Catastrophic overfitting is a bug but also a feature

Guillermo Ortiz-Jimenez [*1] **Pau de Jorge** [*2 3] **Amartya Sanyal** [4 5] **Adel Bibi** [2]
Puneet K. Dokania [2 6] **Pascal Frossard** [1] **Grégory Rogez** [3] **Philip H.S. Torr** [2]

Abstract

Despite clear computational advantages in building robust neural networks, adversarial training (AT) using single-step methods is unstable as it suffers from catastrophic overfitting (CO): Networks gain non-trivial robustness during the first stages of adversarial training, but suddenly reach a breaking point where they quickly lose all robustness in just a few iterations. Although some works have succeeded at preventing CO, the different mechanisms that lead to this remarkable failure mode are still poorly understood. In this work, however, we find that the interplay between the structure of the data and the dynamics of AT plays a fundamental role in CO. Specifically, through active interventions on typical datasets of natural images, we establish a causal link between the structure of the data and the onset of CO in single-step AT methods. This new perspective provides important insights into the mechanisms that lead to CO and paves the way towards a better understanding of the general dynamics of robust model construction.

1. Introduction

Let $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathcal{Y}$ denote a neural network architecture parameterized by a set of weights $\boldsymbol{\theta} \in \mathbb{R}^n$ which maps input samples $\boldsymbol{x} \in \mathbb{R}^d$ to $y \in \mathcal{Y} = \{1, \dots, c\}$. The objective of adversarial training (AT) is to find the network parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ that optimize the following min-max problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\boldsymbol{\delta}\|_p \leq \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y) \right], \tag{1}$$

where $\mathcal{D}$ is some underlying data distribution and $\boldsymbol{\delta} \in \mathbb{R}^d$ represents an adversarial perturbation (Szegedy et al., 2014; Boloor et al., 2019). This is typically solved by alternately minimizing the outer objective and maximizing the inner one via first-order optimization procedures. The outer minimization is tackled via some standard neural network optimizer, *e.g.,* SGD, while the inner maximization problem is approximated with adversarial attacks like Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) and Projected Gradient Descent (PGD) (Madry et al., 2018). Single-step AT methods are built on top of the FGSM attack. In particular, FGSM solves the linearized version of the inner maximization objective. In the ℓ_∞ case, this leads to the following attack:

$$\boldsymbol{\delta}_{\text{FGSM}} = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\right). \tag{2}$$

Note that FGSM is very efficient computationally as it only requires a single forward-backward step. However, the use of the computationally efficient single-step attacks within AT comes with concerns regarding its stability. While training, although there is an initial increase in robustness, the networks often reach a breaking point where they lose all gained robustness in just a few iterations (Wong et al., 2020). This phenomenon is known as *catastrophic overfitting* (CO) (Wong et al., 2020; Andriushchenko and Flammarion, 2020). Nevertheless, given the clear computational advantage of using single-step attacks during AT, a significant body of work has been dedicated to finding ways to circumvent CO via regularization and data augmentation (Andriushchenko and Flammarion, 2020; de Jorge et al., 2022; Kim et al., 2021; Park and Lee, 2021; Vivek and Babu, 2020; Golgooni et al., 2021).

Despite the recent methodological advances in this front, however, the *root cause of CO*, experienced by single step AT methods, remains poorly understood. We show, in this paper, that CO is connected to properties of the data, the effectiveness of the adversarial attacks, and the training dynamics. However, due to the inherent complexity of this problem, it is difficult to disentangle these factors of variation. Hence, we argue that identifying the causal mechanisms behind this failure mode cannot be done through observations alone and requires *active interventions* (Pearl and Mackenzie, 2018). That is, we

[*]Equal contribution. Guillermo Ortiz-Jimenez did this work while visiting the University of Oxford. [1]EPFL, Lausanne, Switzerland [2]University of Oxford, Oxford, UK [3]Naver Labs Europe, Grenoble, France [4]ETH, Zürich, Switzerland [5]ETH AI Center, Zürich, Switzerland [6]Five AI Ltd., Oxford, UK. Correspondence to: Guillermo Ortiz-Jimenez <guillermo.ortizjimenez@epfl.ch>, Pau de Jorge <pau@robots.ox.ac.uk>.

AdvML Frontiers workshop at 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

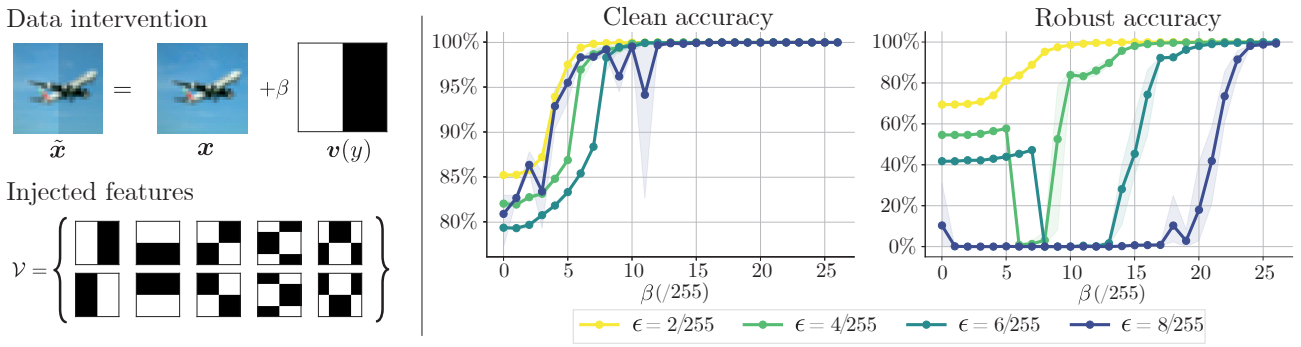


Figure 1: **Left:** Depiction of our data intervention to introduce easy-to-learn, discriminative features. **Right:** Clean and robust performance after FGSM-AT on intervened datasets $\widetilde{\mathcal{D}}_\beta$. We vary the strength of the injected features β ($\beta = 0$ corresponds to the original CIFAR-10) and the robustness budget ϵ (train and test). We observe that for $\epsilon \in \{4/255, 6/255\}$ our intervention can induce CO when the injected features have strength β slightly larger than ϵ while training on the original data does not suffer CO. Results are averaged over 3 seeds and shaded areas report minimum and maximum values.

need to be able to synthetically induce CO in a training context where it would not naturally happen otherwise.

In this work, we identify one such type of intervention that allows to perform abundant *in-silico* experiments to explain multiple aspects of CO. Specifically the main contributions of our work are: (i) We show that CO can be induced by injecting easy-to-learn features that, despite being strongly discriminative, are not sufficient for robust classification by themselves (see Figure 1). (ii) Through extensive empirical analysis, we discover that CO is connected to the preference of the network to learn different features in a dataset, an increase in non-linearity of the loss, and the existence of a learning shortcut that the network exploits to break single-step attacks. (iii) Building upon these insights, we describe and analyse a causal chain of events that can lead to CO. Overall, in this paper we show that:

Catastrophic overfitting can be a consequence of the interaction between easy- and hard-to-learn features in a dataset that can cause single-step AT methods to fail.

Our findings can improve our understanding of CO as they shift focus to the study of how the data influences AT. They can help circumvent the potential pitfalls of single-step AT and design effective and efficient AT methods. Moreover, they also pave the way to gain further insights in the intricate dynamics of robust model construction, where the interaction between robust and non-robust features plays a key role.

2. Why does catastrophic overfitting happen?

We first formulate our arguments by describing three key mechanisms, which together provide a plausible explanation for CO. In the rest of the paper, we provide extensive supporting evidence for our argument. Our starting point is a well known observation: while robust solutions can be attained with non-trivial changes to the standard training procedure, *e.g.,* using AT, they are not the default consequence of standard training. To that end, our first mechanism is concerned with the order at which features are learnt in a dataset (Kalimeris et al., 2019; Rahaman et al., 2019) during AT and how *robustly* separating the data may require additional information with respect to the one used in standard training (Sanyal et al., 2021; Montasser et al., 2019). More specifically, consider the setting where the data is comprised of easy-to-learn, discriminative, but non-robust features along with other harder-to-learn features that are necessary for robust classification. In the following, we will refer to them as *simple* or *complex* features, respectively, to reflect the intrinsic preferences of the network.

Mechanism 1 (Preference of the network – **M1**). *In the context of adversarial training in the setting described above, the network first learns the easy-to-learn features and then combines them with the other (more complex) features in an effort to increase its robustness. When the robustness requirement is lifted, the network defaults back to using just the initial non-robust features.*

Mechanism M1 conjectures that, during adversarial training, networks are biased towards first learning easy-to-learn features (Kalimeris et al., 2019; Shah et al., 2020) and then combining them with additional information, present in the complex features, in order to robustly separate the data. Note that this is a different phenomenon than the one observed in Shah et al. (2020), who only argue that during standard training, networks learn easy features and ignore complex features altogether even when trained for a large number of epochs. Moreover, Mechanism 1 is also implying that this is a forced behaviour of the network, *i.e.,* the network tends to forget this additional information as soon as robustness constraints are removed. Furthemore, in Appendix A we provide a rigorous proof showing that such a dichotomoy between clean and robust solutions can exist in certain learning problems.

Prior work (Andriushchenko and Flammarion, 2020; Kim et al., 2021) has observed that, when a network suffers from CO, the loss landscape becomes highly non-linear with respect to the input. In addition, some works (Fawzi et al., 2018; Jetley et al., 2018; Moosavi-Dezfooli et al., 2018) have also observed that the main curvature directions of the loss landscape are strongly correlated with the directions used to discriminate the data, at least on standard networks. Based on these facts, our second mechanism suggests that this increase in non-linearity is a consequence of the need for the network to fit the additional *complex* information required for robustness.

Mechanism 2 (Non-linear feature extraction – **M2**). *When combining different features to gain robustness, the network increases its non-linearity in order to learn representations that can exploit both features.*

Combining M1 and M2 provides an explanation for the increase of non-linearity AT, giving rise to a possible trigger for CO. However, it does not explain why this increase in non-linearity is worse for single-step AT than multi-step AT. Our final mechanism provides a plausible answer: It explains how the network exploits this trigger to identify a shortcut (Geirhos et al., 2020) which allows it to ignore the additional (complex) information needed for robustness.

