The Best Defense is a Good Offense: Adversarial Augmentation against Adversarial Attacks

Iuri Frosio, NVIDIA
ifrosio@nvidia.com

Jan Kautz, NVIDIA
jkautz@nvidia.com

Abstract

*Many defenses against adversarial attacks (*e.g. *robust classifiers, randomization, or image purification) use countermeasures put to work only after the attack has been crafted. We adopt a different perspective to introduce A^5 (Adversarial Augmentation Against Adversarial Attacks), a novel framework including the first certified preemptive defense against adversarial attacks. The main idea is to craft a defensive perturbation to guarantee that any attack (up to a given magnitude) towards the input in hand will fail. To this aim, we leverage existing automatic perturbation analysis tools for neural networks. We study the conditions to apply A^5 effectively, analyze the importance of the robustness of the to-be-defended classifier, and inspect the appearance of the robustified images. We show effective on-the-fly defensive augmentation with a robustifier network that ignores the ground truth label, and demonstrate the benefits of robustifier and classifier co-training. In our tests, A^5 consistently beats state of the art certified defenses on MNIST, CIFAR10, FashionMNIST and Tinyimagenet. We also show how to apply A^5 to create certifiably robust physical objects. Our code at* https://github.com/NVlabs/A5 *allows experimenting on a wide range of scenarios beyond the man-in-the-middle attack tested here, including the case of physical attacks.*

1. Introduction

Since Deep Neural Networks (DNNs) have been found vulnerable to adversarial attacks [11, 30], researchers studied various protection strategies [4, 12, 20, 42, 43]. For instance, *adversarial training* [11, 30] generates attacks while asking a DNN for the correct output in training; it is simple, partially effective and widely adopted. Certified methods (*e.g.*, IBP [12], CROWN [43], CROWN-IBP [42]) do a step more by estimating correct (although often pessimistic) output bounds (Fig. 1, a) used for training. Adversarial training regularizes the classification landscape against the attacks (Fig. 1, b), but high protection often produces a loss in clean accuracy. Other partially effective defenses are based on randomness [35, 35] or removal of the adversarial signal [13, 18, 27, 40], by moving the input back to the space of natural (non-attacked) data before classification (Fig. 1, c).

All the aforementioned strategies activate the defense mechanism only *after* the attack has been crafted. However, when dealing with adversarial attacks, *the first actor to move has a significant advantage*. For instance, perturbing an image can avoid online person identification and preserve privacy [6, 19, 26]. Acting first is particularly suitable against Man in the Middle (MitM) attacks that may practi-

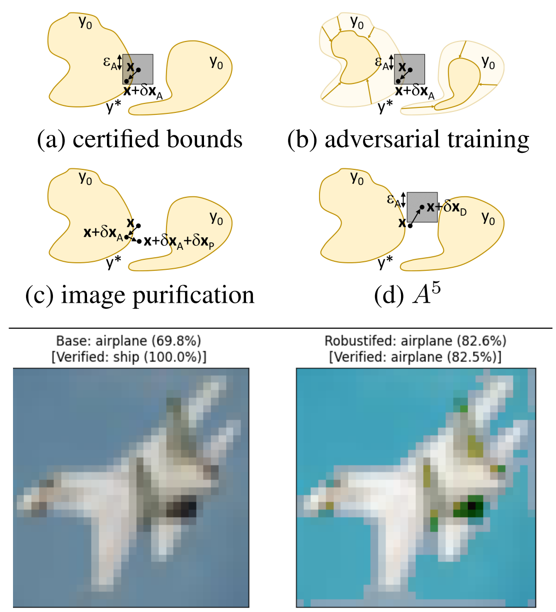

Figure 1. (a) An input $\boldsymbol{x} \in y*$ is correctly classified. The grey box (certified bounds [12, 42, 43]) shows that, under an attack $\delta\boldsymbol{x}_A$, $||\delta\boldsymbol{x}_A||_\infty < \epsilon_A$, misclassification ($(\boldsymbol{x} + \delta\boldsymbol{x}_A) \in y_0$) is possible. (b) Adversarial training [4, 20] creates robust DNNs with regular classification landscapes: misclassification is less likely. (c) Image purification [27, 40] moves the attacked input $\boldsymbol{x} + \delta\boldsymbol{x}_A$ by $\delta\boldsymbol{x}_P$ back to correct classification. (d) A^5 preemptively moves $\boldsymbol{x}$ into a non attackable position. The original CIFAR10 image classified as *airplane* (69.8% confidence) can be misclassified under attack (*ship*, 100% confidence). Once robustified through A^5 (right), misclassification does not occur anymore.

$\boldsymbol{v}$	A vector	v_j	Vector j-th element
C	Classifier	R	Robustifier
$\boldsymbol{x}$	Data	$\delta\boldsymbol{x}_A$	Attacking perturbation
$\delta\boldsymbol{x}_D$	Defensive perturbation	$\boldsymbol{x}+\delta\boldsymbol{x}_A$	Data under attack
$\boldsymbol{x}+\delta\boldsymbol{x}_D$	Robustified data	$\boldsymbol{x}+\delta\boldsymbol{x}_D+\delta\boldsymbol{x}_A$	Robustified data under attack
ϵ_D	Defense magnitude, $\|\|\delta\boldsymbol{x}_D\|\|_p < \epsilon_D$	ϵ_A^R	Attack magnitude, $\|\|\delta\boldsymbol{x}_A\|\|_p < \epsilon_A^R$ while training R
ϵ_A	Attack magnitude $\|\|\delta\boldsymbol{x}_A\|\|_p < \epsilon_A$ while testing	ϵ_A^C	Attack magnitude, $\|\|\delta\boldsymbol{x}_A\|\|_p < \epsilon_A^C$ while training C

Table 1. Notation for data $\boldsymbol{x}$, that are the inputs of the classifier C. We adopt an equivalent notation for physical objects $\boldsymbol{w}$.

cally arise in automotive [32], audio processing [13, 22, 38], or while communicating with a remote machine [39]. Our idea spouses this reasoning line: we investigate a novel way to augment the data to preemptively *certify* that it cannot be attacked (Fig. 1, d). This fits in a recent research area that up to now has received less attention than adversarial training. Researchers explored image and real object augmentation to guarantee a high recognition rate in presence of noise or geometric distortions, but not in adversarial scenarios [24]. Encryption schemes coupled with deep learning [1, 28] or watermarking [16] only partially protect against MitM attacks. The few existing preemptive robustification methods [23] include a procedure [17] that first runs a classifier on clean data and achieves preemptive robustification against MitM attacks through iterative optimization; these are partially effective and do not provide any certification. Our novel framework encompasses most of the aforementioned cases while also introducing for the first time the concept of certified robustification. More in detail, our manifold contributions are:

(i) We introduce A^5 (Adversarial Augmentation Against Adversarial Attacks), a comprehensive framework for preemptive augmentation to make data and physical objects *certifiably* robust against MitM adversarial attacks. As far as we know, this is the first time certified defense is achieved in this way. Since we provide certified robustness, we guarantee protection against any form of white, grey or black box attack.

