Adversarial Training and Robustness for Multiple Perturbations

Florian Tramèr
Stanford University

Dan Boneh
Stanford University

Abstract

Defenses against adversarial examples, such as adversarial training, are typically tailored to a single perturbation type (e.g., small ℓ_∞-noise). For other perturbations, these defenses offer no guarantees and, at times, even increase the model's vulnerability. Our aim is to understand the reasons underlying this robustness trade-off, and to train models that are simultaneously robust to multiple perturbation types.

We prove that a trade-off in robustness to different types of ℓ_p-bounded and spatial perturbations must exist in a natural and simple statistical setting. We corroborate our formal analysis by demonstrating similar robustness trade-offs on MNIST and CIFAR10. We propose new multi-perturbation adversarial training schemes, as well as an efficient attack for the ℓ_1-norm, and use these to show that models trained against multiple attacks fail to achieve robustness competitive with that of models trained on each attack individually. In particular, we find that adversarial training with first-order ℓ_∞, ℓ_1 and ℓ_2 attacks on MNIST achieves merely 50% robust accuracy, partly because of gradient-masking. Finally, we propose *affine attacks* that linearly interpolate between perturbation types and further degrade the accuracy of adversarially trained models.

1 Introduction

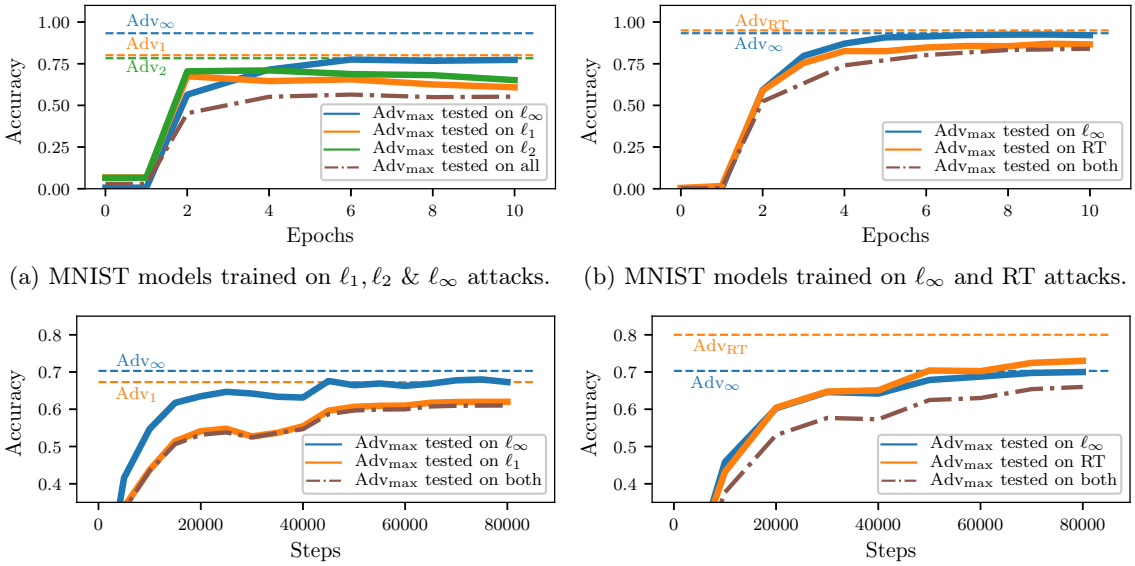
Adversarial examples [37, 15] are proving to be an inherent blind-spot in machine learning (ML) models. Adversarial examples highlight the tendency of ML models to learn superficial and brittle data statistics [19, 13, 18], and present a security risk for models deployed in cyber-physical systems (e.g., virtual assistants [5], malware detectors [16] or ad-blockers [39]).

Known successful defenses are tailored to a specific perturbation type (e.g., a small ℓ_p-ball [25, 28, 42] or small spatial transforms [11]). These defenses provide empirical (or certifiable) robustness guarantees for one perturbation type, but typically offer no guarantees against other attacks [35, 31]. Worse, increasing robustness to one perturbation type has sometimes been found to increase vulnerability to others [11, 31]. This leads us to the central problem considered in this paper:

Can we achieve adversarial robustness to different types of perturbations simultaneously?

Note that even though prior work has attained robustness to different perturbation types [25, 31, 11], these results may not compose. For instance, an ensemble of two classifiers—each of which is robust to a single type of perturbation—may be robust to neither perturbation. Our aim is to study the extent to which it is possible to learn models that are *simultaneously* robust to multiple types of perturbation.

To gain intuition about this problem, we first study a simple and natural classification task, that has been used to analyze trade-offs between standard and adversarial accuracy [41], and the sample-complexity of adversarial generalization [30]. We define *Mutually Exclusive Perturbations (MEPs)* as pairs of perturbation types for which robustness to one type implies vulnerability to the other. For this task, we prove that ℓ_∞ and ℓ_1-perturbations are MEPs and that ℓ_∞-perturbations and input rotations and translations [11] are also MEPs. Moreover, for these MEP pairs, we find that robustness to either perturbation type requires fundamentally different features. The



(a) MNIST models trained on ℓ_1, ℓ_2 & ℓ_∞ attacks.

(b) MNIST models trained on ℓ_∞ and RT attacks.

(c) CIFAR10 models trained on ℓ_1 and ℓ_∞ attacks.

(d) CIFAR10 models trained on ℓ_∞ and RT attacks.

Figure 1: **Robustness trade-off on MNIST (top) and CIFAR10 (bottom).** For a union of ℓ_p-balls (left), or of ℓ_∞-noise and rotation-translations (RT) (right), we train models $\text{Adv}_{\max}$ on the strongest perturbation-type for each input. We report the test accuracy of $\text{Adv}_{\max}$ against each individual perturbation type (solid line) and against their union (dotted brown line). The vertical lines show the adversarial accuracy of models trained and evaluated on a single perturbation type.

existence of such a trade-off for this simple classification task suggests that it may be prevalent in more complex statistical settings.

To complement our formal analysis, we introduce new adversarial training schemes for multiple perturbations. For each training point, these schemes build adversarial examples for all perturbation types and then train either on all examples (the "avg" strategy) or only the worst example (the "max" strategy). These two strategies respectively minimize the *average* error rate across perturbation types, or the error rate against an adversary that picks the worst perturbation type for each input.

For adversarial training to be practical, we also need efficient and strong attacks [25]. We show that Projected Gradient Descent [22, 25] is inefficient in the ℓ_1-case, and design a new attack, *Sparse* ℓ_1 *Descent* (SLIDE), that is both efficient and competitive with strong optimization attacks [8],

We experiment with MNIST and CIFAR10. MNIST is an interesting case-study, as *distinct* models from prior work attain strong robustness to all perturbations we consider [25, 31, 11], yet no *single* classifier is robust to all attacks [31, 32, 11]. For models trained on multiple ℓ_p-attacks ($\ell_1, \ell_2, \ell_\infty$ for MNIST, and ℓ_1, ℓ_∞ for CIFAR10), or on both ℓ_∞ and spatial transforms [11], we confirm a noticeable robustness trade-off. Figure 1 plots the test accuracy of models $\text{Adv}_{\max}$ trained using our "max" strategy. In all cases, robustness to multiple perturbations comes at a cost—usually of 5-10% additional error—compared to models trained against each attack individually (the horizontal lines).