Mechanism 3 (Non-linear shortcut – **M3**). *Catastrophic overfitting occurs when the increased non-linearity of the network hinders the single-step attacks from reliably approximating the inner maximization in Equation (1). With the single step attack rendered ineffective, the network creates a shortcut to focus only on the clean objective as long as it remains highly non-linear. This allows the network to only use the easy, non-robust features and ignore the additional robust information.*

Linking the three mechanisms mentioned above provides a plausible explanation for CO, in short:

Catastrophic overfitting is a consequence of the interaction between different features in a dataset (M1) which leads to an increase in non-linearity (M2) that causes single-step AT to fail (M3).

3. Inducing catastrophic overfitting

In Section 2, we argued that the root cause of CO stems from the bias of the network towards combining easy-to-learn features with other (more complex) features. However, directly identifying these two sets of features in a vision dataset is a difficult, if not impossible, task. Instead, as is standard practice in the field (Ilyas et al., 2019; Ortiz-Jimenez et al., 2020b; Arpit et al., 2017; Shah et al., 2020; Sanyal et al., 2021; Ortiz-Jimenez et al., 2020a), we rely on synthetic interventions that manipulate the data in order to make claims about its structure. In particular, we show that we can induce CO for FGSM-AT on a dataset that is synthetically intervened on and in a training regime that does not exhibit CO without the intervention *e.g.,* $\epsilon < 8/255$ during AT. Specifically, let $(\boldsymbol{x}, y) \sim \mathcal{D}$ be an image-label pair sample from a distribution $\mathcal{D}$. In order to synthetically induce the conditions in M1, we modify the *original* data $\boldsymbol{x}$ by adding an *injected* feature $\mathbf{y}$ that is strongly discriminative and easy-to-learn. Thus, we construct a family of *intervened datasets* $\widetilde{\mathcal{D}}_\beta$ such that

$$(\widetilde{\boldsymbol{x}}, y) \sim \widetilde{\mathcal{D}}_\beta : \quad \widetilde{\boldsymbol{x}} = \boldsymbol{x} + \beta\, \boldsymbol{v}(y) \quad \text{with} \quad (\boldsymbol{x}, y) \sim \mathcal{D}, \tag{3}$$

where $\boldsymbol{v}(y)$ is a label-dependent additive signal from a predefined set of linearly separable vectors $\mathcal{V} = \{\boldsymbol{v}(y) \mid y \in \mathcal{Y}\}$ such that $\|\boldsymbol{v}(y)\|_p = 1$ for all $y \in \mathcal{Y}$ and $\beta > 0$.

Properties of intervened dataset This construction has some interesting properties. Specifically, note that β controls the relative strength of the original and injected features, *i.e.,* $\boldsymbol{x}$ and $\mathbf{y}$, respectively. Since the injected features are linearly separable and perfectly correlated with the labels, a linear classifier can separate $\widetilde{\mathcal{D}}_\beta$ for a large enough β. Moreover, as β also controls the classification margin, if $\beta \gg \epsilon$ this classifier is also robust. However, if $\boldsymbol{x}$ has some components in $\mathrm{span}(\mathcal{V})$, the interaction between $\boldsymbol{x}$ and $\mathbf{y}$ may decrease the robustness of a linear classifier for some β. We rigorously illustrate such a behaviour for linear classifiers in Appendix B. In short, although $\mathbf{y}$ is easy-to-learn in general, the amount of additional information needed from $\boldsymbol{x}$ to achieve robustness will strongly depend on β.

With the aim to control such feature interactions, we design $\mathcal{V}$ by selecting vectors from the low-frequency components of the 2D Discrete Cosine Transform (DCT) (Ahmed et al., 1974) as these have a large alignment with the space of natural images that we use for our experiments (*e.g.,* CIFAR-10). Besides, and since CO has primarily been observed for ℓ_∞ perturbations, we binarize these vectors so that they only take values in ± 1, ensuring a maximal per-pixel perturbation that satisfies $\|\boldsymbol{v}(y)\|_\infty = 1$. The set $\mathcal{V}$ is illustrated in Figure 1(left). These two design constraints also help to visually identify the alignment of adversarial perturbations $\boldsymbol{\delta}$ with $\mathbf{y}$ as these patterns are visually distinctive (see Fig. 2).

Injection strength (β) drives CO To test the hypotheses in Section 2, we train a PreActResNet18 (He et al., 2016) on different intervened versions of CIFAR-10 (Krizhevsky and Hinton, 2009) using FGSM-AT for different robustness budgets ϵ and different β. Fig. 1 (right) shows a summary of these experiments both in terms of clean accuracy and robustness[1]. For clean accuracy, Fig. 1 (right) shows two distinct regimes. First, when $\beta < \epsilon$, the network achieves roughly the same accuracy by training and testing on $\widetilde{\mathcal{D}}_\beta$ as by training and testing on $\mathcal{D}$ (corresponding to $\beta = 0$). This is expected as FGSM does not suffer from CO in

[1]Meeasured using PGD with 50 iterations and 10 restarts.

Figure 2: Different samples of the intervened dataset $\widetilde{\mathcal{D}}_\beta$, and FGSM perturbations before and after CO. While prior to CO perturbations focus on the injected features, after CO they become noisy.

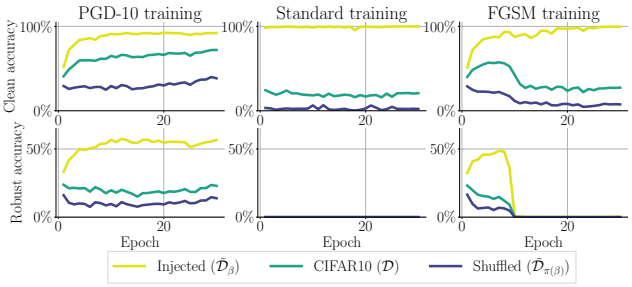


Figure 3: Clean (**top**) and robust (**bottom**) accuracy on 3 different test sets: (i) the original CIFAR-10 ($\mathcal{D}$), (ii) the dataset with injected features $\widetilde{\mathcal{D}}_\beta$ and (iii) the dataset with shuffled injected features $\widetilde{\mathcal{D}}_{\pi(\beta)}$. All training runs use $\beta = {}^8\!/_{255}$ and $\epsilon = {}^6\!/_{255}$ (where FGSM-AT suffers CO).

this setting (see Figure 1 (right)) and effectively ignores the added feature $\mathbf{y}$. Meanwhile, when $\beta > \epsilon$, the clean test accuracy is almost 100% indicating that the network heavily relies on the injected features.

The behaviour with respect to robust accuracy is more diverse. For small ϵ ($\epsilon = {}^2\!/_{255}$) the robust accuracy shows the same trend as the clean accuracy, albeit with lower values. For large ϵ ($\epsilon = {}^8\!/_{255}$), the model incurs CO for most values of β. This is not surprising as CO has already been reported for this value of ϵ on the original CIFAR-10 dataset (Wong et al., 2020). However, the interesting setting is for intermediate values of ϵ ($\epsilon \in \{{}^4\!/_{255}, {}^6\!/_{255}\}$). For these settings, Figure 1 (right) distinguishes between three distinct regimes. The first two regimes are the same as for $\epsilon = {}^2\!/_{255}$: (i) when the strength of the injected features is weak ($\beta \ll \epsilon$), the robust accuracy is similar to that trained on the original data ($\beta = 0$) and (ii) when it is strong ($\beta \gg \epsilon$), the robust accuracy is high as the network can use only $\mathbf{y}$ to classify $\widetilde{\boldsymbol{x}}$ robustly. Nevertheless, there is a third regime where the injected features are mildly robust, *i.e.,* $\beta \approx \epsilon$. Strikingly, in this regime, the training suffers from CO and the robust accuracy drops to zero. This is significant, since training on the original dataset $\mathcal{D}$ ($\beta = 0$) does not suffer from CO for this value of ϵ; but it does so when $\beta \approx \epsilon$. This observation aligns with the intuitions laid in the three mechanisms in Section 2 indicating that the interaction between the data and AT plays a big role in triggering CO.

We replicate these results for different $\mathcal{V}$'s and for ℓ_2 perturbations with similar findings in Appendix D. Results for other datasets and further details of the training protocol are given in Appendices C and D respectively. In the next section, we build upon these observations and show that the mechanisms in Section 2 indeed provide a plausible explanation for the dynamics that induce CO.

4. Analysis of induced catastrophic overfitting

Since we now have a method to intervene in the data using Equation (3) and induce CO, we can use it to validate our explanations for CO. In particular, we explore how the structure of the dataset features can lead to CO for FGSM-AT. First, we show that, indeed, the network combines information from both the easy-to-learn and the more complex features in order to improve robustness as described in M1. Then, we study the non-linearity of the network during AT and show how it leads to CO following M3; this chain of events is driven by the interaction between the features as per M2.

4.1. Robust solutions use simple and complex features

We now show that, in the regime where $\beta \approx \epsilon$, to achieve a high robust accuracy on the intervened dataset $\widetilde{\mathcal{D}}_\beta$, the network uses information from both the original dataset $\mathcal{D}$ and the injected features in $\mathcal{V}$. However, when trained without any adversarial constraints *i.e.,* for standard training, the network only uses the features in $\mathcal{V}$ and achieves close to perfect clean accuracy.

Testing the preference for different features In order to demonstrate this empirically, we perform standard, FGSM-AT, and PGD-AT training of a PreActResNet18 on the intervened dataset $\widetilde{\mathcal{D}}_\beta$ (as described in Section 3) with $\beta = {}^8\!/_{255}$ and $\epsilon = {}^6\!/_{255}$. First, note that Figure 1 (right) shows that an FGSM-AT model suffers from CO when trained on this intervened dataset. Next, we construct three different tests sets and evaluate the clean and robust accuracy of the networks on them in Figure 3. The three different test sets are: (i) CIFAR-10 test set with injected features ($\widetilde{\mathcal{D}}_\beta$), (ii) original CIFAR-10 test set ($\mathcal{D}$), and (iii) CIFAR-10 test set with shuffled injected features ($\widetilde{\mathcal{D}}_{\pi(\beta)}$) where the additive signals are correlated with a permuted set of labels, *i.e.,*

$$(\widetilde{\boldsymbol{x}}^{(\pi)}, y) \sim \widetilde{\mathcal{D}}_{\pi(\beta)} : \quad \widetilde{\boldsymbol{x}}^{(\pi)} = \boldsymbol{x} + \beta\, \boldsymbol{v}(\pi(y))$$

with $(\boldsymbol{x}, y) \sim \mathcal{D}$ and $\boldsymbol{v} \in \mathcal{V}$. Here, $\pi : \mathcal{Y} \to \mathcal{Y}$ is a fixed

permutation operator that shuffles the labels. Note that evaluating these networks (trained on $\widetilde{\mathcal{D}}_\beta$) on data from $\widetilde{\mathcal{D}}_{\pi(\beta)}$ exposes them to contradictory information, since $\boldsymbol{x}$ and $\boldsymbol{v}(\pi(y))$ are correlated with different labels. Thus, depending on which feature the networks relies on more, their performance will vary among the three datasets as discussed below.