(ii) We test different flavours of A^5 on standard datasets. By doing so, we study the connection between the robustness of the legacy classifier, the magnitude of the defensive augmentation, and the protection level delivered by A^5. We show A^5 achieving state-of-the-art certified protection against MitM attacks by training a robustifier DNN coupled with the legacy classifier, and even better results for co-training of the robustifier and classifier. We perform a critical, visual inspection of the robustified images to answer an interesting theoretical question (*how does a non-attackable image look like?*) and potentially provide directions for the acquisition of inherently robust images.

(iii) Using Optical Character Recognition (OCR) as an example, we show the application of A^5 for the design of certifiably robust physical objects, which extends [24] to the case of certified defense against MitM attacks.

(iv) We share our code at https://github.com/NVlabs/A5, to allow replicating our results or testing A^5 in scenarios not considered here, for instance on other datasets or for protection against physical adversarial attack (*e.g.*, adversarial patches).

2. Related Work and Background

We identify three main philosophies for the development of defenses against adversarial attacks. They are not mutually exclusive and can be adopted together.

The first approach aims at creating DNNs that are robust to adversarial attacks: *e.g.*, in adversarial training, attacks are generated while training the DNN that should process them correctly [4, 11, 20]. This regularizes the classification landscape and makes attacks less effective (Fig. 1, b). Adversarial training is simple but has one drawback: as attack generation is computationally demanding (even after speed up [25]), attacks are randomly sampled and cannot cover the entire input space, thus impacting the final accuracy. Certified defense [12, 42, 43] can be seen as a state of the art refined form of adversarial training: robustness is obtained by propagating through the DNN an input interval that encompass all the attacks with magnitude up to ϵ_A^C. The corresponding set of output bounds allows estimating the worst case scenario and derive the training cost function (*e.g.*, worst case entropy [12, 42]). Research focuses on finding bounds that are both tight and computationally light. IBP [12] computes (and backwards) the bounds at the cost of an additional forward and backward pass. Since the bounds are loose at the beginning of training, a complex schedule is required to guarantee stability and convergence. Linear relaxation [43] delivers tighter bounds leveraging the correlation between different layers to linearly propagate signal and bounds, when possible, and eventually put in linear relation the input interval and output bounds. CROWN [43] bounds are often tighter than IBP, but at an impractical computational cost for large DNNs. Among the attempts to speed up CROWN [33, 37], CROWN-IBP [42] strikes the right balance of computational complexity and tightness. It is implemented in the auto_LiRPA[1] library [36, 37] adopted here. One limitation of adversarial training, certified methods, and A^5 as well, is the trade-off between clean and certified (worst case under-attack) accuracy: increasing the latter leads to a decrease in the former [4, 12, 20, 42]. Among the architecture solutions against adversarial attacks, we also mention the recent development of ℓ_∞-dist neurons that are inherently robust [41], and could be leveraged by A^5 as well in future.

The second defense philosophy is based on active countermeasures taken *after* the attack is crafted. Image purification [18, 27, 40] projects an attacked image onto the manifold of natural images through a generative model (Fig. 1, c). It is less effective than adversarial training [7], but does not make any assumption on the classifier. Other forms of signal pre-processing have been proposed in the field of audio processing and automatic speech recognition, where MitM physical attacks can be easily implemented [22, 38] and partially defeated by randomized smoothing, DefenseGAN, variational autoencoder (VAE), and Parallel WaveGAN vocoder [13]. Both certified defenses and pre-processing methods do not assume any specific attack form, but the latter are less effective in white box scenarios [2, 31]. Randomization can also be used after the attack, *e.g*. by changing resolution or padding an image [35], or by picking random experts in a mixture, that is proved to be more robust than deterministic DNNs [21].

A^5 belongs to a third class of methods, that put in place countermeasures *before* the attack (Fig. 1, d). The roots of A^5 lie in two ideas that have been only partially investigated so far. The first is that, in adversarial scenarios, *the first actor to move has a significant advantage*. This rule is already leveraged for online privacy, where a preemptive adversarial attack prevents the identification of the framed subject [6, 19]. Since adversarial attacks often generalize to many classifiers, this ploy is particularly effective [26]. Preemptive protection is also a natural form of defense against MitM attacks that can be easily created in contexts like automotive [32], speech and audio processing [13, 22, 38], or anytime the capturing device communicates with a server [39]. Encryption [1, 28] and watermarking [16] coupled with deep learning work against MitM attacks, but do not achieve full protection, are not certified, require the key to be hidden (grey box scenario) and may significantly alter the appearance of the image. Blind-preprocessing [23] applies a tanh transform, normalization, quantization and thermometer encoding to achieve partial and uncertified defense. The closest algorithm to A^5 is preemptive robustification [17]: given an image $\boldsymbol{x}$ and a classifier C that can be fine-tuned, it computes $y_0 = C(\boldsymbol{x})$ and randomly samples attacks $\delta\boldsymbol{x}_A$ against $\boldsymbol{x}+\delta\boldsymbol{x}_D$ to find a defensive perturbation $\delta\boldsymbol{x}_D$, such that $(\boldsymbol{x}+\delta\boldsymbol{x}_D+\delta\boldsymbol{x}_A) \in y_0$ (correct classification under attack). Since A^5 uses certified bounds, it results in simpler training and higher protection, and does not require running C first, as our robustifier DNN generalizes to all the inputs. The second idea behind A^5 comes from physical adversarial examples, *i.e*., 3d physical objects [3] or 2d patches [5] that consistently fool a DNN under a wide range of viewing conditions. Reverting the attacker and defender roles, the complementary cost function leads to the creation of *unadversarial examples*: patches or textures that, once applied to a real 3d object, reinforce (instead of degrading) the DNN desired behaviour [24]. However, these are not designed to be inherently robust to adversarial attacks. We exploit the idea in A^5 in all those situations where system designers control, to some extent, the inputs fed to the DNN (*e.g*. in OCR, robotic systems or during the design of the road infrastructure). We show as an example the use of A^5 to design *certifiably protected* characters for OCR, something unexplored both in [17] and [24].

[1] Automatic Linear Relaxation based Perturbation Analysis

3. Method

Threat models We summarize our notation in Table 1, whereas Fig. 2 illustrates the full A^5 framework including trainable DNNs in blue, training algorithms in green, physical objects $\boldsymbol{w}$, and acquired data $\boldsymbol{x}$. MitM attacks are crafted while transmitting data from the acquisition device to the classifier C, while physical attacks (not tested here) are crafted onto physical objects $\boldsymbol{w}$, before data acquisition. A^5 can be used in a scenario where the attacker has full access to the C: since it uses certified bounds, it is agnostic to the specific nature of the attack. In other words, we certify protection for any $\delta\boldsymbol{x}_A$, $||\delta\boldsymbol{x}_A|| < \epsilon_A$ and the attacker can use any white-, grey- or black-box attacking algorithm. We assume that A^5 can run in a protected environment, not accessible to the attacker, to perform robustification (light blue rectangles in Fig. 2). More details and limitations of our threat model are discussed in the Supplementary, whereas here we first introduce useful definitions and then provide several recipes for the use of A^5 in different scenarios.