Robustness to ℓ_1, ℓ_2 and ℓ_∞-noise on MNIST is a striking failure case, where the robustness trade-off is compounded by *gradient-masking* [27, 40, 1]. Extending prior observations [25, 31, 23], we show that models trained against an ℓ_∞-adversary learn representations that *mask gradients* for attacks in other ℓ_p-norms. When trained against first-order ℓ_1, ℓ_2 and ℓ_∞-attacks, the model learns to resist ℓ_∞-attacks while giving the illusion of robustness to ℓ_1 and ℓ_2 attacks. This model only achieves 52% accuracy when evaluated on gradient-free attacks [3, 31]. This shows that, unlike previously thought [41], adversarial training with strong first-order attacks can suffer

from gradient-masking. We thus argue that attaining robustness to ℓ_p-noise on MNIST requires new techniques (e.g., training on expensive gradient-free attacks, or scaling certified defenses to multiple perturbations).

MNIST has sometimes been said to be a poor dataset for evaluating adversarial examples defenses, as some attacks are easy to defend against (e.g., input-thresholding or binarization works well for ℓ_∞-attacks [41, 31]). Our results paint a more nuanced view: the simplicity of these ℓ_∞-defenses becomes a disadvantage when training against multiple ℓ_p-norms. We thus believe that MNIST should not be abandoned as a benchmark just yet. Our inability to achieve multi-ℓ_p robustness for this simple dataset raises questions about the viability of scaling current defenses to more complex tasks.

Looking beyond adversaries that choose from a union of perturbation types, we introduce a new *affine adversary* that may linearly interpolate between perturbations (e.g., by compounding ℓ_∞-noise with a small rotation). We prove that for locally-linear models, robustness to a union of ℓ_p-perturbations implies robustness to affine attacks. In contrast, affine combinations of ℓ_∞ and spatial perturbations are provably stronger than either perturbation individually. We show that this discrepancy translates to neural networks trained on real data. Thus, in some cases, attaining robustness to a union of perturbation types remains insufficient against a more creative adversary that composes perturbations.

Our results show that despite recent successes in achieving robustness to single perturbation types, many obstacles remain towards attaining truly robust models. Beyond the robustness trade-off, efficient computational scaling of current defenses to multiple perturbations remains an open problem.

The code used for all of our experiments can be found here: `https://github.com/ftramer/MultiRobustness`

2 Theoretical Limits to Multi-perturbation Robustness

We study statistical properties of adversarial robustness in a natural statistical model introduced in [41], and which exhibits many phenomena observed on real data, such as trade-offs between robustness and accuracy [41] or a higher sample complexity for robust generalization [31]. This model also proves useful in analyzing and understanding adversarial robustness for multiple perturbations. Indeed, we prove a number of results that correspond to phenomena we observe on real data, in particular trade-offs in robustness to different ℓ_p or rotation-translation attacks [11].

We follow a line of works that study distributions for which adversarial examples exist *unconditionally* [41, 21, 33, 12, 14, 26]. These distributions, including ours, are much simpler than real-world data, and thus need not be evidence that adversarial examples are inevitable in practice. Rather, we hypothesize that current ML models are highly vulnerable to adversarial examples because they learn superficial data statistics [19, 13, 18] that share some properties of these simple distributions.

In prior work, a robustness trade-off for ℓ_∞ and ℓ_2-noise is shown in [21] for data distributed over two concentric spheres. Our conceptually simpler model has the advantage of yielding results beyond ℓ_p-norms (e.g., for spatial attacks) and which apply symmetrically to both classes. Building on work by Xu et al. [43], Demontis et al. [9] show a robustness trade-off for dual norms (e.g., ℓ_∞ and ℓ_1-noise) in linear classifiers.

2.1 Adversarial Risk for Multiple Perturbation Models

Consider a classification task for a distribution $\mathcal{D}$ over examples $\boldsymbol{x} \in \mathbb{R}^d$ and labels $y \in [C]$. Let $f : \mathbb{R}^d \to [C]$ denote a classifier and let $l(f(\boldsymbol{x}), y)$ be the zero-one loss (i.e., $\mathbb{1}_{f(\boldsymbol{x}) \neq y}$).

We assume n *perturbation types*, each characterized by a set S of allowed perturbations for an input $\boldsymbol{x}$. The set S can be an ℓ_p-ball [37, 15] or capture other perceptually small transforms such as image rotations and translations [11]. For a perturbation $\boldsymbol{r} \in S$, an adversarial example is

$\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{r}$ (this is pixel-wise addition for ℓ_p perturbations, but can be a more complex operation, e.g., for rotations).

For a perturbation set S and model f, we define $\mathcal{R}_{\text{adv}}(f; S) \coloneqq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\boldsymbol{r}\in S}\ l(f(\boldsymbol{x}+\boldsymbol{r}), y)\right]$ as the adversarial error rate. To extend $\mathcal{R}_{\text{adv}}$ to multiple perturbation sets $S_1, \ldots, S_n$, we can consider the *average* error rate for each S_i, denoted $\mathcal{R}^{\text{avg}}_{\text{adv}}$. This metric most clearly captures the trade-off in robustness across independent perturbation types, but is not the most appropriate from a security perspective on adversarial examples. A more natural metric, denoted $\mathcal{R}^{\text{max}}_{\text{adv}}$, is the error rate against an adversary that picks, for each input, the worst perturbation from the *union* of the S_i. More formally,

$$\mathcal{R}^{\text{max}}_{\text{adv}}(f; S_1, \ldots, S_n) \coloneqq \mathcal{R}_{\text{adv}}(f; \cup_i S_i)\ , \quad \mathcal{R}^{\text{avg}}_{\text{adv}}(f; S_1, \ldots, S_n) \coloneqq \tfrac{1}{n}\textstyle\sum_i \mathcal{R}_{\text{adv}}(f; S_i)\ . \tag{1}$$

Most results in this section are *lower bounds* on $\mathcal{R}^{\text{avg}}_{\text{adv}}$, which also hold for $\mathcal{R}^{\text{avg}}_{\text{max}}$ since $\mathcal{R}^{\text{max}}_{\text{adv}} \geq \mathcal{R}^{\text{avg}}_{\text{adv}}$.

Two perturbation types S_1, S_2 are *Mutually Exclusive Perturbations (MEPs)*, if $\mathcal{R}^{\text{avg}}_{\text{adv}}(f; S_1, S_2) \geq 1/|C|$ for all models f (i.e., no model has non-trivial average risk against both perturbations).

2.2 A binary classification task

We analyze the adversarial robustness trade-off for different perturbation types in a natural statistical model introduced by Tsipras et al. [41]. Their binary classification task consists of input-label pairs $(\boldsymbol{x}, y)$ sampled from a distribution $\mathcal{D}$ as follows (note that $\mathcal{D}$ is $(d+1)$-dimensional):

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad x_0 = \begin{cases} +y, \text{ w.p. } p_0, \\ -y, \text{ w.p. } 1 - p_0 \end{cases}, \quad x_1, \ldots, x_d \overset{i.i.d}{\sim} \mathcal{N}(y\eta, 1)\ , \tag{2}$$

where $p_0 \geq 0.5$, $\mathcal{N}(\mu, \sigma^2)$ is the normal distribution and $\eta = \alpha/\sqrt{d}$ for some positive constant α.

For this distribution, Tsipras et al. [41] show a trade-off between standard and adversarial accuracy (for ℓ_∞ attacks), by drawing a distinction between the "robust" feature x_0 that small ℓ_∞-noise cannot manipulate, and the "non-robust" features $x_1, \ldots, x_d$ that can be fully overridden by small ℓ_∞-noise.