Standard training and PGD Figure 3(left) shows that the PGD-trained network uses features from both $\mathcal{D}$ and $\mathcal{V}$. This is clear as the PGD-trained network achieves better than trivial accuracy on both $\mathcal{D}$, where there is no information coming from $\mathcal{V}$, as well as $\widetilde{\mathcal{D}}_{\pi(\beta)}$ where, by construction (see Section 4.1), the features from $\mathcal{V}$ are correlated with an incorrect label. Besides, the fact that the network achieves higher accuracy on samples from $\widetilde{\mathcal{D}}_\beta$ than on those of $\mathcal{D}$ implies that it also leverages $\mathcal{V}$ for classification. This provides evidence for mechanism M1. On the other hand, standard training shows a completely different behaviour (see Figure 3 (center)). In this case, even though the network achieves excellent clean accuracy on $\widetilde{\mathcal{D}}_\beta$, its accuracy on $\mathcal{D}$ is nearly trivial. Moreover, when asked to classify $\widetilde{\mathcal{D}}_{\pi(\beta)}$, its accuracy is almost zero. This clearly indicates that the network, obtained from standard training, ignores the information present in $\mathcal{D}$ and only uses the non-robust features from $\mathcal{V}$ for classification. As a result, the classifier, in this case, is non-robust. These two observations align with the idea that the network has a preference for easy-to-learn solutions *e.g.,* features in $\mathcal{V}$ (as shown by standard training) which, when not sufficient to classify robustly, are combined with more complex features *e.g.,* in $\mathcal{D}$ (as shown by PGD-AT).

FGSM training The behaviour of the FGSM training in Figure 3 (right) highlights this preference even further. First, note that FGSM-AT undergoes CO around epoch 10 when the robust accuracy on $\widetilde{\mathcal{D}}_\beta$ suddenly drops to zero despite a high clean accuracy on $\widetilde{\mathcal{D}}_\beta$. Next, as seen in Figure 3 (top right), FGSM-AT presents two distinct phases during training: (i) Prior to CO, when the robust accuracy on $\widetilde{\mathcal{D}}_\beta$ is non-zero, the network leverages features from both $\mathcal{D}$ and $\mathcal{V}$, as observed for PGD. (ii) However, with the onset of CO, both the clean and robust accuracy on $\mathcal{D}$ and $\widetilde{\mathcal{D}}_{\pi(\beta)}$ drops, exhibiting behavior similar to standard training. This indicates that, post-CO, the network forgets the information from $\mathcal{D}$ and solely relies on features in $\mathcal{V}$. To understand this behavior, recall mechanism M3. It suggests that when CO occurs, FGSM attacks are rendered ineffective thus effectively eliminating the robustness constraints. At that moment, as shown in Figure 3(right), around epoch 10, the network defaults back to using only the easy-to-learn features ($\mathbf{y} \in \mathcal{V}$) and performance on the original $\mathcal{D}$ drops.

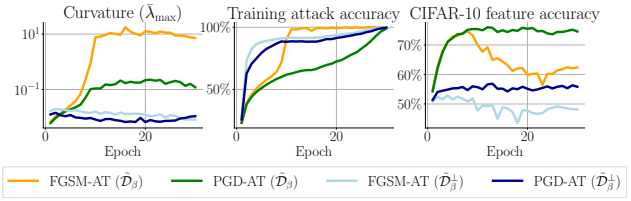


Figure 4: Different metrics on FGSM-AT and PGD-AT on 2 datasets: (i) with injected features ($\widetilde{\mathcal{D}}_\beta$) and (ii) with orthogonally projected features, i.e. with no interaction between the original and injected features ($\widetilde{\mathcal{D}}_\beta^\perp$). AT is performed for $\beta = 8/255$ and $\epsilon = 6/255$ (where FGSM suffers CO).

4.2. Curvature explosion drives catastrophic overfitting

We further investigate the mechanisms formulated in Section 2, and move on to the study of the non-linearity of the loss. In particular, we track the local curvature of the loss landscape during training as a strong proxy for non-linearity and show that, in line with mechanism M3, the non-linearity explodes during CO. Inspired by Moosavi-Dezfooli et al. (2019), we use the average maximum eigenvalue of the Hessian on $N = 100$ fixed training points $\bar{\lambda}_{\max} = \frac{1}{N}\sum_{n=1}^{N} \lambda_{\max}\left(\nabla^2_{\widetilde{\boldsymbol{x}}} \mathcal{L}(f_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{x}}_n), y_n)\right)$ to estimate curvature and record it throughout training. Fig. 4(left) shows the result of this experiment for FGSM-AT (orange line) and PGD-AT (green line) training on $\widetilde{\mathcal{D}}_\beta$ with $\beta = 8/255$ and $\epsilon = 6/255$. Recall that this training regime exhibits CO with FGSM-AT around epoch 10 (see Figure 3 (left)).

Curvature increase Interestingly, we observe that even before the onset of CO, both FGSM-AT and PGD-AT show a steep increase in curvature (the y-axis is in logarithmic scale). While the PGD-AT curvature increases rapidly before the 10^{th} epoch, it stabilizes soon after. Prior work has observed that PGD-AT acts as a regularizer on the curvature (Moosavi-Dezfooli et al., 2019; Qin et al., 2019) which explains why we observe that this curvature increase is eventually dampened in the PGD-AT run. Unlike PGD-AT, FGSM-AT is based on a linear (first order) approximation of the loss, which means that it is not effective at regularising the curvature, which is a second-order property of the loss surface. Indeed, we see that FGSM-AT cannot contain the curvature increase, which eventually explodes around the 10^{th} epoch and saturates at the moment that the training attack accuracy reaches its maximum. Quite remarkably, the final curvature of the FGSM-AT model is 100 times that of the PGD-AT model.

Meaningless perturbations The fact that the curvature increases rapidly during CO, when the attack accuracy also increases, agrees with the findings of Andriushchenko and Flammarion (2020), who claimed that CO happens as a result of gradient misalignment, *i.e.,* , the loss becomes highly non-linear and thus reduces the success rate of FGSM. To show that CO indeed occurs due to the increased curvature

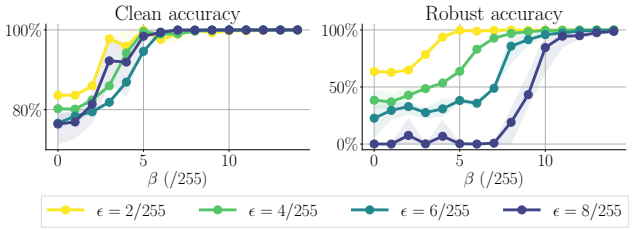


Figure 5: Clean (**left**) and robust (**right**) accuracy for FGSM-AT on a dataset with orthogonally injected features ($\widetilde{\mathcal{D}}_\beta^\perp$) i.e. no interaction between original and injected features.

breaking FGSM, we visualise the adversarial perturbations before and after CO. As observed in Figure 2, before CO, the adversarial perturbations point in the direction of $\mathcal{V}$, albeit with some corruptions originating from $\boldsymbol{x}$. Nonetheless, after CO, the new adversarial perturbations point towards meaningless directions; they do not align with $\mathcal{V}$ even though the network is heavily reliant on this information for classifying the data (cf. Section 4.1). This reinforces the idea that the increase in curvature indeed causes a breaking point after which FGSM is no longer an effective adversarial attack. We would like to highlight that this behaviour of the adversarial perturbations after CO is radically different from the behaviour on standard and robust networks (in the absence of CO) where adversarial perturbations and curvature are strongly aligned with discriminative directions (Ilyas et al., 2019; Jetley et al., 2018; Fawzi et al., 2018).

4.3. Curvature increase due to feature interaction

Why does the network increase the curvature in the first place? In Section 4.1, we observed that this is a shared behaviour of PGD-AT and FGSM-AT, at least during the stage before CO. Therefore, it should not be a mere "bug". As presented in mechanisms M1 and M2, we conjecture that the curvature increases as a result of the interaction between features of the dataset which forces the network to increase its non-linearity in order to combine them effectively to obtain a robust model.

To demonstrate this, we perform a new experiment in which we again intervene on the dataset $\mathcal{D}$ (as in Section 3). However, this time, we ensure that there is no interaction, *i.e.*, correlation, between the injected features $\boldsymbol{v}(y)$ and the features from $\mathcal{D}$. We do so by creating $\widetilde{\mathcal{D}}_\beta^\perp$ such that:

$$(\widetilde{\boldsymbol{x}}^\perp, y) \sim \widetilde{\mathcal{D}}_\beta^\perp : \quad \widetilde{\boldsymbol{x}}^\perp = \mathcal{P}_{\mathcal{V}^\perp}(\boldsymbol{x}) + \beta \boldsymbol{v}(y)$$

with $(\boldsymbol{x}, y) \sim \mathcal{D}$ and $\boldsymbol{v} \in \mathcal{V}$, and where $\mathcal{P}_{\mathcal{V}^\perp}$ denotes the projection operator onto the orthogonal complement of $\mathcal{V}$. Since the injected features $\boldsymbol{v}(y)$ are orthogonal to $\mathcal{D}$, a simple linear classifier relying only on $\boldsymbol{v}(y)$ can robustly separate the data up to a radius that depends solely on β.

Interestingly, we find that, in this dataset, none of the (β, ϵ) configurations used in Figure 5 induce CO. Here, we observe only two regimes: one that ignores $\mathcal{V}$ (when $\beta < \epsilon$) and the one that ignores $\mathcal{D}$ (when $\beta > \epsilon$). This supports our conjecture that the interaction between the features of $\boldsymbol{x}$ and $\boldsymbol{v}(y)$ is the true cause of CO in $\widetilde{\mathcal{D}}_\beta$. Moreover, Figure 4 (left) shows that, when performing either FGSM-AT (light blue) or PGD-AT (dark blue) on $\widetilde{\mathcal{D}}_\beta^\perp$, the curvature is consistently low. This agrees with the fact that in this case there is no need for the network to combine the injected and the original features to achieve robustness and hence the network does not need to increase its non-linearity to separate the data.