Definitions For a classifier C and input $\boldsymbol{x}$, $\boldsymbol{y} = C(\boldsymbol{x})$ is the logit vector output by C. Under attack, we have:

$$\boldsymbol{y} + \delta\boldsymbol{y} = C(\boldsymbol{x} + \delta\boldsymbol{x}_A),\ ||\delta\boldsymbol{x}_A||_p < \epsilon_A, \quad (1)$$

where ϵ_A is the magnitude of the attack $\delta\boldsymbol{x}_A$ in p-norm and $\delta\boldsymbol{y}$ is the logit perturbation. Linear relaxation [43] estimates the lower ($\boldsymbol{y}^l$) and upper ($\boldsymbol{y}^u$) bounds of $\boldsymbol{y} + \delta\boldsymbol{y}$, for instance using IBP [12], CROWN [43], CROWN-IBP [42], or a combination of them. The margin for the class j is the difference between the lower bound of the ground truth class j^* and the upper bound of j, while it is zero by definition for j^*. More formally, for M classes the elements of the margin vector $\boldsymbol{m} = [m_0, m_1, ..., m_{M-1}]$ are: $m_{j^*} = 0$

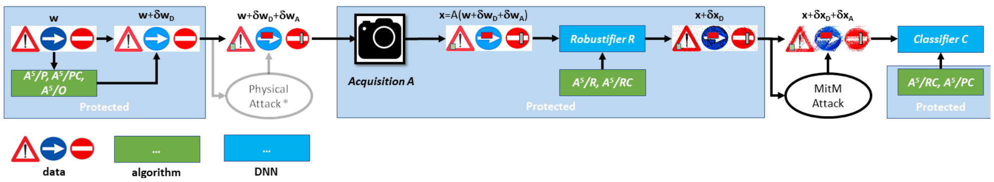

Figure 2. Schematic representation of the full A^5 framework. It allows the robust augmentation of both physical objects $\boldsymbol{w}$ and acquired data $\boldsymbol{x}$ to make them certifiably robust against MitM and physical (not tested here) adversarial attacks.

and $m_j = \boldsymbol{y}^l_{j^*} - \boldsymbol{y}^u_j$, $\forall j \neq j^*$. The confidence is obtained by applying softmax to $\boldsymbol{m}$:

$$p_j(\boldsymbol{x}) = e^{-m_j} / \sum_{k=0}^{M-1} e^{-m_k}, \tag{2}$$

and thus the worst case cross entropy $E(\boldsymbol{x})$ is:

$$E(\boldsymbol{x}) = -\sum_{j=0}^{M-1} \hat{y}_j log(p_j(\boldsymbol{x})), \tag{3}$$

where $\hat{y}_j$ is the one-hot-encoding of j^*. To formally introduce A^5, we first give the following definitions:

Definition 1 (ϵ_A-robustness)**.** Given an input $\boldsymbol{x}$ and a classifier C, we say that $\boldsymbol{x}$ is *ϵ_A-robust with respect to C in norm p* if $\forall \delta\boldsymbol{x}_A$, $||\delta\boldsymbol{x}_A||_p < \epsilon_A$, $C(\boldsymbol{x} + \delta\boldsymbol{x}_A)$ generates the correct classification output.

Definition 2 (robustification)**.** For a given input $\boldsymbol{x}$ and classifier C, *robustification* is the process of finding a defensive augmentation $\delta\boldsymbol{x}_D$, $||\delta\boldsymbol{x}_D||_p < \epsilon_D$, such that $\boldsymbol{x} + \delta\boldsymbol{x}_D$ is ϵ_A-robust with respect to C in norm p.

We compute certified CROWN-IBP bounds through auto_LiRPA [37] and optimize $E(\boldsymbol{x} + \delta\boldsymbol{x}_D)$ to find preemptive, defensive perturbations $\delta\boldsymbol{x}_D$ that make $\boldsymbol{x} + \delta\boldsymbol{x}_D$ ϵ_A-robust, following the training recipes provided here. We also show how to find defensive perturbations $\delta\boldsymbol{w}_D$ for physical objects $\boldsymbol{w}$ using the same approach.

Offline robustification with ground truth (A^5/O): Consider a legacy classifier C that has been certifiably trained (*e.g.* through CROWN-IBP [42]) against attacks of magnitude ϵ^C_A. Given an input $\boldsymbol{x}$ and its class j^*, we want to find a defensive perturbation $\delta\boldsymbol{x}_D$, $||\delta\boldsymbol{x}_D||_p < \epsilon_D$, such that $\boldsymbol{x} + \delta\boldsymbol{x}_D$ is ϵ^R_A-robust with respect to C in p-norm. Our solution to this problem is A^5/Offline (or A^5/O).

To force $||\delta\boldsymbol{x}_D||_p < \epsilon_D$, we parameterize $\delta\boldsymbol{x}_D$ via a vector $\boldsymbol{z}$ to which we apply an element-wise sigmoid:

$$\delta\boldsymbol{x}_D = \delta\boldsymbol{x}_D(\boldsymbol{z}) = [\delta x_D(\boldsymbol{z})_0, \ldots, \delta x_D(\boldsymbol{z})_{K-1}] \tag{4}$$

$$\delta x_D(\boldsymbol{z})_j = 2\epsilon_D[1/(1 + e^{-z_j}) - 0.5], \tag{5}$$

where K is the number of elements in $\boldsymbol{x}$. This is a robust classification problem where $\boldsymbol{z}$ is the only unknown. For any $\boldsymbol{x}$, we minimize the worst case cross entropy:

$$\boldsymbol{z} = \text{argmin}_{\tilde{\boldsymbol{z}}}\, E[\boldsymbol{x} + \delta\boldsymbol{x}_D(\tilde{\boldsymbol{z}})], \tag{6}$$

using RMSProp. Notice that the magnitude of the attack used during robustification (ϵ^R_A) and that adopted to train C (ϵ^C_A) can be different. Furthermore, similarly to adversarial training and certified methods, these can also be different from the target attack magnitude, ϵ_A.

A^5/O is similar in spirit to preemptive image robustification [17], but does not need C to run on the clean input before robustification. Although A^5/O could be used in a similar way to robustify $\boldsymbol{x}$ accordingly to the class $C(\boldsymbol{x})$, running C first and then iteratively solving the optimization problem in Eq. 6 makes it hardly usable in practice. Therefore, it does not directly serve any practical purpose, but it helps answering important theoretical questions (*e.g.*, under which conditions can we robustify $\boldsymbol{x}$? What does a robustified image looks like?) and establishing a baseline with known ground truth and no computational constraints.