2.3 Small ℓ_∞ and ℓ_1 Perturbations are Mutually Exclusive

The starting point of our analysis is the observation that the robustness of a feature depends on the considered perturbation type. To illustrate, we recall two classifiers from [41] that operate on disjoint feature sets. The first, $f(\boldsymbol{x}) = \text{sign}(x_0)$, achieves accuracy p_0 for all ℓ_∞-perturbations with $\epsilon < 1$ but is highly vulnerable to ℓ_1-perturbations of size $\epsilon \geq 1$. The second classifier, $h(\boldsymbol{x}) = \text{sign}(\sum_{i=1}^{d} x_i)$ is robust to ℓ_1-perturbations of average norm below $\mathbb{E}[\sum_{i=1}^{d} x_i] = \Theta(\sqrt{d})$, yet it is fully subverted by a ℓ_∞-perturbation that shifts the features $x_1, \ldots, x_d$ by $\pm 2\eta = \Theta(1/\sqrt{d})$. We prove that this tension between ℓ_∞ and ℓ_1 robustness, and of the choice of "robust" features, is inherent for this task:

Theorem 1. *Let f be a classifier for $\mathcal{D}$. Let S_∞ be the set of ℓ_∞-bounded perturbations with $\epsilon = 2\eta$, and S_1 the set of ℓ_1-bounded perturbations with $\epsilon = 2$. Then, $\mathcal{R}^{avg}_{adv}(f; S_\infty, S_1) \geq 1/2$.*

The proof is in Appendix F. The bound shows that no classifier can attain better $\mathcal{R}^{\text{avg}}_{\text{adv}}$ (and thus $\mathcal{R}^{\text{max}}_{\text{adv}}$) than a trivial constant classifier $f(x) = 1$, which satisfies $\mathcal{R}_{\text{adv}}(f; S_\infty) = \mathcal{R}_{\text{adv}}(f; S_1) = 1/2$.

Similar to [9], our analysis extends to arbitrary dual norms ℓ_p and ℓ_q with $1/p + 1/q = 1$ and $p < 2$. The perturbation required to flip the features $x_1, \ldots, x_n$ has an ℓ_p norm of $\Theta(d^{\frac{1}{p}-\frac{1}{2}}) = \omega(1)$ and an ℓ_q norm of $\Theta(d^{\frac{1}{q}-\frac{1}{2}}) = \Theta(d^{\frac{1}{2}-\frac{1}{p}}) = o(1)$. Thus, feature x_0 is more robust than features $x_1, \ldots, x_n$ with respect to the ℓ_q-norm, whereas for the dual ℓ_p-norm the situation is reversed.

2.4 Small ℓ_∞ and Spatial Perturbations are (nearly) Mutually Exclusive

We now analyze two other orthogonal perturbation types, ℓ_∞-noise and rotation-translations [11]. In some cases, increasing robustness to ℓ_∞-noise has been shown to decrease robustness to rotation-translations [11]. We prove that such a trade-off is inherent for our binary classification task.

To reason about rotation-translations, we assume that the features x_i form a 2D grid. We also let x_0 be distributed as $\mathcal{N}(y, \alpha^{-2})$, a technicality that does not qualitatively change our prior results. Note that the distribution of the features $x_1, \ldots, x_d$ is permutation-invariant. Thus, the only power of a rotation-translation adversary is to "move" feature x_0. Without loss of generality, we identify a small rotation-translation of an input $\boldsymbol{x}$ with a permutation of its features that sends x_0 to one of N fixed positions (e.g., with translations of $\pm$3px as in [11], x_0 can be moved to $N = 49$ different positions).

A model can be robust to these permutations by ignoring the N positions that feature x_0 can be moved to, and focusing on the remaining permutation-invariant features. Yet, this model is vulnerable to ℓ_∞-noise, as it ignores x_0. In turn, a model that relies on feature x_0 can be robust to ℓ_∞-perturbations, but is vulnerable to a spatial perturbation that "hides" x_0 among other features. Formally, we show:

Theorem 2. *Let f be a classifier for $\mathcal{D}$ (with $x_0 \sim \mathcal{N}(y, \alpha^{-2})$). Let S_∞ be the set of ℓ_∞-bounded perturbations with $\epsilon = 2\eta$, and S_{RT} be the set of perturbations for an RT adversary with budget N. Then, $\mathcal{R}^{avg}_{adv}(f; S_\infty, S_{RT}) \geq 1/2 - O(1/\sqrt{N})$.*

The proof, given in Appendix G, is non-trivial and yields an asymptotic lower-bound on $\mathcal{R}^{\text{avg}}_{\text{adv}}$. We can also provide tight numerical estimates for concrete parameter settings (see Appendix G.1).

2.5 Affine Combinations of Perturbations

We defined $\mathcal{R}^{\max}_{\text{adv}}$ as the error rate against an adversary that may choose a different perturbation type for each input. If a model were robust to this adversary, what can we say about the robustness to a more creative adversary that *combines* different perturbation types? To answer this question, we introduce a new adversary that mixes different attacks by linearly interpolating between perturbations.

For a perturbation set S and $\beta \in [0, 1]$, we denote $\beta \cdot S$ the set of perturbations scaled down by β. For an ℓ_p-ball with radius ϵ, this is the ball with radius $\beta \cdot \epsilon$. For rotation-translations, the attack budget N is scaled to $\beta \cdot N$. For two sets S_1, S_2, we define $S_{\text{affine}}(S_1, S_2)$ as the set of perturbations that compound a perturbation $\boldsymbol{r}_1 \in \beta \cdot S_1$ with a perturbation $\boldsymbol{r}_2 \in (1 - \beta) \cdot S_2$, for any $\beta \in [0, 1]$.

Consider one adversary that chooses, for each input, ℓ_p or ℓ_q-noise from balls S_p and S_q, for $p, q > 0$. The affine adversary picks perturbations from the set S_{affine} defined as above. We show:

Claim 3. *For a linear classifier $f(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x}+b)$, we have $\mathcal{R}^{max}_{adv}(f; S_p, S_q) = \mathcal{R}_{adv}(f; S_{affine})$.*

Thus, for linear classifiers, robustness to a union of ℓ_p-perturbations implies robustness to affine adversaries (this holds for any distribution). The proof, in Appendix H extends to models that are *locally linear* within balls S_p and S_q around the data points. For the distribution $\mathcal{D}$ of Section 2.2, we can further show that there are settings (distinct from the one in Theorem 1) where: (1) robustness against a union of ℓ_∞ and ℓ_1-perturbations is possible; (2) this requires the model to be non-linear; (3) yet, robustness to affine adversaries is impossible (see Appendix I for details). Our experiments in Section 4 show that neural networks trained on CIFAR10 have a behavior that is consistent with locally-linear models, in that they are as robust to affine adversaries as against a union of ℓ_p-attacks.

