Q. Kang, Y. Song, Q. Ding, W.P. Tay, Stable neural ode with Lyapunov-stable equilibrium points for defending against adversarial attacks, Adv. Neural Inf. Process. Syst. 34 (2021) 14925–14937.
[24] S. Peng, W. Xu, C. Cornelius, M. Hull, K. Li, R. Duggal, M. Phute, J. Martin, D.H. Chau, Robust principles: architectural design principles for adversarially robust CNNs, in: BMVC, BMVA Press, 2023, pp. 739–740.
[25] S. Addepalli, S. Jain, G. Sriramanan, R. Venkatesh Babu, Scaling adversarial training to large perturbation bounds, in: European Conf. on Computer Vision, Springer, 2022, pp. 301–316.
[26] J. Cui, Z. Tian, Z. Zhong, X. Qi, B. Yu, H. Zhang, Decoupled Kullback-Leibler divergence loss, in: NeurIPS, 2024.
[27] Y. Xu, Y. Sun, M. Goldblum, T. Goldstein, F. Huang, Exploring and exploiting decision boundary dynamics for adversarial robustness, in: ICLR, OpenReview.net, 2023.
[28] T. Pang, M. Lin, X. Yang, J. Zhu, S. Yan, Robustness and accuracy could be reconcilable by (proper) definition, in: Int. Conf. on Machine Learning, PMLR, 2022, pp. 17258–17277.
[29] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, S. Yan, Better diffusion models further improve adversarial training, in: Int. Conf. on Machine Learning, PMLR, 2023, pp. 36246–36263.
[30] E. Wong, L. Rice, J.Z. Kolter, Fast is better than free: revisiting adversarial training, in: ICLR, OpenReview.net, 2020.
[31] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: ICML, 119 of *Proceedings of machine learning research*, PMLR, 2020, pp. 2206–2216.
[32] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. 29 (6) (2012) 141–142.
[33] W.J. Scheirer, A. de Rezen, A. Sapkota, T.E. Boult, Toward open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1757–1772.
[34] J. Yang, K. Zhou, Y. Li, Z. Liu, Generalized out-of-distribution detection: a survey, Int. J. Comput. Vis. 132 (12) (2024) 5635–5662.
[35] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun, X. Du, K. Zhou, W. Zhang, D. Hendrycks, Y. Li, Z. Liu, OpenOOD: benchmarking generalized out-of-distribution detection, in: NeurIPS, 2022.
[36] Y. Huang, Y. Yu, H. Zhang, Y. Ma, Y. Yao, Adversarial robustness of stabilized neural ODE might be from obfuscated gradients, in: MSML, 145 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 497–515.
[37] A. Athalye, N. Carlini, D.A. Wagner, Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, in: ICML, 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 274–283.
[38] F. Croce, N.D. Singh, M. Hein, Towards reliable evaluation and fast training of robust semantic segmentation models, in: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXIX, 15137 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 180–197.
[39] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, IEEE Computer Society, 2015, pp. 3431–3440.
[40] G. Carbone, M. Wicker, L. Laurenti, A. Patané, L. Bortolussi, G. Sanguinetti, Robustness of Bayesian neural networks to gradient-based attacks, in: NeurIPS, 2020.

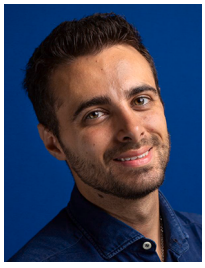
Emanuele Ledda is a Postdoctoral Researcher at CINI, "Consorzio Interuniversitario Nazionale per l'Informatica", Italy. In January 2025 he received his PhD in "Artificial Intelligence" from the University of Roma La Sapienza. His research interests include uncertainty quantification and adversarial machine learning, with applications to computer vision and intelligent video surveillance. He serves as a reviewer for top-tier journals, including Pattern Recognition and Information Sciences.

Giovanni Scodeller was born in Venice in 1997. He studied at Ca' Foscari University of Venice where he received his BSc in "computer science - data science" in 2020, same university where he received his MSc in "computer science - data management and analytics" in 2023. He is now a PhD. student in Cybersecurity and Reliable AI at the University of Genoa, focusing on uncertainty quantification and security in machine learning.

Daniele Angioni is a Postdoctoral Researcher at the University of Cagliari, Italy. He received his PhD in Artificial Intelligence in January 2025 from the Sapienza University of Rome. His research addresses machine learning security in the real world, applied to both malware and image domains. He serves as a reviewer for top-tier journals, including IEEE TIFS and Pattern Recognition.



Giorgio Piras is a Postdoctoral Researcher at the University of Cagliari, Italy. He received is BSc degree in Electrical, Electronic and Computer Engineering from the University of Cagliari in 2019. He then received, from the same university, his MSc in Computer Engineering in 2021, with honors. In January 2025 received, with honors, his PhD in "Artificial Intelligence" from the University of Roma La Sapienza. His research interests include adversarial machine learning, explainability methods, and neural network compression techniques.



Antonio Emanuele Cinà (MSc 2019, PhD 2023) is an Assistant Professor at the University of Genoa, Italy. His work sits at the intersection of artificial intelligence and security, with core expertise in training-time (poisoning) and inference-time (evasion) attacks. Antonio's research aims to uncover and address the risks that undermine the availability, reliability, and trustworthiness of modern AI.

Giorgio Fumera (MSc 1997, PhD 2002) is an Associate Professor of Computer Engineering of the University of Cagliari, and a member of the PRA Lab since 1999. His research interests are related to statistical pattern recognition and machine learning, including ensemble methods, uncertainty quantification, and adversarial machine learning, with applications to computer vision for intelligent video surveillance. He is an Associate Editor for the Pattern Recognition journal (Elsevier) and the Pattern Analysis and Applications journal (Springer).

Battista Biggio (MSc 2006, PhD 2010) is Full Professor of Computer Engineering at the University of Cagliari, Italy. He has provided pioneering contributions to machine learning security. His paper "Poisoning Attacks against Support Vector Machines" won the prestigious 2022 ICML Test of Time Award. He chaired IAPR TC1 (2016–2020), and served as Associate Editor for IEEE TNNLS and IEEE CIM. He is now Associate Editor-in-Chief for Pattern Recognition and serves as Area Chair for NeurIPS and IEEE Symp. SP. He is Fellow of IEEE and AAIA, ACM Senior Member, and Member of IAPR, AAAI, and ELLIS.

Fabio Roli is a Full Professor of Computer Science at the University of Genoa, Italy. He has been appointed Fellow of the IEEE and Fellow of the International Association for Pattern Recognition. He is a recipient of the Pierre Devijver Award for his contributions to statistical pattern recognition.