Online robustification with robustifier R, legacy C (A^5/R): We introduce a second recipe, A^5/R, where we train a robustifier DNN R that takes in input $\boldsymbol{x}$, no ground truth label, and outputs $\boldsymbol{z}$, from which a defensive perturbation $\delta\boldsymbol{x}_D$, $||\delta\boldsymbol{x}_D||_p < \epsilon_D$, is computed as in Eq. 5. We assume again that a robust legacy C is given and we train R to make each $\boldsymbol{x} + \delta\boldsymbol{x}_D$ in the training dataset ϵ^R_A-robust with respect to C in p-norm. More formally, the weights $\boldsymbol{\theta}_R$ of R are found by minimization of the average worst case entropy on the training dataset $E(\boldsymbol{\theta}_R)$, defined as:

$$\boldsymbol{z} = R(\boldsymbol{x}|\boldsymbol{\theta}_R),\; E(\boldsymbol{\theta}_R) = (1/N)\sum_{\boldsymbol{x}} E[\boldsymbol{x} + \delta\boldsymbol{x}_D(\boldsymbol{z})], \tag{7}$$

where N is the dataset size. Unlike preemptive robustification [17], R does not iteratively solve a complex optimization problem, neither we need running C before R. Therefore, we can use A^5/R to protect $\boldsymbol{x}$ against MitM adversarial attacks on-the-fly, ignoring its class, soon after its acquisition, and before its transmission to C (see A^5/R in Supplementary). Furthermore, since R is trained for a specific

C, it can be deployed on the field (*e.g.*, into the firmware of the acquisition devices) without changing (and thus maintaining the legacy of) the classifier infrastructure.

Online robustification with robustifier R, retrained C (A^5/RC): The A^5/RC recipe is equivalent to A^5/R, but it leverages the co-adaptation of C and R by joint training (see A^5/RC in Supplementary). The cost function is again the average worst case entropy $E(\boldsymbol{\theta}_C, \boldsymbol{\theta}_R)$, defined now as:

$$\boldsymbol{z} = R(\boldsymbol{x}|\boldsymbol{\theta}_R), \quad \boldsymbol{y} = C(\boldsymbol{x} + \delta\boldsymbol{x}_D|\boldsymbol{\theta}_C), \tag{8}$$

$$E(\boldsymbol{\theta}_C, \boldsymbol{\theta}_R) = (1/N)\sum_{\boldsymbol{x}} E[\boldsymbol{x} + \delta\boldsymbol{x}_D(\boldsymbol{z})], \tag{9}$$

where $\boldsymbol{\theta}_C$ are the parameters of C. Training R and C from scratch is often unstable and requires careful tuning to achieve convergence (likewise robust classifiers [12, 42]). Therefore, we first train R with A^5/C and then fine tune R and C together. During the tuning phase, we can use random input augmentation to prevent overfitting (see Section 4 and Supplementary for details). Differently from A^5/R, A^5/RC does not preserve the legacy classifier C, but it achieves the same scope (with more effectiveness) in terms of protection against MitM attacks.

Offline physical robustification with (A^5/PC) and without (A^5/P) retraining C, with ground truth: This recipe aims at producing physical objects that are certifiably robust against MitM attacks (see A^5/P, A^5/PC in Supplementary). Similarly to unadversarial objects [24], we assume that the system creator can design the entire infrastructure and the attacker cannot interfere in this phase. The problem is formally defined for A^5/PC as:

$$\boldsymbol{x} + \delta\boldsymbol{x}_D = A(\boldsymbol{w} + \delta\boldsymbol{w}_D), \ ||\delta\boldsymbol{w}_D)|| < \epsilon_D, \tag{10}$$

$$E(\boldsymbol{\theta}_C, \delta\boldsymbol{w}_D) = (1/N)\sum_{\boldsymbol{x}} E(\boldsymbol{x} + \delta\boldsymbol{x}_D) \tag{11}$$

where $\boldsymbol{w}$ is the object appearance, whereas $\boldsymbol{x} = A(\boldsymbol{w})$ is the data acquisition model (*e.g.*, image formation). N is here the number of random samples generated as $\boldsymbol{x} + \delta\boldsymbol{x}_D = A(\boldsymbol{w} + \delta\boldsymbol{w}_D)$; it is important noticing in fact that the acquisition process is not deterministic: as an exemplary case we mention OCR, where $\boldsymbol{w}$ is the shape of a character, while A may include random perspective transformations, contrast and brightness changes, noise addition and blurring, that may occur on a real camera. The defensive perturbation $\delta\boldsymbol{w}_D$ represents a change in the physical character appearance $\boldsymbol{w}$ that decreases the success rate of an attack towards the acquired image $\boldsymbol{x}$ (therefore after framing the character with A) and the classifier C.

In A^5/P we change the character shapes and use the legacy classifier C; in A^5/PC we also train C while estimating the robust characters $\boldsymbol{w} + \delta\boldsymbol{w}$. In both cases C is previously trained to be robust, *e.g.*, through CROWN-IBP [42]. The A^5/P and A^5/PC recipes find application in all those situations where the developer can design the entire infrastructure; beyond OCR, other examples are road sign or robotic tool design and classification.

ϵ_A	$\epsilon_A^C = 0.0$	$\epsilon_A^C = 0.1$	$\epsilon_A^C = 0.2$	$\epsilon_A^C = 0.3$	$\epsilon_A^C = 0.4$
0.1	—	—	1.17 [—, 2.24] [42]	—	—
0.1	0.81 [89.02, 100.00]	0.92 [2.00, 5.41]	1.32 [2.14, 2.60]	1.44 [2.33, 2.63]	2.42 [3.36, 3.41]
0.3	—	—	—	1.82 [—, 7.02] [42]	—
0.3	0.69 [100.00, 100.00]	0.94 [98.41, 100.00]	1.30 [99.62, 100.00]	1.41 [7.06, 9.98]	2.58 [7.03, 7.68]

Table 2. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for MNIST classifiers trained by us with CROWN-IBP [42], under attack $\epsilon_A = \{0.1, 0.3\}$; we also show the metrics reported in [42]. A 100% certified error rate identifies DNNs that are not protected against attacks.

$\epsilon_A^R = \epsilon_A$	ϵ_D	$\epsilon_A^C = 0.0$	$\epsilon_A^C = 0.1$	$\epsilon_A^C = 0.2$	$\epsilon_A^C = 0.3$	$\epsilon_A^C = 0.4$
0.1	0.05	0.79 [87.09, 100.00]	0.82 [1.51, 2.89]	1.22 [1.64, 1.86]	1.34 [1.76, 1.87]	2.38 [2.91, 2.92]
0.1	0.10	0.79 [86.51, 100.00]	0.72 [1.21, 1.92]	1.15 [1.43, 1.49]	1.34 [1.50, 1.58]	2.41 [2.71, 2.74]
0.1	0.20	0.89 [86.40, 100.00]	0.56 [0.78, 1.04]	1.13 [1.18, 1.21]	1.35 [1.37, 1.38]	2.40 [2.55, 2.55]
0.1	0.30	0.93 [86.44, 100.00]	0.51 [0.57, 0.70]	1.14 [1.16, 1.17]	1.34 [1.37, 1.37]	2.38 [2.52, 2.52]
0.1	0.40	0.99 [85.93, 100.00]	0.52 [0.53, 0.60]	1.14 [1.16, 1.16]	1.35 [1.38, 1.38]	2.44 [2.51, 2.51]
0.3	0.05	0.71 [100.00, 100.00]	0.93 [97.18, 100.00]	1.34 [99.17, 100.00]	1.35 [5.16, 6.97]	2.55 [6.00, 6.42]
0.3	0.10	0.72 [100.00, 100.00]	0.88 [95.32, 100.00]	1.36 [98.58, 100.00]	1.33 [4.12, 5.19]	2.49 [5.54, 5.85]
0.3	0.20	0.72 [100.00, 100.00]	0.88 [90.62, 100.00]	1.41 [96.21, 100.00]	1.32 [3.21, 3.77]	2.35 [5.02, 5.22]
0.3	0.30	0.73 [100.00, 100.00]	0.89 [86.28, 100.00]	1.41 [94.18, 100.00]	1.31 [2.91, 3.40]	2.32 [4.66, 4.81]
0.3	0.40	0.81 [100.00, 100.00]	0.85 [83.49, 100.00]	1.42 [92.63, 100.00]	1.32 [2.74, 3.20]	2.28 [4.16, 4.29]