In contrast, compounding ℓ_∞ and spatial perturbations yields a stronger attack, even for linear models:

Theorem 4. *Let $f(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x} + b)$ be a linear classifier for $\mathcal{D}$ (with $x_0 \sim \mathcal{N}(y, \alpha^{-2})$). Let S_∞ be some ℓ_∞-ball and S_{RT} be rotation-translations with budget $N > 2$. Define S_{affine} as above. Assume $w_0 > w_i > 0, \forall i \in [1, d]$. Then $\mathcal{R}_{adv}(f; S_{affine}) > \mathcal{R}^{max}_{adv}(f; S_\infty, S_{RT})$.*

Input: Input $\boldsymbol{x} \in [0,1]^d$, steps k, step-size γ, percentile q, ℓ_1-bound ϵ
Output: $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{r}$ s.t. $\|\boldsymbol{r}\|_1 \leq \epsilon$

$\boldsymbol{r} \leftarrow \mathbf{0}^d$
for $1 \leq i \leq k$ **do**
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{r}} L(\theta, \boldsymbol{x} + \boldsymbol{r}, y)$
  $e_i = \text{sign}(g_i)$ if $|g_i| \geq P_q(|\boldsymbol{g}|)$, else 0
  $\boldsymbol{r} \leftarrow \boldsymbol{r} + \gamma \cdot \boldsymbol{e} / \|\boldsymbol{e}\|_1$
  $\boldsymbol{r} \leftarrow \mathit{\Pi}_{S_1^\epsilon}(\boldsymbol{r})$
end

Algorithm 1: The Sparse ℓ_1 Descent Attack (SLIDE). $P_q(|\boldsymbol{g}|)$ denotes the q^{th} percentile of $|\boldsymbol{g}|$ and $\mathit{\Pi}_{S_1^\epsilon}$ is the projection onto the ℓ_1-ball (see [10]).

This result (the proof is in Appendix J) draws a distinction between the strength of affine combinations of ℓ_p-noise, and combinations of ℓ_∞ and spatial perturbations. It also shows that robustness to a union of perturbations can be insufficient against a more creative affine adversary. These results are consistent with behavior we observe in models trained on real data (see Section 4).

3 New Attacks and Adversarial Training Schemes

We complement our theoretical results with empirical evaluations of the robustness trade-off on MNIST and CIFAR10. To this end, we first introduce new adversarial training schemes tailored to the multi-perturbation risks defined in Equation (1), as well as a novel attack for the ℓ_1-norm.

Multi-perturbation adversarial training. Let

$$\hat{\mathcal{R}}_{\text{adv}}(f; S) = \sum_{i=1}^{m} \max_{\boldsymbol{r} \in S} L(f(\boldsymbol{x}^{(i)} + \boldsymbol{r}), y^{(i)}) \ ,$$

bet the empirical adversarial risk, where L is the training loss and D is the training set. For a single perturbation type, $\hat{\mathcal{R}}_{\text{adv}}$ can be minimized with *adversarial training* [25]: the maximal loss is approximated by an attack procedure $\mathcal{A}(\boldsymbol{x})$, such that $\max_{\boldsymbol{r} \in S} L(f(\boldsymbol{x} + \boldsymbol{r}), y) \approx L(f(\mathcal{A}(\boldsymbol{x})), y)$.

For $i \in [1, d]$, let $\mathcal{A}_i$ be an attack for the perturbation set S_i. The two multi-attack robustness metrics introduced in Equation (1) immediately yield the following natural adversarial training strategies:

1. **"Max" strategy:** For each input $\boldsymbol{x}$, we train on the strongest adversarial example from all attacks, i.e., the max in $\hat{\mathcal{R}}_{\text{adv}}$ is replaced by $L(f(\mathcal{A}_{k^*}(\boldsymbol{x})), y)$, for $k^* = \arg\max_k L(f(\mathcal{A}_k(\boldsymbol{x})), y)$.
2. **"Avg" strategy:** This strategy simultaneously trains on adversarial examples from all attacks. That is, the max in $\hat{\mathcal{R}}_{\text{adv}}$ is replaced by $\frac{1}{n} \sum_{i=1}^{n} L(f(\mathcal{A}_i(\boldsymbol{x}), y))$.

The sparse ℓ_1-descent attack (SLIDE). Adversarial training is contingent on a *strong* and *efficient* attack. Training on weak attacks gives no robustness [40], while strong optimization attacks (e.g., [6, 8]) are prohibitively expensive. Projected Gradient Descent (PGD) [22, 25] is a popular choice of attack that is both efficient and produces strong perturbations. To complement our formal results, we want to train models on ℓ_1-perturbations. Yet, we show that the ℓ_1-version of PGD is highly inefficient, and propose a better approach suitable for adversarial training.

PGD is a *steepest descent* algorithm [24]. In each iteration, the perturbation is updated in the steepest descent direction $\arg\max_{\|\boldsymbol{v}\| \leq 1} \boldsymbol{v}^T \boldsymbol{g}$, where $\boldsymbol{g}$ is the gradient of the loss. For the ℓ_∞-norm, the steepest descent direction is $\texttt{sign}(\boldsymbol{g})$ [15], and for ℓ_2, it is $\boldsymbol{g} / \|\boldsymbol{g}\|_2$. For the ℓ_1-norm, the steepest descent direction is the unit vector $\boldsymbol{e}$ with $e_{i^*} = \text{sign}(g_{i^*})$, for $i^* = \arg\max_i |g_i|$.

This yields an inefficient attack, as each iteration updates a single index of the perturbation $\boldsymbol{r}$. We thus design a new attack with finer control over the sparsity of an update step. For $q \in [0, 1]$, let $P_q(|\boldsymbol{g}|)$ be the q^{th} *percentile* of $|\boldsymbol{g}|$. We set $e_i = \text{sign}(g_i)$ if $|g_i| \geq P_q(|\boldsymbol{g}|)$ and 0 otherwise, and normalize $\boldsymbol{e}$ to unit ℓ_1-norm. For $q \gg 1/d$, we thus update many indices of $\boldsymbol{r}$ at once. We introduce another optimization to handle clipping, by ignoring gradient components where the update step cannot make progress (i.e., where $x_i + r_i \in \{0, 1\}$ and g_i points outside the domain). To project $\boldsymbol{r}$ onto an ℓ_1-ball, we use an algorithm of Duchi et al. [10]. Algorithm 1 describes our attack. It outperforms the steepest descent attack as well as a recently proposed Frank-Wolfe algorithm for ℓ_1-attacks [20] (see Appendix B). Our attack is competitive with the more expensive EAD attack [8] (see Appendix C).

4 Experiments

We use our new adversarial training schemes to measure the robustness trade-off on MNIST and CIFAR10.[1] MNIST is an interesting case-study as *distinct* models achieve strong robustness to different ℓ_p and spatial attacks[31, 11]. Despite the dataset's simplicity, we show that no single model achieves strong ℓ_∞, ℓ_1 and ℓ_2 robustness, and that new techniques are required to close this gap. The code used for all of our experiments can be found here: `https://github.com/ftramer/MultiRobustness`

Training and evaluation setup. We first use adversarial training to train models on a single perturbation type. For MNIST, we use $\ell_1(\epsilon = 10)$, $\ell_2(\epsilon = 2)$ and $\ell_\infty(\epsilon = 0.3)$. For CIFAR10 we use $\ell_\infty(\epsilon = \frac{4}{255})$ and $\ell_1(\epsilon = \frac{2000}{255})$. We also train on rotation-translation attacks with $\pm$3px translations and $\pm 30°$ rotations as in [11]. We denote these models Adv_1, Adv_2, Adv_∞, and Adv_{RT}. We then use the "max" and "avg" strategies from Section 3 to train models Adv_{max} and Adv_{avg} against multiple perturbations. We train once on all ℓ_p-perturbations, and once on both ℓ_∞ and RT perturbations. We use the same CNN (for MNIST) and wide ResNet model (for CIFAR10) as Madry et al. [25]. Appendix A has more details on the training setup, and attack and training hyper-parameters.