Non-linear feature extraction Finally, we perform an experiment to gauge the connection between the quality of feature representations and the network's curvature: We train multiple logistic regression models to classify $\mathcal{D}$ using the feature representations (output of the penultimate layer) of networks trained on $\widetilde{\mathcal{D}}_\beta$. Note that the accuracy of these simple classifiers strictly depends on how well the network (trained on $\widetilde{\mathcal{D}}_\beta$) has learned to combine information from both $\mathcal{D}$ and $\mathcal{V}$, as explained in M2. We will call this metric *feature accuracy*. Figure 4(right) shows the evolution of the feature accuracy of the networks during training. Observe that, for PGD-AT (green), the feature accuracy on $\mathcal{D}$ progressively grows during training. High feature accuracy indicates that this network has learned to meaningfully extract information from $\mathcal{D}$, even if it was trained on $\widetilde{\mathcal{D}}_\beta$. Moreover, note that the feature accuracy closely matches the curvature trajectory in Figure 4 (left). On the other hand, for FGSM-AT the feature accuracy has two phases: First, it grows at the same rate as for PGD-AT, but after CO it starts to decrease. Nonetheless, when CO happens the curvature does not decrease. We argue this happens because the network has learned a shortcut in order to ignore $\mathcal{D}$. As described in mechanism M3, if the curvature is very high, FGSM is rendered ineffective and allows the network to focus only on the easy-to-learn, non-robust, features. On the other hand, if we use the features from networks trained on $\widetilde{\mathcal{D}}_\beta^\perp$ we observe that the accuracy on $\mathcal{D}$ is always low. This reinforces the view that the network is increasing the curvature in order to improve its feature representation: In $\widetilde{\mathcal{D}}_\beta^\perp$ the network does not need to combine information from both $\mathcal{D}$ and $\mathcal{V}$ to become robust, and hence it does not learn to disentangle $\mathcal{D}$ using mechanism M2.

5. Concluding remarks

In this work, we have presented a thorough empirical study to establish a causal link between the features of the data and the onset of CO in FGSM-AT. Specifically, using controlled data interventions we have seen that, (i) when imposed with robustness contraints, networks have a preference to combine easy-to-learn, discriminative, but non-robust features with other (complex) features to achieve robustness. (ii) Moreover, if there is an interaction between these features the network tends to increase its non-linearity.

(iii) If unchecked, this increase in non-linearity can trigger CO. This new perspective has allowed us to shed new light on the mechanisms that trigger CO, as it shifted our focus towards studying the way the data structure influences the learning algorithm. We believe this opens the door to promising future work focused on understanding the intricacies of these learning mechanisms. In general, we consider that deriving methods that allow to inspect the data and identify how different features of a dataset interact within each other is another interesting avenue for future work.

Acknowledgements

We thank Maksym Andriushchenko, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli and Ricardo Volpi for the fruitful discussions and feedback. This work is supported by the UKRI grant: Turing AI Fellowship EP/W002981/1 and EPSRC/MURI grant: EP/N019474/1. We would also like to thank the Royal Academy of Engineering and FiveAI. Guillermo Ortiz-Jimenez acknowledges travel support from ELISE (GA no 951847) in the context of the ELLIS PhD Program. Amartya Sanyal acknowledges partial support from the ETH AI Center.

References

N. Ahmed, T. Natarajan, and K. R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, 1974.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 1994.

Adith Boloor, Xin He, Christopher D. Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Simple physical adversarial examples against end-to-end autonomous driving models. *arxiv:1903.05157*, 2019.

Pau de Jorge, Adel Bibi, Riccardo Volpi, Amartya Sanyal, Philip H. S. Torr, Grégory Rogez, and Puneet K. Dokania. Make some noise: Reliable and efficient single-step adversarial training. *arxiv:2202.01181*, 2022.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban. Zerograd: Mitigating and explaining catastrophic overfitting in fgsm adversarial training. *arXiv:2103.15476*, 2021.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Neural Information Processing Systems (NeurIPS)*, 2019.

Saumya Jetley, Nicholas Lord, and Philip Torr. With friends like these, who needs adversaries? *Neural Information Processing Systems (NeurIPS)*, 2018.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in neural information processing systems*, 32, 2019.

Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Annual Conference on Learning Theory (COLT)*, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard, and Stefano Soatto. Robustness of classifiers to universal perturbations: A geometric perspective. In *International Conference on Learning Representations (ICLR)*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems (NeurIPS), Workshops*, 2011.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi, and Pascal Frossard. Neural anisotropy directions. *Neural Information Processing Systems (NeurIPS)*, 2020a.

Guillermo Ortiz-Jimenez, Apostolos Modas, Seyed-Mohsen Moosavi, and Pascal Frossard. Hold me tight! influence of discriminative features on deep network boundaries. *Neural Information Processing Systems (NeurIPS)*, 2020b.

Geon Yeong Park and Sang Wan Lee. Reliably fast adversarial training via latent adversarial perturbation. In *International Conference on Learning Representations (ICLR), Workshops*, 2021.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

Amartya Sanyal, Puneet K. Dokania, Varun Kanade, and Philip Torr. How benign is benign overfitting ? In *International Conference on Learning Representations (ICLR)*, 2021.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Neural Information Processing Systems (NeurIPS)*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

BS Vivek and R Venkatesh Babu. Single-step adversarial training with dropout scheduling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.

A. Robust classification can require non-linear features

Given some $p \in \mathbb{N}$, let $\mathbb{R}^{p+1}$ be the input domain. A concept class, defined over $\mathbb{R}^{p+1}$ is a set of functions from $\mathbb{R}^{p+1}$ to $\{0,1\}$. We next define two properties of concepts in $\mathcal{H}$:

- A hypothesis h is s-non-linear if the polynomial with the smallest degree that can represent h has a degree (largest order polynomial term) of s.
- A hypothesis h is said to be *r-junta* if it depends on at most r coordinates of the input space.

Theorem 1. *For any $p, k \in \mathbb{N}, \epsilon < 0.5$ such that $k < p$, there exits a family of distributions $\mathcal{D}_k$ over $\mathbb{R}^{p+1}$ and a concept class $\mathcal{H}$ defined over $\mathbb{R}^{p+1}$ such that*

1. *$\mathcal{H}$ is PAC learnable (with respect to the clean error) with a* 1-junta linear (degree 1) *classifier.*
2. *There exists an efficient learning algorithm, that given a dataset sampled i.i.d. from a distribution $\mathcal{D} \in \mathcal{D}_k$ robustly learns $\mathcal{H}$. In particular, the algorithm returns a k-non-linear, k-junta classifier.*

Proof. We now define the construction of the distributions in $\mathcal{D}_k$. Every distribution $\mathcal{D}$ in the family of distribution $\mathcal{D}_k$ is uniquely defined by three parameters: a threshold parameter $\rho \in \{4t\epsilon : t \in \{0, \cdots, k\}\}$ (one can think of this as the non-robust, easy-to-learn feature) and a p dimensional bit vector $c \in \{0,1\}^p$ such that $\|c\|_1 = k$ (this is the non-linear but robust feature) and $epsilon$. Therefore, given ρ and c (and ϵ which we discuss when necessary and ignore from the notation for simplicity), we can define the distribution $\mathcal{D}^{c,\rho}$. For brevity, we eliminate ρ and c from the notation of the distribution.

To sample a point $(\mathbf{x}, y) \in \mathbb{R}^{p+1}$ from the distribution $D^{c,\rho}$, first, sample a random bit vector $\widehat{\mathbf{x}} \in \mathbb{R}^p$ from the uniform distribution over the boolean hypercube $\{0,1\}^p$. Let $\hat{y} = \sum_{i=1}^{p} \widehat{\mathbf{x}}_i \cdot c_i \pmod 2$ be the label of the parity function with respect to c evaluated on $\widehat{\mathbf{x}}$. The marginal distribution over $\widehat{y}$, if sampled this way, is equivalent to the Bernoulli distribution with parameter $\frac{1}{2}$. To see why, fix all bits in the input except one (chosen arbitrarily from the variables of the parity function), which is distributed uniformly over $\{0,1\}$. It is easy to see that this forces the output of the parity function to be distributed uniformly over $0, 1$ as well. Intuitively, $\widehat{\mathbf{x}}$ constitutes the robust non-linear feature of this distribution.

Next, we sample the non-robust simple feature x_1. To ensure that $\widehat{\mathbf{x}}$ is not perfectly correlated with the true label, we sample the true label y from a Bernoulli distribution with parameter $\frac{1}{2}$. Then we sample the non-robust feature $\mathbf{x}_1$ as follows

$$\mathbf{x}_1 \sim \begin{cases} \text{Unif}\left(X_1^-\right) & y = 0 \wedge \widehat{y} = 0 \\ \text{Unif}\left(X_1^+\right) & y = 1 \wedge \widehat{y} = 1 \\ \text{Unif}\left(X_2^-\right) & y = 0 \wedge \widehat{y} = 1 \\ \text{Unif}\left(X_2^+\right) & y = 1 \wedge \widehat{y} = 0 \end{cases}$$

where

$$X_1^+ = [\rho, \rho + \epsilon] \text{ and } \quad X_2^+ = [(\rho + 2\epsilon, \rho + 3\epsilon)], X_1^- = [\rho - \epsilon, \rho] \text{ and } \quad X_2^- = [(\rho - 3\epsilon, \rho - 2\epsilon)].$$

Finally, we return $(\mathbf{x}, y)$ where $\mathbf{x} = (x_1; \widehat{\mathbf{x}})$ is the concatenation of x_1 and $\widehat{\mathbf{x}}$ and y is already defined.

Linear non-robust solution It is simple to see that there is a linear, accurate, but non-robust solution to this problem. To obtain this solution, simply sample m points from the distribution $\mathcal{D}^{c,\rho}$ to form a dataset $S_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \in \mathbb{R}^{p+1} \times \{-1, 1\}$. Then, sort S_m on the basis of the first index of the covariates. In the sorted list, let the largest element whose label is 0 be $(\mathbf{x}_j, 0)$ and let $\hat{\rho} = \mathbf{x}_j[0]$ be the first coordinate of $\mathbf{x}_j$. Define $f_{\text{lin},\hat{\rho}}$as the linear threshold function on the first coordinate i.e. $f_{\text{lin},\hat{\rho}}(\mathbf{x}) = \mathbb{I}\{\mathbf{x}[0] \geq \hat{\rho}\}$. Then, using standard VC arguments about linear threshold functions in one dimension, if $m \geq \kappa_0 \left(\frac{1}{\alpha} \log\left(\frac{1}{\beta}\right) + \frac{1}{\alpha} \log\left(\frac{1}{\alpha}\right)\right)$, where κ_0 is some universal constant, we have that with probability at least $1 - \beta$,

$$\text{Err}\left(f_{\text{lin},\hat{\rho}}; \mathcal{D}^{c,\rho}\right) \leq \alpha,$$

where the bound follows from standard VC arguments on the linear threshold function.