Table 3. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for MNIST, A^5/O, under attack $\epsilon_A = \{0.1, 0.3\}$. During training we use $\epsilon_A^R = \epsilon_A$.

$\epsilon_A^R = \epsilon_A$	ϵ_D	$\epsilon_A^C = 0.0$	$\epsilon_A^C = 0.1$	$\epsilon_A^C = 0.2$	$\epsilon_A^C = 0.3$	$\epsilon_A^C = 0.4$
0.1	0.05	0.82 [85.50, 100.00]	0.95 [1.71, 3.58]	1.30 [1.82, 2.02]	1.40 [1.86, 2.03]	2.43 [2.96, 2.97]
0.1	0.10	0.81 [82.87, 100.00]	1.00 [1.48, 2.54]	1.19 [1.62, 1.73]	1.40 [1.68, 1.74]	2.38 [2.77, 2.79]
0.1	0.20	0.88 [78.48, 100.00]	0.93 [1.33, 1.77]	1.20 [1.42, 1.52]	1.31 [1.53, 1.59]	2.26 [2.55, 2.57]
0.1	0.30	0.99 [75.89, 100.00]	0.91 [1.16, 1.54]	1.14 [1.31, 1.36]	1.26 [1.43, 1.47]	2.20 [2.42, 2.43]
0.1	0.40	1.04 [75.69, 100.00]	0.87 [1.05, 1.35]	1.14 [1.27, 1.33]	1.24 [1.43, 1.46]	2.22 [2.41, 2.43]
0.3	0.05	0.69 [100.00, 100.00]	0.94 [96.93, 100.00]	1.33 [99.11, 100.00]	1.41 [5.13, 6.91]	2.58 [5.77, 6.09]
0.3	0.10	0.71 [100.00, 100.00]	0.91 [95.09, 100.00]	1.33 [98.47, 100.00]	1.38 [3.86, 4.81]	2.55 [4.71, 4.83]
0.3	0.20	0.72 [100.00, 100.00]	0.92 [90.13, 100.00]	1.37 [96.08, 100.00]	1.41 [2.26, 2.63]	2.55 [3.33, 3.40]
0.3	0.30	0.75 [100.00, 100.00]	0.93 [85.54, 100.00]	1.35 [93.36, 100.00]	1.32 [1.86, 1.99]	2.37 [2.86, 2.88]
0.3	0.40	0.83 [100.00, 100.00]	0.96 [82.48, 100.00]	1.36 [92.15, 100.00]	1.29 [1.69, 1.83]	2.35 [2.67, 2.69]

Table 4. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for MNIST, A^5/R, under attack $\epsilon_A = \{0.1, 0.3\}$. During training we use $\epsilon_A^R = \epsilon_A$.

4. Results and discussion

We test A^5/O, A^5/R, and A^5/RC on MNIST [10], CIFAR10 [14], FashionMNIST [34], and Tinyimageenet [15], that have been widely used to establish significant milestones in the adversarial defense field. Training schedules and DNN architectures are given in the Supplementary. In all cases we use a $p = \infty$ norm for the attack and defense. Our intention is to show the improvement achieved by A^5 over traditional, certified methods like CROWN-IBP. The reader should anyway keep in mind that the direct comparison of A^5 and CROWN-IBP is partially unfair, as CROWN-IBP does not apply any preemptive defense.

We first use CROWN-IBP to train five certifiably robust MNIST classifiers with varying levels of robustness, for training attacks $\epsilon_A^C = 0.0$ (unprotected C) and $\epsilon_A^C = \{0.1, 0.2, 0.3, 0.4\}$. Their clean and certified errors are reported in Table 2, for attacks $\epsilon_A = \{0.1, 0.3\}$. We also

$\epsilon_A^R = \epsilon_A$	ϵ_D	$\epsilon_A^C = 0.0$	$\epsilon_A^C = 0.1$	$\epsilon_A^C = 0.2$	$\epsilon_A^C = 0.3$	$\epsilon_A^C = 0.4$
0.1	0.05	88.66 [88.66, 88.66]	1.00 [1.38, 2.35]	1.16 [1.61, 2.46]	1.14 [1.54, 2.29]	1.37 [1.72, 2.33]
0.1	0.10	88.66 [88.66, 88.66]	0.77 [0.95, 1.40]	0.99 [1.16, 1.56]	0.86 [1.03, 1.42]	1.13 [1.31, 1.72]
0.1	0.20	88.66 [88.66, 88.66]	0.69 [0.80, 1.05]	0.84 [0.94, 1.25]	0.96 [1.07, 1.33]	0.88 [1.10, 1.54]
0.1	0.30	88.66 [88.66, 88.66]	0.64 [0.70, 1.02]	0.91 [0.99, 1.16]	0.79 [0.88, 1.21]	1.00 [1.12, 1.43]
0.1	0.40	88.66 [88.66, 88.66]	0.61 [0.68, 0.88]	0.93 [0.99, 1.18]	0.69 [0.76, 1.10]	1.00 [1.03, 1.30]
0.3	0.05	88.66 [88.66, 88.66]	88.66 [88.66, 88.66]	2.57 [8.35, 10.86]	3.42 [9.51, 11.72]	8.00 [18.05, 19.77]
0.3	0.10	88.66 [88.66, 88.66]	88.66 [88.66, 88.66]	2.07 [5.66, 7.56]	2.35 [6.18, 8.07]	2.35 [6.31, 7.93]
0.3	0.20	88.66 [88.66, 88.66]	1.68 [3.11, 4.79]	1.40 [2.42, 3.62]	1.28 [2.82, 4.01]	1.88 [3.53, 4.90]
0.3	0.30	88.66 [88.66, 88.66]	0.84 [0.96, 1.38]	1.00 [1.14, 1.36]	0.94 [1.15, 1.59]	1.07 [1.39, 1.90]
0.3	0.40	88.66 [88.66, 88.66]	0.72 [0.80, 1.03]	0.75 [0.82, 1.06]	0.91 [1.14, 1.68]	0.87 [1.14, 1.50]

Table 5. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for MNIST, A^5/RC, under attack $\epsilon_A = \{0.1, 0.3\}$. During training we use $\epsilon_A^R = \epsilon_A$, and data augmentation.

report the error rates estimated with autoattack [8, 9], an automatic toll based on an ensemble of four attacks to evaluate the practical DNN robustness. These error rates are in the same ballpark of those reported in the CROWN-IBP paper [42]: they represent the baseline for the analysis of the different A^5 recipes considered here.