We evaluate robustness of all models using multiple attacks: (1) we use *gradient-based attacks* for all ℓ_p-norms, i.e., PGD [25] and our SLIDE attack with 100 steps and 40 restarts (20 restarts on CIFAR10), as well as Carlini and Wagner's ℓ_2-attack [6] (C&W), and an ℓ_1-variant—EAD [8]; (2) to detect gradient-masking, we use *decision-based attacks*: the Boundary Attack [3] for ℓ_2, the Pointwise Attack [31] for ℓ_1, and the Boundary Attack++ [7] for ℓ_∞; (3) for spatial attacks, we use the optimal attack of [11] that enumerates all small rotations and translations. For unbounded attacks (C&W, EAD and decision-based attacks), we discard perturbations outside the ℓ_p-ball.

For each model, we report accuracy on 1000 test points for: (1) individual perturbation types; (2) the union of these types, i.e., $1 - \mathcal{R}_{\text{adv}}^{\text{max}}$; and (3) the average of all perturbation types, $1 - \mathcal{R}_{\text{adv}}^{\text{avg}}$. We briefly discuss the optimal error that can be achieved if there is no robustness trade-off. For perturbation sets $S_1, \dots S_n$, let $\mathcal{R}_1, \dots, \mathcal{R}_n$ be the optimal risks achieved by distinct models. Then, a single model can at best achieve risk $\mathcal{R}_i$ for each S_i, i.e., $\text{OPT}(\mathcal{R}_{\text{adv}}^{\text{avg}}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}_i$. If the errors are fully correlated, so that a maximal number of inputs admit *no* attack, we have $\text{OPT}(\mathcal{R}_{\text{adv}}^{\text{max}}) = \max\{\mathcal{R}_1, \dots, \mathcal{R}_n\}$. Our experiments show that these optimal error rates are not achieved.

Results on MNIST. Results are in Table 1. The left table is for the union of ℓ_p-attacks, and the right table is for the union of ℓ_∞ and RT attacks. In both cases, the multi-perturbation training strategies "succeed", in that models Adv_{avg} and Adv_{max} achieve higher multi-perturbation accuracy than any of the models trained against a single perturbation type.

The results for ℓ_∞ and RT attacks are promising, although the best model Adv_{max} only achieves $1 - \mathcal{R}_{\text{adv}}^{\text{max}} = 83.8\%$ and $1 - \mathcal{R}_{\text{adv}}^{\text{avg}} = 87.6\%$, which is far less than the optimal values,

[1] Kang et al. [20] recently studied the transfer between ℓ_∞, ℓ_1 and ℓ_2-attacks for adversarially trained models on ImageNet. They show that models trained on one type of perturbation are not robust to others, but they do not attempt to train models against multiple attacks simultaneously.

Table 1: **Evaluation of MNIST models trained on ℓ_∞, ℓ_1 and ℓ_2 attacks (left) or ℓ_∞ and rotation-translation (RT) attacks (right).** Models Adv_∞, Adv_1, Adv_2 and Adv_{RT} are trained on a single attack, while Adv_{avg} and Adv_{max} are trained on multiple attacks using the "avg" and "max" strategies. The columns show a model's accuracy on individual perturbation types, on the union of them ($1 - \mathcal{R}_{\text{adv}}^{\text{max}}$), and the average accuracy across them ($1 - \mathcal{R}_{\text{adv}}^{\text{avg}}$). The best results are in bold (at 95% confidence). Results in red indicate gradient-masking, see Appendix C for a breakdown of all attacks.

Model	Acc.	ℓ_∞	ℓ_1	ℓ_2	$1 - \mathcal{R}_{\text{adv}}^{\text{max}}$	$1 - \mathcal{R}_{\text{adv}}^{\text{avg}}$
Nat	**99.4**	0.0	12.4	8.5	0.0	7.0
Adv_∞	**99.1**	**91.1**	12.1	11.3	6.8	38.2
Adv_1	98.9	0.0	**78.5**	50.6	0.0	43.0
Adv_2	98.5	0.4	68.0	**71.8**	0.4	46.7
Adv_{avg}	97.3	76.7	53.9	58.3	**49.9**	**63.0**
Adv_{max}	97.2	71.7	62.6	56.0	**52.4**	**63.4**

Model	Acc.	ℓ_∞	RT	$1 - \mathcal{R}_{\text{adv}}^{\text{max}}$	$1 - \mathcal{R}_{\text{adv}}^{\text{avg}}$
Nat	**99.4**	0.0	0.0	0.0	0.0
Adv_∞	**99.1**	**91.4**	0.2	0.2	45.8
Adv_{RT}	**99.3**	0.0	**94.6**	0.0	47.3
Adv_{avg}	**99.2**	88.2	86.4	**82.9**	**87.3**
Adv_{max}	98.9	89.6	85.6	**83.8**	**87.6**

$1 - \text{OPT}(\mathcal{R}_{\text{adv}}^{\text{max}}) = \min\{91.4\%, 94.6\%\} = 91.4\%$ and $1 - \text{OPT}(\mathcal{R}_{\text{adv}}^{\text{avg}}) = (91.4\% + 94.6\%)/2 = 93\%$. Thus, these models do exhibit some form of the robustness trade-off analyzed in Section 2.

The ℓ_p results are surprisingly mediocre and re-raise questions about whether MNIST can be considered "solved" from a robustness perspective. Indeed, while training *separate* models to resist ℓ_1, ℓ_2 or ℓ_∞ attacks works well, resisting all attacks simultaneously fails. This agrees with the results of Schott et al. [31], whose models achieve either high ℓ_∞ or ℓ_2 robustness, but not both simultaneously. We show that in our case, this lack of robustness is partly due to gradient masking.

First-order adversarial training and gradient masking on MNIST. The model Adv_∞ is not robust to ℓ_1 and ℓ_2-attacks. This is unsurprising as the model was only trained on ℓ_∞-attacks. Yet, comparing the model's accuracy against multiple types of ℓ_1 and ℓ_2 attacks (see Appendix C) reveals a more curious phenomenon: Adv_∞ has high accuracy against *first-order* ℓ_1 and ℓ_2-attacks such as PGD, but is broken by decision-free attacks. This is an indication of gradient-masking [27, 40, 1].

This issue had been observed before [31, 23], but an explanation remained illusive, especially since ℓ_∞-PGD does not appear to suffer from gradient masking (see [25]). We explain this phenomenon by inspecting the learned features of model Adv_∞, as in [25]. We find that the model's first layer learns threshold filters $\boldsymbol{z} = \text{ReLU}(\alpha \cdot (\boldsymbol{x} - \epsilon))$ for $\alpha > 0$. As most pixels in MNIST are zero, most of the z_i cannot be activated by an ϵ-bounded ℓ_∞-attack. The ℓ_∞-PGD thus optimizes a smooth (albeit flat) loss function. In contrast, ℓ_1- and ℓ_2-attacks can move a pixel $x_i = 0$ to $\hat{x}_i > \epsilon$ thus activating z_i, but have no gradients to rely on (i.e, $\text{d}z_i/\text{d}x_i = 0$ for any $x_i \leq \epsilon$). Figure 3 in Appendix D shows that the model's loss resembles a step-function, for which first-order attacks such as PGD are inadequate.