Non-linear robust solution Next, we propose an algorithm to find a robust solution and show that this solution has a non-linearity of degree k. First sample m points from the distribution as before to create the dataset S_m, then use the method described above to find $\hat{\rho}$. We then create a modified dataset by first removing all points $\mathbf{x}$ from S_m such that $|\mathbf{x}[0] - \hat{\rho}| \geq \frac{\epsilon}{16}$ and then we remove the first coordinate of the remaining points to create a p dimensional dataset.

In this p dimensional dataset, the algorithm, then, learns the parity function using Gaussian elimination to obtain the parity bit vector $\hat{\mathbf{c}}$ and, consequently, the parity classifier $f_{\text{par},\hat{c}} = \sum_{i=1}^{p} \mathbf{x}_i \mathbf{c}_i \pmod 2$.

Finally, the algorithm returns the classifier $g_{\hat{\rho},\hat{\mathbf{c}}}$, which acts as follows:

$$g_{\hat{\rho},\hat{\mathbf{c}}}(\mathbf{x}) = \begin{cases} 1 & \mathbb{I}\left\{\mathbf{x}[0] \geq \rho + \epsilon + \frac{\epsilon}{8}\right\} \\ 0 & \mathbb{I}\left\{\mathbf{x}[0] \leq \rho - \epsilon - \frac{\epsilon}{8}\right\} \\ f_{\text{par},\hat{c}}(\widetilde{\mathbf{x}}) & \text{o.w.} \end{cases} \tag{4}$$

where $\widetilde{\mathbf{x}} = \text{round}(\mathbf{x}[1, \ldots, p+1])$ is obtained by rounding off $\mathbf{x}$ starting from the second index till the last. For example, if $\mathbf{x} = [0.99, 0.4, 0.9, 0.4, 0.8]$, $\epsilon = 0.2$, and $\mathbf{c} = [0, 0, 1, 1]$ then $\widetilde{\mathbf{x}} = [0, 1, 0, 1]$ and $g_{0.5,\hat{\mathbf{c}}}[\widetilde{x}] = 1$. Finally, it is easy to verify that the algorithm is accurate on all training points and as it has finite VC dimension, by standard VC arguments, as long as $m \geq \kappa_0 \left(\frac{1}{\alpha}\log\left(\frac{1}{\beta}\right) + \frac{1}{\alpha}\log\left(\frac{1}{\alpha}\right)\right)$, where κ_1 is some universal constant, we have that with probability at least $1 - \beta$,

$$\text{Err}\left(g_{\hat{\rho},\hat{c}}; \mathcal{D}^{c,\rho}\right) \leq \alpha.$$

As long as $\alpha \leq \frac{\epsilon}{16}$, due to the uniform distribution of $\mathbf{x}[1]$ in the interval $[\rho, \rho+\epsilon] \cup [\rho, \rho-\epsilon]$, we have that $|\hat{\rho} - \rho| \leq \frac{\epsilon}{8}$. Intuitively, this guarantees that $g_{\hat{\rho},\hat{\mathbf{c}}}$ uses $\mathbf{x}[0]$ in $[\rho, \rho+\epsilon] \cup [\rho, \rho-\epsilon]$ and $f_{\text{par},\hat{c}}$ in the $[\rho+2\epsilon, \rho+3\epsilon] \cup [\rho-2\epsilon, \rho-3\epsilon]$.

A crucial property of $g_{\hat{\rho},\hat{c}}$ is that for all $x \in \text{Supp}\left(\mathcal{D}^{c,\rho}\right)$, the classifier $g_{\hat{\rho},\hat{c}}$ does not alter its prediction in an ℓ_∞-ball of radius ϵ. When $|\mathbf{x}[0] - \rho| \geq \epsilon + \frac{\epsilon}{8}$, due to (4), we have that $g_{\hat{\rho},\hat{c}}$ is invariant to all $x[i]$ for all $i > 1$. When $|\mathbf{x}[0] - \rho| < \epsilon + \frac{\epsilon}{8}$, due to Equation (4), we have that $g_{\hat{\rho},\hat{c}} = f_{\text{par},\hat{c}}(\widetilde{\mathbf{x}})$ where $\widetilde{\mathbf{x}} = \text{round}(\mathbf{x}[1, \ldots, p+1])$ is obtained by rounding off all indices of $\mathbf{x}$ except the first. As the rounding operation on the boolean hypercube is robust to any ℓ_∞ perturbation of radius less than 0.5, we have that $g_{\hat{\rho},\hat{c}}$ is robust to all ℓ_∞ perturbations of radius less than 0.5 on the support of the distribution $\mathcal{D}^{c,\rho}$. Next, we prove the robustness along the first coordinate. Let $0 < \delta < 0.5$ represent an adversarial perturbation. Without loss of generality, assume that $x[0] > 0$ as similar arguments apply for when it is negative. When $|\mathbf{x}[0] - \rho| \leq \epsilon + \frac{\epsilon}{8}$, we also have that $|x - \delta| \leq \epsilon + \frac{\epsilon}{8}$ and hence, $g_{\hat{\rho},\hat{c}}(x) = g_{\hat{\rho},\hat{c}}(x - \delta)$. On the other hand, for all δ, we have that $g_{\hat{\rho},\hat{c}}(x + \delta) = 1$ if $g_{\hat{\rho},\hat{c}}(x + \delta) = 1$. This completes the proof of robustness of $g_{\hat{\rho},\hat{c}}$ along all dimensions to ℓ_∞ perturbations of radius less than ϵ. Combining this with its error bound, we have that $\text{Adv}_{\epsilon,\infty}\left(g_{\hat{\rho},\hat{c}}; \mathcal{D}^{c,\rho}\right) \leq \alpha$.

To show that the parity function is non-linear, we use a classical result from Aspnes et al. (1994). Theorem 2.2 in Aspnes et al. (1994) shows that approximating the parity function in k bits using a polynomial of degree ℓ incurs at least $\binom{\sum_{i=0}^{\lfloor k-\ell-1/2 \rfloor}}{ki}$ mistakes. Therefore, the lowest degree polynomial that can do the approximation accurately is at least k.

This completes our proof of showing that the robust classifier is of non-linear degree k while the accurate classifier is a linear. □

B. Analysis of the separability of the intervened data

With the aim to illustrate how the interaction between $\mathcal{D}$ and $\mathcal{V}$ can influence the robustness of a classifier trained on $\widetilde{\mathcal{D}}_\beta$ we now provide a toy theoretical example in which we discuss this interaction. Specifically, without loss of generality, consider the binary classification setting on the dataset $(\boldsymbol{x}, y) \sim \mathcal{D}$ where $y \in \{-1, +1\}$ and $\|\boldsymbol{x}\|_2 = 1$, for ease. Let's now consider the injected dataset $\widetilde{\mathcal{D}}_\beta$ and further assume that $\boldsymbol{v}(+1) = \boldsymbol{u}$ and $\boldsymbol{v}(-1) = -\boldsymbol{u}$ with $\boldsymbol{u} \in \mathbb{R}^d$ and $\|\boldsymbol{u}\|_2 = 1$, such that $\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \beta y \boldsymbol{u}$. Moreover, let $\gamma \in [0, 1]$ denote the *interaction coefficient* between $\mathcal{D}$ and $\mathcal{V}$, such that $-\gamma \leq \boldsymbol{x}^\top \boldsymbol{u} \leq \gamma$.

We are interested in characterizing the robustness of a classifier that only uses information in $\mathcal{V}$ when classifying $\widetilde{\mathcal{D}}_\beta$ depending on the strength of the interaction coefficient. In particular, as we are dealing with the binary setting, we will characterize the robustness of a linear classifier $h : \mathbb{R}^d \to \{-1, +1\}$ that discriminates the data based only on $\mathcal{V}$, *i.e.,*

$h(\widetilde{\boldsymbol{x}}) = \text{sign}(\boldsymbol{u}^\top \widetilde{\boldsymbol{x}})$. In our setting, we have

$$\boldsymbol{u}^\top \widetilde{\boldsymbol{x}} = \boldsymbol{u}^\top \boldsymbol{x} + \beta \boldsymbol{u}^\top \boldsymbol{u} = \boldsymbol{u}^\top \boldsymbol{x} + \beta \quad \text{if } y = +1$$
$$\boldsymbol{u}^\top \widetilde{\boldsymbol{x}} = \boldsymbol{u}^\top \boldsymbol{x} - \beta \boldsymbol{u}^\top \boldsymbol{u} = \boldsymbol{u}^\top \boldsymbol{x} - \beta \quad \text{if } y = -1$$

Proposition B.1 (Clean performance)**.** *If $\beta > \gamma$, then h achieves perfect classification accuracy on $\widetilde{\mathcal{D}}_\beta$.*

Proof. Observe that if $\gamma = 0$, i.e. the features from original dataset $\mathcal{D}$ do not interact with the injected features $\mathcal{V}$, the dataset is perfectly linearly separable. However, if the data $\boldsymbol{x}$ from $\mathcal{D}$ interacts with the injected signal $\boldsymbol{u}$, i.e. non zero projection, then the dataset is still perfectly separable but for a sufficiently larger β, such that $\boldsymbol{u}^\top \boldsymbol{x} + \beta > 0$ when $y = +1$ and $\boldsymbol{u}^\top \boldsymbol{x} + \beta < 0$ when $y = -1$. Because $-\gamma \leq \boldsymbol{x}^\top \boldsymbol{u} \leq \gamma$ this is achieved for $\beta > \gamma$. □

Proposition B.2 (Robustness)**.** *If $\beta > \gamma$, the linear classifier h is perfectly accurate and robust to adversarial perturbations in an ℓ_2-ball of radius $\epsilon \leq \beta - \gamma$. Or, equivalently, for h to be ϵ-robust, the injected features must have a strength $\beta \geq \epsilon + \gamma$.*

Proof. Given $\widetilde{\boldsymbol{x}}$, we seek to find the minimum distance to the decision boundary of such a classifier. A minimum distance problem can be cast as solving the following optimization problem:

$$\epsilon^\star(\widetilde{\boldsymbol{x}}) = \min_{\boldsymbol{r} \in \mathbb{R}^d} \|\boldsymbol{r} - \widetilde{\boldsymbol{x}}\|_2^2 \text{ subject to } \boldsymbol{r}^\top \boldsymbol{u} = 0,$$

which can be solved in closed form

$$\epsilon^\star(\widetilde{\boldsymbol{x}}) = \frac{|\boldsymbol{u}^\top \widetilde{\boldsymbol{x}}|}{\|\boldsymbol{u}\|} = |\boldsymbol{u}^\top \boldsymbol{x} + y\beta|.$$

The robustness radius of the classifier h will therefore be $\epsilon = \inf_{\widetilde{\boldsymbol{x}} \in \text{supp}(\widetilde{\mathcal{D}}_\beta)} \epsilon^\star(\widetilde{\boldsymbol{x}})$, which in our case can be bounded by

$$\epsilon = \inf_{(\widetilde{\boldsymbol{x}}, y) \in \text{supp}(\widetilde{\mathcal{D}}_\beta)} \epsilon^\star(\widetilde{\boldsymbol{x}}) \leq \min_{|\boldsymbol{u}^\top \boldsymbol{x}| \leq \gamma, y = \pm 1} |\boldsymbol{u}^\top \boldsymbol{x} + y\beta| = |\mp\gamma \pm \beta| = \beta - \gamma.$$

□

Based on these propositions, we can clearly see that the interaction coefficient γ reduces the robustness of the additive features $\mathcal{V}$. In this regard, if $\epsilon \geq \beta - \gamma$, robust classification at a radius ϵ can only be achieved by also leveraging information within $\mathcal{D}$.