For each C we run A^5/O on the MNIST test set, for defense magnitudes $\epsilon_D = \{0.05, 0.1, 0.2, 0.3, 0.4\}$. During the optimization of the entropy in Eq. 6, we use attacks $\epsilon_A^R = \{0.1, 0.3\}$, not necessarily equal to ϵ_A^C. Table 3 reports the clean, autoattack, and certified errors for a testing attack $\epsilon_A = \epsilon_A^R$. When C is not robust ($\epsilon_A^C = 0.0$), A^5/O does not reduce the certified error[2], otherwise it generally overcomes the base CROWN-IBP classifier, as shown by comparing against the error rates in Table 2. As a rule of thumb, the best clean / certified error trade off is often achieved for $\epsilon_A^R = \epsilon_A^C = \epsilon_A$. Results also improve, as expected, for larger defenses ϵ_D, when A^5/O beat CROWN-IBP by a significant margin on clean and certified errors. However, A^5/O does not work well when the legacy C is trained with a large ϵ_A^C, possibly because of its initial low clean accuracy. The results are consistent with our interpretation of the effects of adversarial training and robustification on the classification landscape (Fig. 1): training C with a large ϵ_A^C creates classification valleys wide enough to accommodate $\boldsymbol{x} + \delta\boldsymbol{x}_D + \delta\boldsymbol{x}_A$, rendering the adversarial attacks ineffective. A^5 moves any $\boldsymbol{x}$ towards the centers of the valleys; a large ϵ_D allows moving them more easily. Too large ϵ_A^C remain however detrimental for the clean accuracy.

Table 4 reports results for A^5/R, demonstrating that training a DNN robustifier R for on-the-fly, certified robustification, ignoring the ground truth, is feasible. Likewise A^5/O, A^5/R beats the base CROWN-IBP classifier by a significant margin. Sometimes A^5/O does better by leveraging the knowledge of the ground truth class, whereas the regular robustification signal generated by A^5/R is more effective in other cases.

Our last MNIST experiment investigates A^5/RC that does not preserve the legacy C like A^5/R, but achieves better clean accuracy and protection (Table 5). In the Supplementary we discuss the role of data augmentation to achieve this result. Interestingly, the best results are obtained when C is initially trained for $\epsilon_A^C \leq \epsilon_A$, probably because C has an initial high clean accuracy and subsequently co-adapts with R to also guarantee a small certified error.

Fig. 3 shows examples of robustification on MNIST. At visual inspection, A^5/O, A^5/R, and A^5/RC act similarly on these images: they add contrast and enhance the high frequency features (*i.e.*, edges) of the digits. The few failure cases are often pictures that look ambiguous even to a human observer (right panels in Fig. 3).

We perform tests on CIFAR10 and compare A^5 against a state of the art CROWN-IBP classifiers C, trained with $\epsilon_A^C = 8.8/255$, that we also adopt as legacy C for A^5. To test the effect of the magnitude of the attack used for robustification (with A^5/O) or for training R (for A^5/R, A^5/RC), we use $\epsilon_A^R = \{4/255, 8/255, 16/255\}$ that are

[2]Coherently with [24], the worst case cross entropy improves, but not enough to affect the ranking of the classified classes under attack.

ϵ_A^C	ϵ_D	$\epsilon_A^R = 4/255$	$\epsilon_A^R = 8/255$	$\epsilon_A^R = 16/255$
8/255	4/255	52.75 [63.47, 64.09]	53.34 [62.42, 62.79]	53.75 [64.92, 65.49]
8/255	8/255	51.06 [60.04, 60.73]	52.36 [59.24, 59.58]	53.16 [63.50, 64.03]
8/255	16/255	48.72 [54.89, 55.25]	50.60 [55.22, 55.38]	52.23 [60.79, 61.14]
8/255	32/255	45.68 [49.51, 49.74]	48.06 [50.86, 50.94]	50.99 [57.37, 57.61]

Table 6. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for CIFAR10, A^5/O, under attack $\epsilon_A = 8/255$. During training we use different ϵ_A^R and ϵ_D. The robust classifier trained with CROWN-IBP has clean [and certified] errors equal to 54.02 [66.94] or 45.47 [69.55], depending on the CROWN-IBP training configuration [42].

ϵ_A^C	ϵ_D	$\epsilon_A^R = 4/255$	$\epsilon_A^R = 8/255$	$\epsilon_A^R = 16/255$
8/255	0/255	54.60 [66.34, 67.11]	54.60 [66.34, 67.11]	54.60 [66.31, 67.11]
8/255	4/255	53.73 [64.07, 64.57]	54.12 [63.32, 63.74]	54.37 [63.58, 64.03]
8/255	8/255	52.91 [62.59, 63.12]	53.24 [61.38, 61.93]	54.54 [60.85, 61.16]
8/255	16/255	51.58 [60.40, 61.02]	51.95 [59.21, 59.67]	54.44 [59.87, 60.11]
8/255	32/255	50.91 [57.26, 57.75]	51.70 [56.91, 57.43]	54.05 [58.39, 58.73]

Table 7. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for CIFAR10, A^5/R, under attack $\epsilon_A = 8/255$. During training we use different ϵ_A^R and ϵ_D. The robustifier R in the first row uses $\epsilon_D = 0$ and thus it does not provide any form of defense — the error reported here is equivalent to that of a robust classifier trained with CROWN-IBP [42].

ϵ_A^C	ϵ_D	$\epsilon_A^R = 4/255$	$\epsilon_A^R = 8/255$	$\epsilon_A^R = 16/255$
8/255	0/255	50.24 [69.17, 73.65]	58.63 [69.25, 70.63]	74.28 [77.82, 77.91]
8/255	4/255	46.76 [63.03, 70.99]	54.05 [62.80, 64.44]	67.59 [71.20, 71.32]
8/255	8/255	41.24 [54.01, 64.31]	50.74 [56.63, 59.49]	65.96 [67.63, 67.78]
8/255	16/255	36.96 [42.45, 52.90]	39.15 [41.33, 44.22]	58.17 [58.67, 58.85]
8/255	32/255	35.26 [37.13, 42.76]	41.59 [43.01, 45.55]	42.25 [43.02, 44.08]

Table 8. Error on clean data and (within brackets) under autoattack [8, 9] and certified, for CIFAR10, A^5/RC, under attack $\epsilon_A = 8/255$. During training we use different ϵ_A^R and ϵ_D. The robustifier R in the first row uses $\epsilon_D = 0$ and thus it does not provide any form of defense, whereas the corresponding robust CROWN-IBP classifier is fine tuned using $\epsilon_A^C = \epsilon_R$ during training.