Note that training against first-order ℓ_1 or ℓ_2-attacks directly (i.e., models Adv_1 and Adv_2 in Table 1), seems to yield genuine robustness to these perturbations. This is surprising in that, because of gradient masking, model Adv_∞ actually achieves lower training loss against first-order ℓ_1 and ℓ_2-attacks than models Adv_1 and Adv_2. That is, Adv_1 and Adv_2 converged to sub-optimal local minima of their respective training objectives, yet these minima generalize much better to stronger attacks.

The models Adv_{avg} and Adv_{max} that are trained against ℓ_∞, ℓ_1 and ℓ_2-attacks also learn to use thresholding to resist ℓ_∞-attacks while spuriously masking gradient for ℓ_1 and ℓ_2-attacks. This is evidence that, unlike previously thought [41], training against a strong first-order attack (such as PGD) can cause the model to minimize its training loss via gradient masking. To circumvent this issue, alternatives to first-order adversarial training seem necessary. Potential (costly) approaches include training on gradient-free attacks, or extending certified defenses [28, 42] to multiple perturbations. Certified defenses provide provable bounds that are much weaker than the robustness

Table 2: **Evaluation of CIFAR10 models trained against ℓ_∞ and ℓ_1 attacks (left) or ℓ_∞ and rotation-translation (RT) attacks (right).** Models Adv_∞, Adv_1 and Adv_{RT} are trained against a single attack, while Adv_{avg} and Adv_{max} are trained against two attacks using the "avg" and "max" strategies. The columns show a model's accuracy on individual perturbation types, on the union of them ($1-\mathcal{R}_{\text{adv}}^{\text{max}}$), and the average accuracy across them ($1-\mathcal{R}_{\text{adv}}^{\text{avg}}$). The best results are in bold (at 95% confidence). A breakdown of all ℓ_1 attacks is in Appendix C.

Model	Acc.	ℓ_∞	ℓ_1	$1-\mathcal{R}_{\text{adv}}^{\text{max}}$	$1-\mathcal{R}_{\text{adv}}^{\text{avg}}$
Nat	**95.7**	0.0	0.0	0.0	0.0
Adv_∞	92.0	**71.0**	16.4	16.4	44.9
Adv_1	90.8	53.4	**66.2**	53.1	60.0
Adv_{avg}	91.1	64.1	60.8	**59.4**	**62.5**
Adv_{max}	91.2	65.7	62.5	**61.1**	**64.1**

Model	Acc.	ℓ_∞	RT	$1-\mathcal{R}_{\text{adv}}^{\text{max}}$	$1-\mathcal{R}_{\text{adv}}^{\text{avg}}$
Nat	**95.7**	0.0	5.9	0.0	3.0
Adv_∞	92.0	**71.0**	8.9	8.7	40.0
Adv_{RT}	**94.9**	0.0	**82.5**	0.0	41.3
Adv_{avg}	93.6	67.8	78.2	**65.2**	**73.0**
Adv_{max}	93.1	**69.6**	75.2	**65.7**	**72.4**

Table 3: **Evaluation of affine attacks.** For models trained with the "max" strategy, we evaluate against attacks from a union S_U of perturbation sets, and against an affine adversary that interpolates between perturbations. Examples of affine attacks are in Figure 4.

Dataset	Attacks	acc. on S_{U}	acc. on S_{affine}
MNIST	ℓ_∞ & RT	83.8	62.6
CIFAR10	ℓ_∞ & RT	65.7	56.0
CIFAR10	ℓ_∞ & ℓ_1	61.1	58.0

attained by adversarial training, and certifying multiple perturbation types is likely to exacerbate this gap.

Results on CIFAR10. The left table in Table 2 considers the union of ℓ_∞ and ℓ_1 perturbations, while the right table considers the union of ℓ_∞ and RT perturbations. As on MNIST, the models Adv_{avg} and Adv_{max} achieve better multi-perturbation robustness than any of the models trained on a single perturbation, but fail to match the optimal error rates we could hope for. For ℓ_1 and ℓ_∞-attacks, we achieve $1-\mathcal{R}_{\text{adv}}^{\text{max}} = 61.1\%$ and $1-\mathcal{R}_{\text{adv}}^{\text{avg}} = 64.1\%$, again significantly below the optimal values, $1-\text{OPT}(\mathcal{R}_{\text{adv}}^{\text{max}}) = \min\{71.0\%, 66.2\%\} = 66.2\%$ and $1-\text{OPT}(\mathcal{R}_{\text{adv}}^{\text{avg}}) = (71.0\%+66.2\%)/2 = 68.6\%$. The results for ℓ_∞ and RT attacks are qualitatively and quantitatively similar. [2]

Interestingly, models Adv_{avg} and Adv_{max} achieve 100% *training accuracy*. Thus, multi-perturbation robustness increases the *adversarial generalization gap* [30]. These models might be resorting to more memorization because they fail to find features robust to both attacks.

Affine Adversaries. Finally, we evaluate the affine attacks introduced in Section 2.5. These attacks take affine combinations of two perturbation types, and we apply them on the models Adv_{max} (we omit the ℓ_p-case on MNIST due to gradient masking). To compound ℓ_∞ and ℓ_1-noise, we devise an attack that updates both perturbations in alternation. To compound ℓ_∞ and RT attacks, we pick random rotation-translations (with $\pm 3\beta$px translations and $\pm 30\beta^\circ$ rotations), apply an ℓ_∞-attack with budget $(1-\beta)\epsilon$ to each, and retain the worst example.

The results in Table 3 match the predictions of our formal analysis: (1) affine combinations of ℓ_p perturbations are no stronger than their union. This is expected given Claim 3 and prior observations that neural networks are close to linear near the data [15, 29]; (2) combining of ℓ_∞ and RT attacks does yield a stronger attack, as shown in Theorem 4. This demonstrates that robustness to a union of perturbations can still be insufficient to protect against more complex combinations of perturbations.

[2] An interesting open question is why the model Adv_{avg} trained on ℓ_∞ and RT attacks does not attain optimal average robustness $\mathcal{R}_{\text{adv}}^{\text{avg}}$. Indeed, on CIFAR10, detecting the RT attack of [11] is easy, due to the black in-painted pixels in a transformed image. The following "ensemble" model thus achieves optimal $\mathcal{R}_{\text{adv}}^{\text{avg}}$ (but not necessarily optimal $\mathcal{R}_{\text{adv}}^{\text{max}}$): on input $\hat{\boldsymbol{x}}$, return $\text{Adv}_{\text{RT}}(\hat{\boldsymbol{x}})$ if there are black in-painted pixels, otherwise return $\text{Adv}_\infty(\hat{\boldsymbol{x}})$. The fact that model Adv_{avg} did not learn such a function might hint at some limitation of adversarial training.

5 Discussion and Open Problems

Despite recent success in defending ML models against some perturbation types [25, 11, 31], extending these defenses to multiple perturbations unveils a clear robustness trade-off. This tension may be rooted in its unconditional occurrence in natural and simple distributions, as we proved in Section 2.

Our new adversarial training strategies fail to achieve competitive robustness to more than one attack type, but narrow the gap towards multi-perturbation robustness. We note that the optimal risks $\mathcal{R}_{\text{adv}}^{\max}$ and $\mathcal{R}_{\text{adv}}^{\text{avg}}$ that we achieve are very close. Thus, for most data points, the models are either robust to all perturbation types or none of them. This hints that some points (sometimes referred to as *prototypical examples* [4, 36]) are inherently easier to classify robustly, regardless of the perturbation type.