C. Experimental details

In this section we provide the experimental details for all results presented in the paper. Adversarial training for all methods and datasets follows the fast training schedules with a cyclic learning rate introduced in (Wong et al., 2020). We train for 30 epochs on CIFAR (Krizhevsky and Hinton, 2009) and 15 epochs for SVHN (Netzer et al., 2011) following (Andriushchenko and Flammarion, 2020). When we perform PGD-AT we use 10 steps and a step size $\alpha = 2/255$; FGSM uses a step size of $\alpha = \epsilon$. Regularization parameters for GradAlign (Andriushchenko and Flammarion, 2020) and N-FGSM (de Jorge et al., 2022) will vary and are stated when relevant in the paper. The architecture employed is a PreactResNet18 (He et al., 2016). Robust accuracy is evaluated by attacking the trained models with PGD-50-10. That is PGD with 50 iterations and 10 restarts. In this case we also employ a step size of $2/255$ as in (Wong et al., 2020). All accuracies are averaged after training and evaluating with 3 random seeds.

The curvature computation is performed following the procedure described in Moosavi-Dezfooli et al. (2019). As they propose, we use finite differences to estimate the directional second derivative of the loss with respect to the input, *i.e.,*

$$\boldsymbol{w}^\top \nabla_{\boldsymbol{x}}^2 \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y) \approx \frac{\nabla_{\boldsymbol{x}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x} + t\boldsymbol{w}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x} - t\boldsymbol{w}), y)}{2t},$$

with $t > 0$ and use the Lanczos algorithm to perform a partial eigendecomposition of the Hessian without the need to compute the full matrix. In particular, we pick $t = 0.1$.

All our experiments were performed using a cluster equipped with GPUs of various architectures. The estimated compute budget required to produce all results in this work is around $2,000$ GPU hours (in terms of NVIDIA V100 GPUs).

D. Inducing catastrophic overfitting with other settings

In Section 3 we have shown that CO can be induced with data interventions for CIFAR-10 and ℓ_∞ perturbations. Here we present similar results when using other datasets (i.e. CIFAR-100 and SVHN) and other types of perturbations (i.e. ℓ_2 attacks). Moreover, we also report results when the injected features $\mathbf{y}$ follow random directions (as opposed to low-frequency DCT components). Overall, we find similar findings to those reported the main text.

D.1. Other datasets

Similarly to Section 3 we intervene the SVHN and CIFAR-100 datasets to inject highly discriminative features $\mathbf{y}$. Since SVHN also has 10 classes, we use the exact same settings as in CIFAR-10 and we train and evaluate with $\epsilon = 4$ where training on the original data does not lead to CO (recall $\beta = 0$ corresponds to the unmodified dataset). On the other hand, for CIFAR-100 we select $\mathbf{y}$ to be the 100 DCT components with lowest frequency and we present results with $\epsilon = 5$. In both datasets we can observe similar trends as with CIFAR-10. For small values of β the injected features are not very discriminative due to their interaction with the dataset images and the model largely ignores them. As we increase β, there is a range in which they become more discriminative but not yet robust and we observe CO. Finally for large values of β the injected features become robust and the models can achieve very good performance focusing only on those.

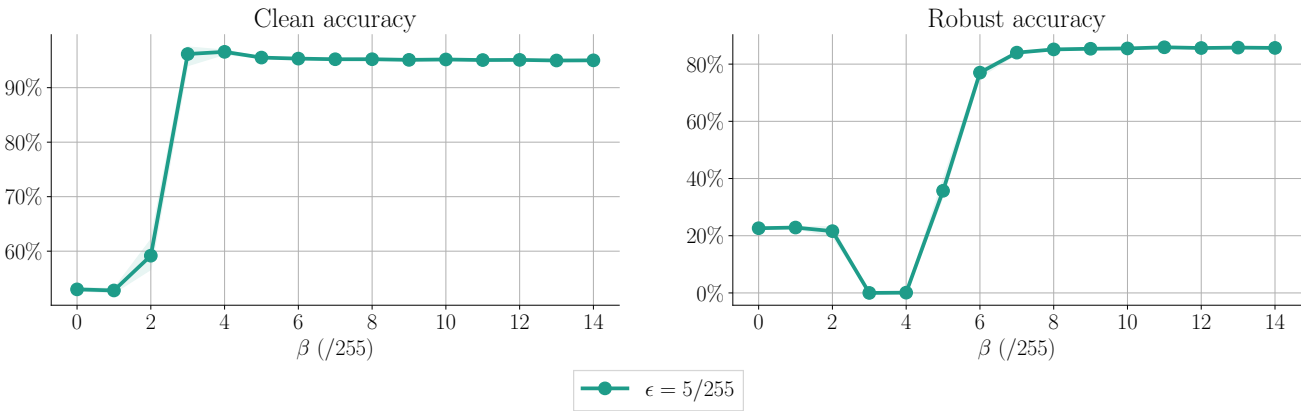


Figure 6: Clean and robust performance after FGSM-AT on intervened datasets $\widetilde{\mathcal{D}}_\beta$ constructed from CIFAR-100. As FGSM-AT already suffers CO on CIFAR-100 at $\epsilon = 6/255$ we use $\epsilon = 5/255$ in this experiment where FGSM-AT does not suffer from CO as seen for $\beta = 0$. In this setting, we observe CO happening when β is slightly smaller than ϵ. Results are averaged over 3 seeds and shaded areas report minimum and maximum values.

D.2. Other norms

Catastrophic overfitting has been mainly studied for ℓ_∞ perturbations and thus we presented experiments with ℓ_∞ attacks following related work. However, in this section we also present results where we induce CO with ℓ_2 perturbation which are also widely used in adversarial robustness. In Figure 8 we show the clean (left) and robust (right) accuracy after FGM-AT[2] on our intervened dataset from CIFAR-10 ($\widetilde{\mathcal{D}}_\beta$). Similarly to our results with ℓ_∞ attacks, we also observe CO as the injected features become more discriminative (increased β). It is worth mentioning that the ℓ_2 norm we use ($\epsilon = 1.5$) is noticeably larger than typically used in the literature, however, it would roughly match the magnitude of an ℓ_∞ perturbation with $\epsilon = 7/255$. Interestingly, we did not observe CO for this range of β with $\epsilon = 1$.

[2]FGM is the ℓ_2 version of FGSM where we do not take the sign of the gradient.

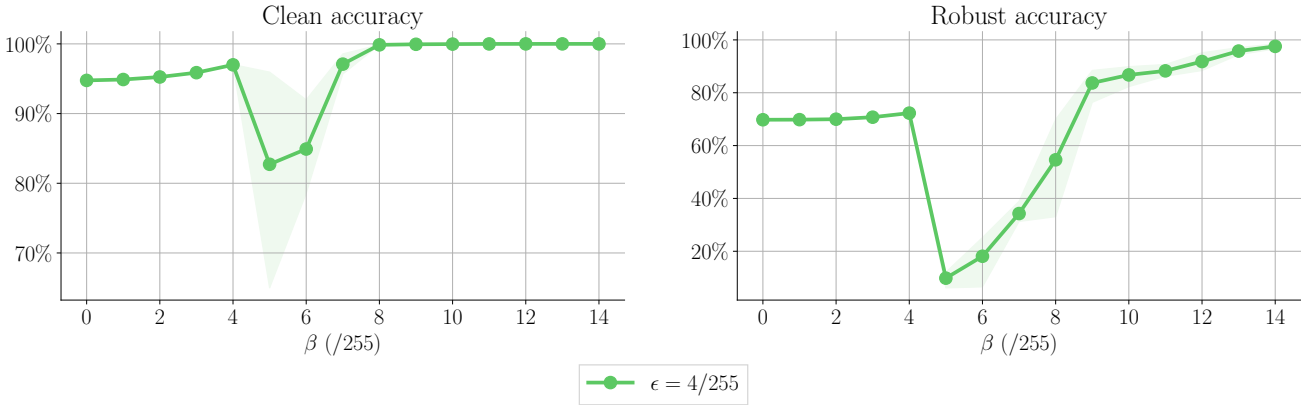


Figure 7: Clean and robust performance after FGSM-AT on intervened datasets $\widetilde{\mathcal{D}}_\beta$ constructed from SVHN. As FGSM-AT already suffers CO on SVHN at $\epsilon = 6/255$ we use $\epsilon = 4/255$ in this experiment where FGSM-AT does not suffer from CO as seen for $\beta = 0$. In this setting, we observe CO happening when $\beta \approx \epsilon$. Results are averaged over 3 seeds and shaded areas report minimum and maximum values.