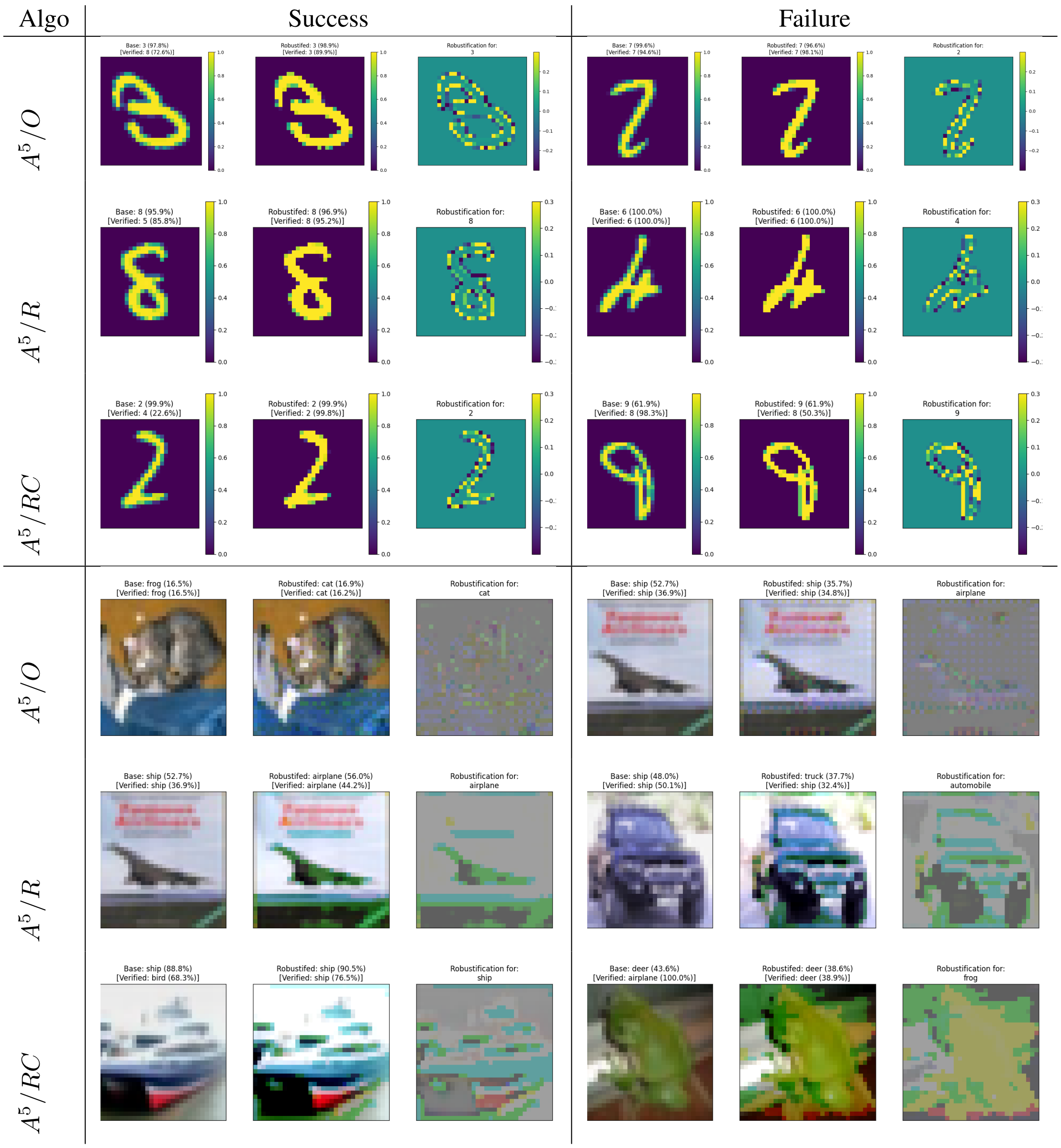

Figure 3. Successful and failed robustifications for A^5/O, A^5/R, and A^5/RC, for MNIST ($\epsilon_A = \epsilon_A^C = \epsilon_A^R = \epsilon_D = 0.3$) and CIFAR10 ($\epsilon_A = 8.8/255$, $\epsilon_A^C = 8/255$, $\epsilon_A^R = 4/255$, and $\epsilon_D = 32/255$). Each triplet shows the original (left) and robustified (center) images; the rightmost panel is the defensive augmentation. All the A^5 recipes consistently increase the contrast and the high frequency content on MNIST, where failure cases are ambiguous even for a human observer. For CIFAR10, A^5/O leverage isolated pixel changes, whereas A^5/R and A^5/RC consistently increase the color contrast and saturation, similarly to results in literature [17].

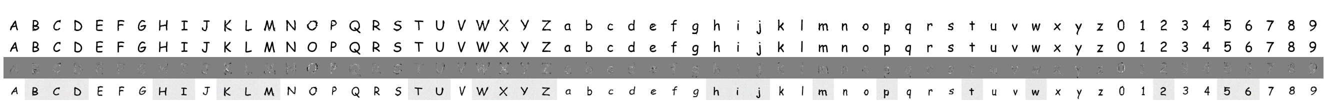

Figure 4. The first, second and third row show respectively the robustified ($\boldsymbol{w} + \delta\boldsymbol{w}_D$), nominal ($\boldsymbol{w}$) and robustification ($\delta\boldsymbol{w}_D$) sets of 62 letters considered for testing A^5/PC. The last row show examples of the robustified letters after framing them, $\boldsymbol{x}+\delta\boldsymbol{x}_D = A(\boldsymbol{w}+\delta\boldsymbol{w}_D)$.

smaller, equal or greater than ϵ_A^C. Results in Table 6 are consistent with those measured for MNIST: A^5/O significantly improves the clean and certified error rates over the base CROWN-IBP classifier. Training R through A^5/R

Algo	Error [certified error]
Vanilla	0.89% [100.00%]
CROWN-IBP	3.85% [13.85%]
A^5/P	3.08% [11.84%]
A^5/PC	0.73% [4.20%]

Table 9. Error and certified error for the classification of the 62 characters in Fig. 4, for $\epsilon_A = 0.1$ and different algorithms. A^5/P changes the physical shape of the letters to achieve robustness after framing them; A^5/PC also trains the associated classifier.

produces a smaller improvement, still sufficient to beat the CROWN-IBP classifier by a significant margin (Table 7). The value of ϵ_A^R has a significant impact on the final result: if too large ($\epsilon_A^R > \epsilon_A$), the clean accuracy is penalized, because A^5 looks for very large plateaus in the classification landscape, that are hard or impossible to find. In other words, we observe for A^5 a trade-off between the clean and certified accuracy, likewise adversarial training and certified methods. Best results on CIFAR10 are achieved again by A^5/RC, when ϵ_A^R is equal or slightly smaller than $\epsilon_A = \epsilon_A^C$. The improvements in clean and certified error are in this case in the impressive range of 19% and 24% respectively (Table 8). We also notice here that the theoretical certified error rates measured with auto_LiRPA [37] in A^5 and those measured experimentally by autoattack [8, 9] are close for A^5, whose working point at convergence may favor the consistency between these two estimates.