We showed that first-order adversarial training for multiple ℓ_p-attacks suffers from gradient masking on MNIST. Achieving better robustness on this simple dataset is an open problem. Another challenge is reducing the cost of our adversarial training strategies, which scale linearly in the number of perturbation types. Breaking this linear dependency requires efficient techniques for finding perturbations in a union of sets, which might be hard for sets with near-empty intersection (e.g., ℓ_∞ and ℓ_1-balls). The cost of adversarial training has also be reduced by merging the inner loop of a PGD attack and gradient updates of the model parameters [34, 44], but it is unclear how to extend this approach to a union of perturbations (some of which are not optimized using PGD, e.g., rotation-translations).

Hendrycks and Dietterich [17], and Geirhos et al. [13] recently measured robustness of classifiers to multiple common (i.e., non-adversarial) image corruptions (e.g., random image blurring). In that setting, they also find that different classifiers achieve better robustness to some corruptions, and that no single classifier achieves the highest accuracy under all forms. The interplay between multi-perturbation robustness in the adversarial and common corruption case is worth further exploration.

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[2] A. C. Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

[3] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[4] N. Carlini, U. Erlingsson, and N. Papernot. Prototypical examples in deep learning: Metrics, characteristics, and utility. 2018.

[5] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530, 2016.

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[7] J. Chen and M. I. Jordan. Boundary attack++: Query-efficient decision-based adversarial attack. *arXiv preprint arXiv:1904.02144*, 2019.

[8] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI Conference on Artificial Intelligence*, 2018.

[9] A. Demontis, P. Russu, B. Biggio, G. Fumera, and F. Roli. On security and sparsity of linear classifiers for adversarial settings. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 322–332. Springer, 2016.

[10] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*, 2008.

[11] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.

[12] A. Fawzi, H. Fawzi, and O. Fawzi. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems*, pages 1186–1195, 2018.

[13] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2019.

[14] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[16] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, 2017.

[17] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[18] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[19] J. Jo and Y. Bengio. Measuring the tendency of CNNs to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

[20] D. Kang, Y. Sun, T. Brown, D. Hendrycks, and J. Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.

[21] M. Khoury and D. Hadfield-Menell. On the geometry of adversarial examples, 2019.

[22] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

[23] B. Li, C. Chen, W. Wang, and L. Carin. Second-order adversarial attack and certifiable robustness. *arXiv preprint arXiv:1809.03113*, 2018.

[24] A. Madry and Z. Kolter. Adversarial robustness: Theory and practice. In *Tutorial at NeurIPS 2018*, 2018.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[26] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. *arXiv preprint arXiv:1809.03063*, 2018.

[27] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *ASIACCS*, pages 506–519. ACM, 2017.

[28] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*. ACM, 2016.

[30] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5019–5031, 2018.

[31] L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on mnist. In *International Conference on Learning Representations (ICLR)*, 2019.

[32] L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on mnist (OpenReview comment on spatial transformations), 2019.

[33] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations (ICLR)*, 2019.

[34] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

[35] Y. Sharma and P.-Y. Chen. Attacking the madry defense model with l1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

[36] P. Stock and M. Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018.

[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[38] F. Tramèr and D. Boneh. Adversarial training and robustness for multiple perturbations. In *Neural Information Processing Systems (NeurIPS) 2019*, 2019. arXiv preprint arXiv:1904.13000.

[39] F. Tramèr, P. Dupré, G. Rusak, G. Pellegrino, and D. Boneh. Ad-versarial: Perceptual ad-blocking meets adversarial machine learning. arXiv preprint arXiv:1811:03194, Nov 2018.

[40] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.

[41] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[42] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

[43] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009.

[44] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Painless adversarial training using maximal principle. *arXiv preprint arXiv:1905.00877*, 2019.

A Experimental Setup

MNIST. We use the CNN model from Madry et al. [25] and train for 10 epochs with Adam and a learning rate of 10^{-3} reduced to 10^{-4} after 5 epochs (batch size of 100). To accelerate convergence, we train against a weaker adversary in the first epoch (with 1/3 of the perturbation budget). For training, we use PGD with 40 iterations for ℓ_∞ and 100 iterations for ℓ_1 and ℓ_2. For rotation-translations, we use the attack from [11] that picks the worst of 10 random rotation-translations.

CIFAR10. We use the same wide ResNet model as [25]. We train for 80k steps of gradient descent with batch size 128 (205 epochs). When using the "avg" strategy for wide ResNet models, we had to halve the batch size to avoid overflowing the GPU's memory. We accordingly doubled the number of training steps and learning rate schedule. We use a learning rate of 0.1 decayed by a factor 10 after 40k and 60k steps, a momentum of 0.9, and weight decay of 0.0002. Except for the RT attack, we use standard data augmentation with random padding, cropping and horizontal flipping (see [11] for details). We extract 1,000 points from the CIFAR10 test as a validation set for early-stopping.

For training, we use PGD with 10 iterations for ℓ_∞, and 20 iterations for ℓ_1. [3] For rotation-translations, we also use the attack from [11] that trains on the worst of 10 randomly chosen rotation-translations.

B Performance of the Sparse ℓ_1-Descent Attack

In Figure 2, we compare the performance of our new Sparse ℓ_1-Descent Attack (SLIDE) for different choices of gradient sparsity. We also compare to the standard PGD attack with the steepest-descent update rule, as well as a recent attack proposed in [20] that adapts the Frank-Wolfe optimization

[3] Our new attack ℓ_1-attack, described in Section 3, has a parameter q to controls the sparsity of the gradient updates. When leaving this parameter constant during training, the model overfits and fails to achieve general robustness. To resolve this issue, we sample $q \in [80\%, 99.5\%]$ at random for each attack during training. We also found that 10 iterations were insufficient to get a strong attack and thus increased the iteration count to 20.

algorithm for finding ℓ_1-bounded adversarial examples. As we explained in Section 3, we expect our attack to outperform PGD as the steepest-descent vector is too sparse in the ℓ_1-case, and we indeed observe a significant improvement by choosing denser updates.

The subpar performance of the Frank-Wolfe algorithm is also intriguing. We believe it is due to the attack's linearly decreasing step-size (the k^{th} iteration has a step-size of $O(1/k)$, see [20] for details). While this choice is appropriate for optimizing convex functions, in the non-convex case it overly emphasizes the first steps of the attack, which intuitively should increase the likelihood of landing in a local minima.

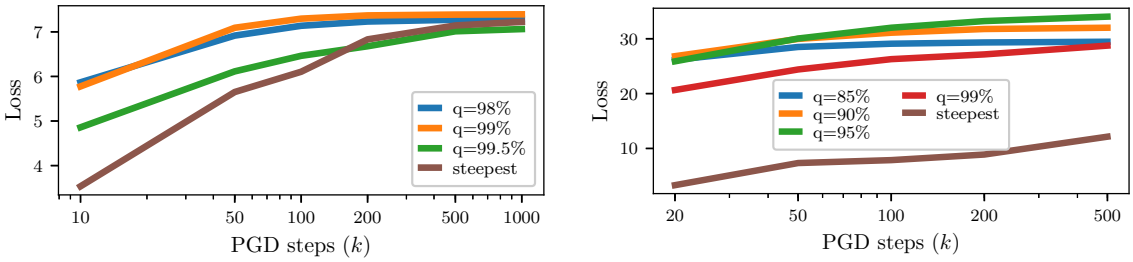


Figure 2: **Performance of the Sparse ℓ_1-Descent Attack on MNIST (left) and CIFAR10 (right) for different choices of descent directions.** We run the attack for up to 1,000 steps and plot the evolution of the cross-entropy loss, for an undefended model. We vary the sparsity of the gradient updates (controlled by the parameter q), and compare to the standard PGD attack that uses the steepest descent vector, as well as the Frank-Wolfe ℓ_1-attack from [20]. For appropriate q, our attack vastly outperforms PGD and Frank-Wolfe.