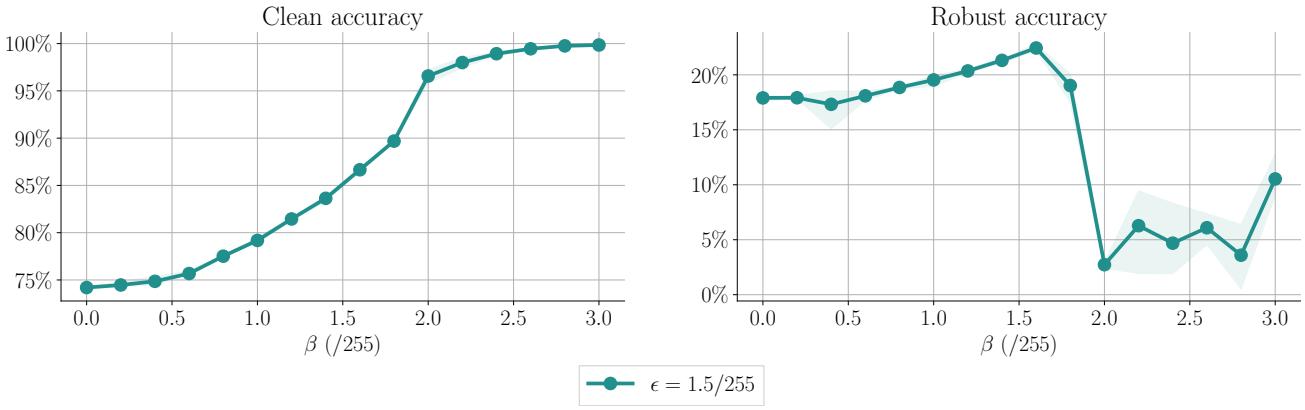


Figure 8: Clean and ℓ_2 robust performance after FGSM-AT on intervened datasets $\widetilde{\mathcal{D}}_\beta$ constructed from CIFAR-10. FGM-AT suffers CO on CIFAR-10 around $\epsilon = 2$, so we use $\epsilon = 1.5$ in this experiment where FGM-AT does not suffer from CO as seen for $\beta = 0$. In this setting, we observe CO happening when $\beta \approx \epsilon$. Results are averaged over 3 seeds and shaded areas report minimum and maximum values.

D.3. Other injected features

We selected the injected features for our intervened dataset from the low frequency components of the DCT to ensure an interaction with the features present on natural images (Ahmed et al., 1974). However, this does not mean that other types of features could not induce CO. In order to understand how unique was our choice of features we also created another family of intervened datasets but this time using a set of 10 randomly generated vectors as features. As in the main text, we take the sign of each random vector to ensure they take values in $\{-1, +1\}$ and assign one vector per class. In Figure 9 we observe that using random vectors as injected features we can also induce CO. Note that since our results are averaged over 3 random seeds, each seed uses a different set of random vectors.

E. Learned features at different β

In Section 3 we discussed how, based on the strength of the injected features β, our intervened datasets seem to have 3 distinct regimes: (i) When β is small we argued that the network would not use the injected features as these would not be very helpful. (ii) When β would have a very large value then the network would only look at these features since they would be easy-to-learn and provide enough margin to classify robustly. (iii) Finally, there was a middle range of β usually when $\beta \sim \epsilon$ where the injected features would be strongly discriminative but not enough to provide robustness on their own. This

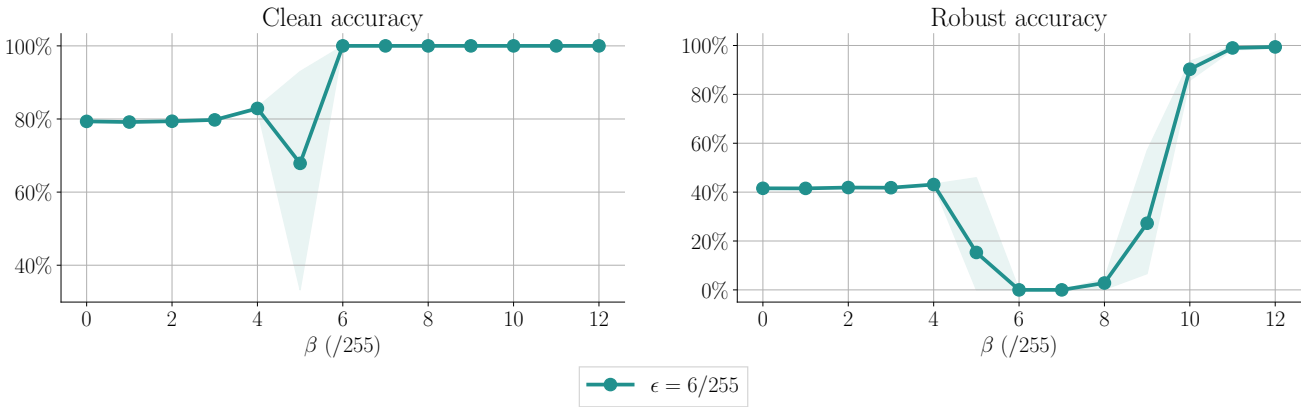


Figure 9: Clean and robust performance after FGSM-AT on intervened datasets $\widetilde{\mathcal{D}}_\beta$ constructed from CIFAR-10 using random signals in $\mathcal{V}$. We perform this experiments at $\epsilon = 6/255$ where we saw intervening the dataset with the DCT basis vectors did induce CO. In the random $\mathcal{V}$ setting, we observe the same behaviour, with CO happening when $\beta \approx \epsilon$. Results are averaged over 3 seeds and shaded areas report minimum and maximum values.

regime is where we observe CO.

In this section we present an extension of Figure 3 where we take FGSM trained models on the intervened datasets ($\widetilde{\mathcal{D}}_\beta$) and evaluate them on three test sets: (i) The injected test set ($\widetilde{\mathcal{D}}_\beta$) with the same features as the training set. (ii) The original dataset ($\mathcal{D}$) where the images are unmodified. (iii) The shuffled dataset ($\widetilde{\mathcal{D}}_{\pi(\beta)}$) where the injected features are permuted. That is, the set of injected features is the same but the class assignments are shuffled. Therefore, the injected features will provide conflicting information with the features present on the original image.

In Figure 10 we show the performance on the aforementioned datasets for three different values of β. For $\beta = 2/255$ we are in regime (i) : we observe that the tree datasets have the same performance, i.e. the information of the injected features does not seem to alter the result. Therefore, we can conclude the network is mainly using the features from the original dataset $\mathcal{D}$. When $\beta = 20/255$ we are in regime (ii) : the clean and robust performance of the network is almost perfect on the injected test set $\widetilde{\mathcal{D}}_\beta$ while it is close to 0% (note this is worse than random classifier) for the shuffled dataset. So when the injected and original features present conflicting information the network aligns with the injected features. Moreover, the performance on the original dataset is also very low. Therefore, the network is mainly using the injected features. Lastly, $\beta = 8/255$ corresponds to regime (iii) : as discussed in Section 4.1, in this regime the network initially learns to combine information from both the original and injected features. However, after CO, the network seems to focus only on the injected features and discards the information from the original features.

F. Analysis of curvature in different settings

In Figure 4 (left) we track the curvature of the loss surface while training on different intervened datasets with either PGD-AT or FGSM-AT. We observe that (i) Curvature rapidly increases both for PGD-AT and FGSM-AT during the initial epochs of training. (ii) In those runs that presented CO, the curvature explodes around the 10^{th} epoch along with the training accuracy. (iii) When training with the dataset with orthogonally injected features ($\widetilde{\mathcal{D}}_\beta^\perp$) the curvature does not increase. This is aligned with our proposed mechanisms to induce CO whereby the network increases the curvature in order to combine different features to learn better representations. In this section we extend this analysis to the original CIFAR-10 dataset (as opposed to our intervened datasets) and to different values of feature strength β on the intervened dataset ($\widetilde{\mathcal{D}}_\beta$). For details on how we estimate the curvature refer to Appendix C.

In Figure 11 we show the curvature when training on the original CIFAR-10 dataset with $\epsilon = 8/255$ (where CO happens for FGSM-AT). Similarly to our observations on the intervened datasets, the curvature during FGSM-AT explodes along with the training accuracy while for PGD-AT the curvature increases at a very similar rate than FGSM-AT during the first epochs and later stabilizes. This indicates that our described mechanisms may as well apply to induce CO on natural image datasets.

On the other hand, Figure 12 presents the curvature for different values of feature strength β on the intervened dataset ($\widetilde{\mathcal{D}}_\beta$).

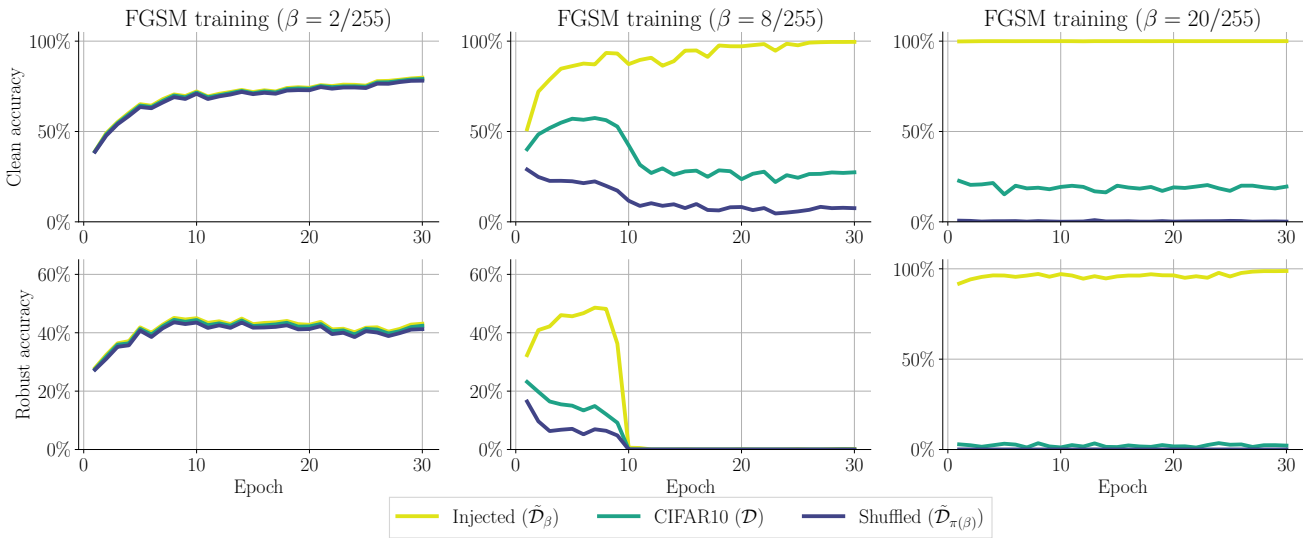


Figure 10: Clean (**top**) and robust (**bottom**) accuracy of FGSM-AT on $\widetilde{\mathcal{D}}_\beta$ at different β values on 3 different test sets: (i) the original CIFAR-10 ($\mathcal{D}$), (ii) the dataset with injected features $\widetilde{\mathcal{D}}_\beta$ and (iii) the dataset with shuffled injected features $\widetilde{\mathcal{D}}_{\pi(\beta)}$. All training runs use $\epsilon = 6/255$. **Left**: $\beta = 2/255$ **Center**: $\beta = 8/255$ **Right**: $\beta = 20/255$.