Also for CIFAR10 we perform a visual inspection of defensive augmentation (Fig. 3). A^5/O exploits a slight contrast increase and modifies isolated pixels: likewise adversarial attacks with small p norm (up to single pixel attacks [29]), the same seems possible when crafting defensive perturbations. Preemptive robustification [17] shows a similar pattern (see their Fig. 6): non natural textures and enhanced color saturation emerge in their images for large defensive perturbations. The robust images crafted by R in A^5/R or A^5/RC, however, are different: they generally show increased contrast and color saturation, without the textures generated by A^5/O and [17]. We speculate that these textures may be associated with the iterative optimization in A^5/O and [17], that may both suffer some form of obfuscated gradient [2]; a more regular defense is instead generated by R without resorting to any iterative process.

Additional results on FashionMNIST and Tinyimagenet, overall consistent with the ones presented here and showing the quantitative advantages provided by A^5 on these datasets, are reported in the Supplementary. These experiments highlight two important facts. The first one is that the integration of the design philosophy of A^5 with that of other recent ideas in the field of adversarial defense (like the ℓ_∞-dist neurons that implement 1-Lipschitz functions and are inherently robust to adversarial attacks) may lead to even bigger improvements in terms of certified robustness. The second is that the integration of different defense strategies may indeed be strictly needed to guarantee further progresses, as scaling to very large dataset remains problematic even for a preemptive robustification algorithms like A^5.

Finally, for A^5/P and A^5/PC, we test the classification of an alphabet of 62 characters (Fig. 4), after random rotation, shift, perspective distortion, noise addition, blurring and color jittering (details in Supplementary), that simulate the scanning of a document for OCR. Table 9 shows the clean and certified errors for: a vanilla, non robust classifier; a CROWN-IBP classifier; A^5/P; and A^5/PC. A^5 achieves again a significant improvement both on clean and certified errors: A^5/PC has better (0.73%) accuracy of the vanilla C (0.89%) and can be attacked in only 4.20% of the cases. This results is obtained with a slight modification of the shape of the prototype characters shown in Fig. 4.

5. Discussion and conclusion

We introduce A^5, the first framework that leverages adversarial augmentation to preemptively modify the input of a DNN or a physical object, to make it certifiable robust against adversarial attacks. A^5 is not simply complementary to other defense methods: its tights with them are strict and they can be used together to achieve higher robustness. For instance, one of the limitations we found is that A^5/R requires a somehow robust initial classifier C. Co-training R and a poorly robust C easily compensates for the initially low robustness of C and boosts its high clean accuracy while also achieving state of the art results in terms of certified accuracy; finding the optimal initial robustness of C remains an open question though.

Here we do not investigate all the possible A^5 recipes: in its most general implementation (A^5/PRC) the physical objects, R and C could all be optimized at the same time. This opportunity can be explored through the released code at https://github.com/NVlabs/A5. This will also serve answering other questions that do not find space here: for instance, we do not know if robustification generalizes to other classifiers as adversarial attacks do, how A^5 works in norms other than $p = \infty$, and how much it can further improve if coupled with state-of-the-art solutions like the recently proposed ℓ_∞-dist neurons [41].

Different A^5 recipes find applications in scenarios where the user can control the acquisition device, equip it with R and guarantee its protection (*e.g.*, image acquisition on a phone before server communication) or while designing the infrastructure (*e.g.*, robust road signs creation); it may be not suitable for more general adversarial scenarios. Overall, we believe that the practical deployment of robust systems may benefit from (or even require) methods for both robust classifiers and preemptive robustification.

References

[1] Maungmaung Aprilpyone and Hitoshi Kiya. Encryption inspired adversarial defense for visual classification. *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1681–1685, 2020. 2, 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 10–15 Jul 2018. 3, 8

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples, 2018. 3

[4] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4312–4321. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track. 1, 2, 3

[5] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial patch. 2017. 3

[6] Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John P Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *International Conference on Learning Representations*, 2021. 1, 3

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 13–18 Jul 2020. 3

[8] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 5, 6, 8

[9] Francesco Croce and Matthias Hein. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In *ICML*, 2021. 5, 6, 8

[10] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015. 1, 2

[12] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models, 2018. 1, 2, 3, 5

[13] Sonal Joshi, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velázquez, and Najim Dehak. Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems. *IEEE Transactions on Information Forensics and Security*, 16:4811–4826, 2021. 1, 2, 3

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[15] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 5

[16] Xiaoting Li, Lingwei Chen, Jinquan Zhang, James Larus, and Dinghao Wu. Watermarking-based defense against adversarial attacks on deep neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. 2, 3

[17] Seungyong Moon, Gaon An, and Hyun Oh Song. Preemptive image robustification for protecting users against man-in-the-middle adversarial attacks. In *AAAI Conference on Artificial Intelligence*, 2022. 2, 3, 4, 7, 8

[18] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022. 1, 3

[19] Seong Joon Oh, Mario Fritz, and Bernt Schiele. Adversarial image perturbation for privacy protection a game theory perspective. pages 1491–1500, 10 2017. 1, 3

[20] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2021. 1, 2, 3

[21] Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7717–7727. PMLR, 13–18 Jul 2020. 3

[22] Yao Qin, Nicholas Carlini, Ian J. Goodfellow, G. Cottrell, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. *ArXiv*, abs/1903.10346, 2019. 2, 3

[23] Adnan Siraj Rakin, Zhezhi He, Boqing Gong, and Deliang Fan. Blind pre-processing: A robust defense method against adversarial examples. *ArXiv*, abs/1802.01549, 2018. 2, 3

[24] Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2, 3, 5, 6

[25] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[26] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. *Fawkes: Protecting Privacy against Unauthorized Deep Learning Models*. USENIX Association, USA, 2020. 1, 3

[27] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*, 2021. 1, 3

[28] Jinmyeong Shin, Seok-Hwan Choi, and Yoon-Ho Choi. Is homomorphic encryption-based deep learning secure enough? *Sensors*, 21(23), 2021. 2, 3

[29] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 8

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. Jan. 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014. 1

[31] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1633–1645. Curran Associates, Inc., 2020. 3

[32] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. Man-in-the-middle attacks against machine learning classifiers via malicious generative models. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2074–2087, sep 2021. 2, 3

[33] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[34] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 5

[35] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 1, 3

[36] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *arXiv: Learning*, 2020. 3

[37] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *International Conference on Learning Representations*, 2021. 2, 3, 4, 8

[38] Hiromu Yakura and Jun Sakuma. Robust audio adversarial example for a physical attack. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5334–5341. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 2, 3

[39] Jianfei Yang, Han Zou, and Lihua Xie. Robustsense: Defending adversarial attack for secure device-free human activity recognition. *ArXiv*, abs/2204.01560, 2022. 2, 3

[40] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12062–12072. PMLR, 18–24 Jul 2021. 1, 3

[41] Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12368–12379. PMLR, 18–24 Jul 2021. 3, 8

[42] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 4, 5, 6

[43] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems (NIPS), arXiv preprint arXiv:1811.00866*, dec 2018. 1, 2, 3