We show three different values of β representative of the three regimes discussed in Appendix E. Recall that when β is small ($\beta = 2/255$) we observe that the model seems to focus only on CIFAR-10 features. Thus, we observe a curvature increase aligned with (CIFAR-10) feature combination. However, since for the chosen robustness radii $\epsilon = 6/255$ there is no CO, we observe that the curvature increase remains stable. When β is quite large ($\beta = 20/255$) then the model largely ignores CIFAR-10 information and focuses on the easy-to-learn injected features. Since these features are already robust, there is no need to combine them and the curvature does not need to increase. In the middle range when CO happens ($\beta = 8/255$) we observe again the initial curvature increase and then curvature explosion.

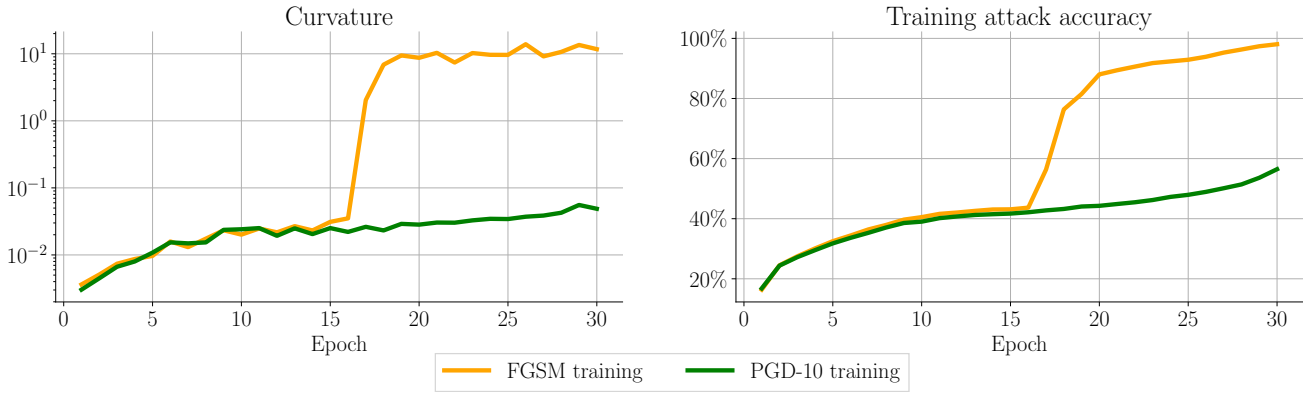


Figure 11: Evolution of curvature and training attack accuracy of FGSM-AT and PGD-AT trained on the original CIFAR-10 with $\epsilon = 8/255$. When CO happens the curvature explodes according to mechanism M3.

G. Adversarial perturbations before and after CO

In order to further understand the change in behaviour after CO we presented visualizations of the FGSM perturbations before and after CO in Figure 2. We observed that while prior to CO, the injected feature components $\mathbf{y}$ were clearly identifiable, after CO the perturbations do not seem to point in those directions although the network is strongly relying on them to classify. In Figure 13 and Figure 14 we show further visualizations of the perturbations obtained both with FGSM or PGD attacks on networks trained with either PGD-AT or FGSM-AT respectively.

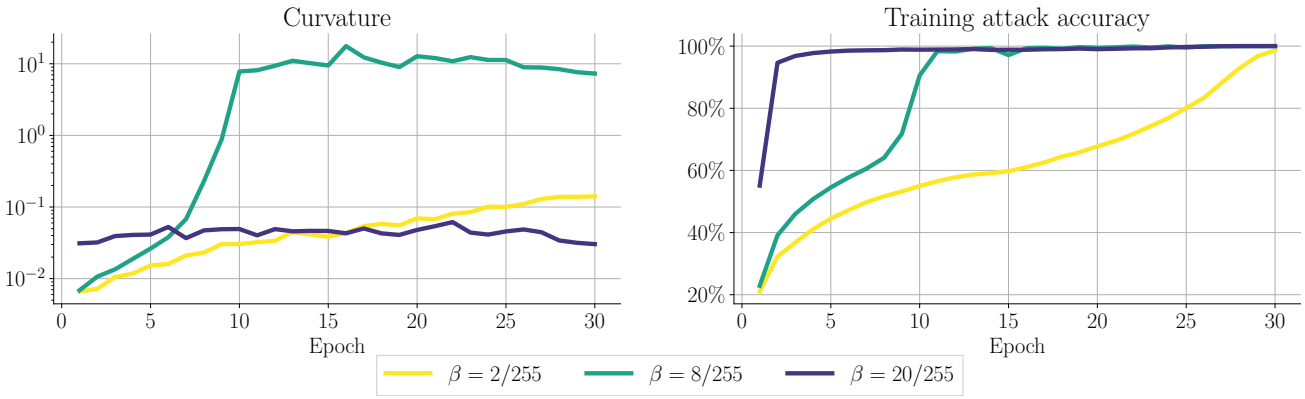
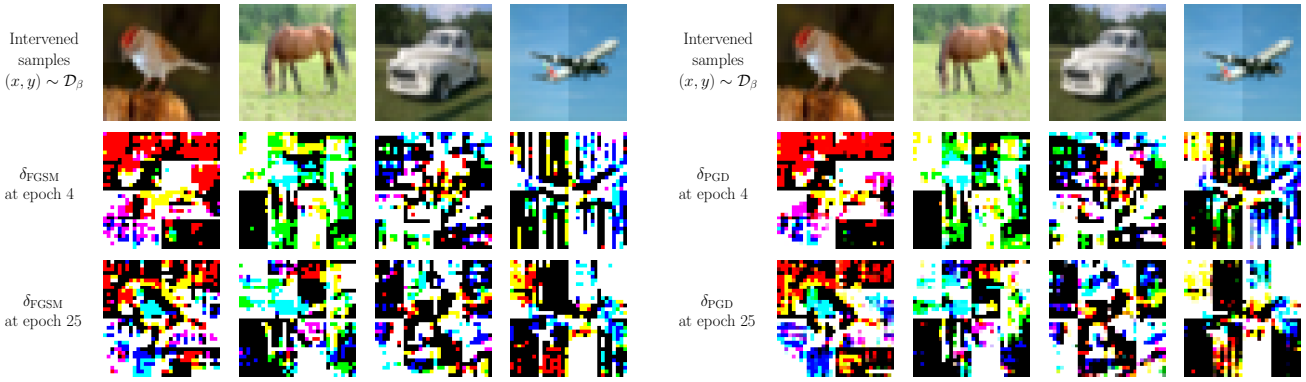


Figure 12: Evolution of curvature and training attack accuracy of FGSM-AT and PGD-AT trained on $\widetilde{\mathcal{D}}_\beta$ at different β and for $\epsilon = 6/255$. Only when CO happens (for $\beta = 8/255$) the curvature explodes according to mechanism M3. For the other two interventions, because the network does not need to disentangle $\mathcal{D}$ from $\mathcal{V}$, as it ignores either one of them, the curvature does not increase so much.

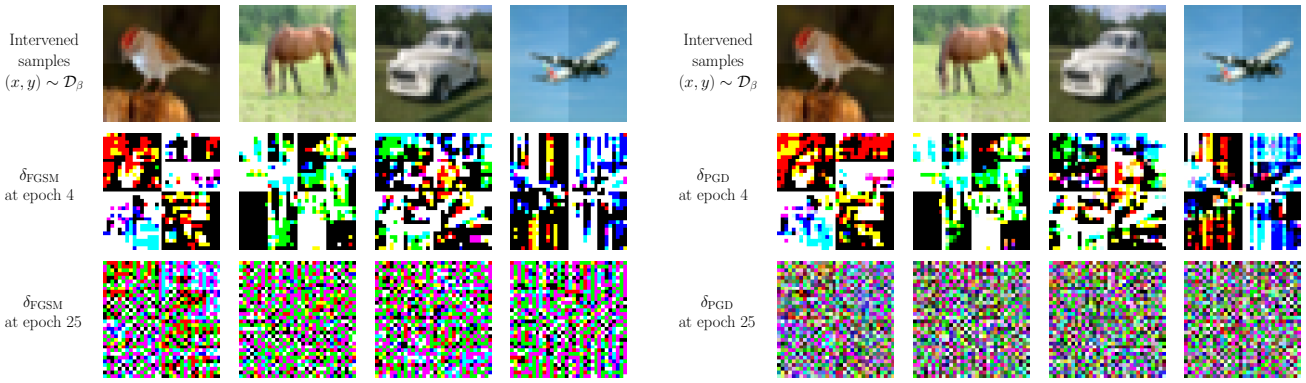
We observe that when training with PGD-AT, i.e. the training does not suffer from CO, both PGD and FGSM attacks produce qualitatively similar results. In particular, all attacks seem to target the injected features with some noise due to the interaction with the features from CIFAR-10. For FGSM-AT, we observe that at the initial epochs (prior to CO) the pertubations are similar to those of PGD-AT, however, after CO perturbations change dramatically both for FGSM and PGD attacks. This aligns with the fact that the loss landscape of the network has dramatically changed, becoming strongly non-linear. This change yields single-step FGSM ineffective, however, the network remains vulnerable and multi-step attacks such as PGD are still able to find adversarial examples, which in this case do not point in the direction of discriminative features Jetley et al. (2018); Ilyas et al. (2019).



(a) FGSM adversarial perturbations

(b) PGD adversarial perturbations

Figure 13: Different samples of the intervened dataset $\widetilde{\mathcal{D}}_\beta$, and adversarial perturbations at epoch 4 and 22 of PGD-AT on $\widetilde{\mathcal{D}}_\beta$ at $\epsilon = 6/255$ and $\beta = 8/255$ (where FGSM-AT suffers CO). The adversarial perturbations remain qualitatively similar throughout training and align significantly with $\mathcal{V}$.



(a) FGSM adversarial perturbations

(b) PGD adversarial perturbations

Figure 14: Different samples of the intervened dataset $\widetilde{\mathcal{D}}_\beta$, and adversarial perturbations at epoch 4 (before CO) and 22 (after CO) of FGSM-AT on $\widetilde{\mathcal{D}}_\beta$ at $\epsilon = {}^6\!/_{255}$ and $\beta = {}^8\!/_{255}$ (where FGSM-AT suffers CO). The adversarial perturbations change completely before and after CO. Prior to CO, they align significantly with $\mathcal{V}$, but after CO they point to meaningless directions.