C Breakdown of ℓ_p-Attacks on Adversarially Trained Models

Tables 4 and 5 below give a more detailed breakdown of each model's accuracy against each ℓ_p attack we considered. For each model and attack, we evaluate the attack on 1,000 test points and report the accuracy. For each individual perturbation type (i.e., $\ell_\infty, \ell_1, \ell_2$), we further report the accuracy obtained by choosing the worst attack for each input. Finally, we report the accuracy against the union of all attacks ($1 - \mathcal{R}^{\text{max}}_{\text{adv}}$) as well as the average accuracy across perturbation types ($1 - \mathcal{R}^{\text{avg}}_{\text{adv}}$).

Table 4: **Breakdown of all attacks on MNIST models.** For ℓ_∞, we use PGD and the Boundary Attack++ (BAPP) [7]. For ℓ_1, we use our new Sparse ℓ_1-Descent Attack (SLIDE), EAD [8] and the Pointwise Attack (PA) [31]. For ℓ_2, we use PGD, C&W [6] and the Boundary Attack (BA) [3].

		ℓ_∞			ℓ_1				ℓ_2					
Model	Acc.	PGD	BAPP	All ℓ_∞	SLIDE	EAD	PA	All ℓ_1	PGD	C&W	BA	All ℓ_2	$1 - \mathcal{R}^{\text{max}}_{\text{adv}}$	$1 - \mathcal{R}^{\text{avg}}_{\text{adv}}$
Nat	**99.4**	0.0	13.0	0.0	13.0	18.8	72.1	12.4	11.0	10.4	31.0	8.5	0.0	7.0
Adv_∞	**99.1**	91.1	98.5	**91.1**	66.9	58.4	15.0	12.1	78.1	78.4	14.0	11.3	6.8	38.2
Adv_1	98.9	0.0	43.5	0.0	78.6	81.0	91.6	**78.5**	53.0	52.0	69.7	50.6	0.0	43.0
Adv_2	98.5	0.4	78.5	0.4	70.4	69.3	89.7	68.0	74.7	74.5	81.7	**71.8**	0.4	46.7
Adv_{avg}	97.3	76.7	98.0	76.7	66.3	62.4	68.6	53.9	77.7	72.3	64.6	58.3	**49.9**	**63.0**
Adv_{max}	97.2	71.7	98.5	71.7	72.1	70.0	69.6	62.6	75.7	71.8	59.7	56.0	**52.4**	**63.4**

Table 5: **Breakdown of all attacks on CIFAR10 models.** For ℓ_∞, we use PGD. For ℓ_1, we use our new Sparse ℓ_1-descent attack (SLIDE), EAD [8] and the Pointwise Attack (PA) [31].

		ℓ_∞		ℓ_1					
Model	Acc.	PGD	All ℓ_∞	SLIDE	EAD	PA	All ℓ_1	$1 - \mathcal{R}^{\text{max}}_{\text{adv}}$	$1 - \mathcal{R}^{\text{avg}}_{\text{adv}}$
Nat	**95.7**	0.0	0.0	0.2	0.0	29.6	0.0	0.0	0.0
Adv_∞	92.0	71.0	**71.0**	19.4	17.6	52.7	16.4	16.4	44.9
Adv_1	90.8	53.4	53.4	66.6	66.6	84.7	**66.2**	53.1	60.0
Adv_{avg}	91.1	64.1	64.1	61.1	61.5	81.7	60.8	**59.4**	**62.5**
Adv_{max}	91.2	65.7	65.7	63.1	63.0	83.4	62.5	**61.1**	**64.1**

D Gradient Masking as a Consequence of ℓ_∞-Robustness on MNIST.

Multiple works have reported on a curious phenomenon that affects the ℓ_∞-adversarially trained model of Madry et al. [25] on MNIST. This model achieves strong robustness to the ℓ_∞ attacks it was trained on, as one would expect. Yet, on other ℓ_p-norms (e.g., ℓ_1 [8, 31] and ℓ_2 [23, 31]), its robustness is no better—or even worse—than for an undefended model. Some authors have referred to this effect as *overfitting*, a somewhat unfair assessment of the work of [25], as their model actually achieves exactly what it was trained to do—namely resist ℓ_∞-bounded attacks. Moreover, as our theoretical results suggest, this trade-off may be inevitable (a similar point was made in [21]).

The more intriguing aspect of Madry et al.'s MNIST model is that, when attacked by ℓ_1 or ℓ_2 adversaries, first-order attacks are sub-optimal. This was previously observed in [31] and in [23], where decision-based or second-order attacks vastly outperformed gradient descent for finding ℓ_1 or ℓ_2 adversarial examples. Li et al. [23] argue that this effect is due to the gradients of the adversarially trained model having much smaller magnitude than in a standard model. Yet, this

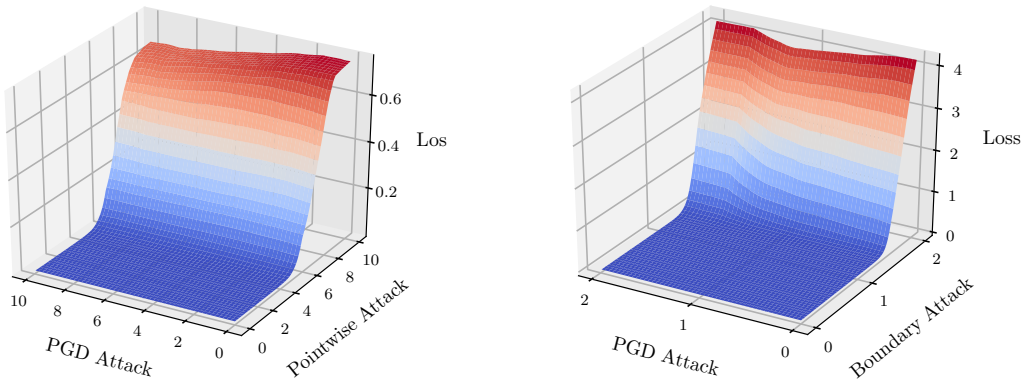


Figure 3: **Gradient masking in an ℓ_∞-adversarially trained model on MNIST, evaluated against ℓ_1-attacks (left) and ℓ_2-attacks (right).** The model is trained against an ℓ_∞-PGD adversary with $\epsilon = 0.3$. For a randomly chosen data point $\boldsymbol{x}$, we compute an adversarial perturbation $\boldsymbol{r}_{\text{PGD}}$ using PGD and $\boldsymbol{r}_{\text{GF}}$ using a gradient-free attack. The left plot is for ℓ_1-attacks with $\epsilon = 10$ and the right plot is for ℓ_2-attacks with $\epsilon = 2$. The plots display the loss on points of the form $\hat{\boldsymbol{x}} := \boldsymbol{x} + \alpha \cdot \boldsymbol{r}_{\text{PGD}} + \beta \cdot \boldsymbol{r}_{\text{GF}}$, for $\alpha, \beta \in [0, \epsilon]$. The loss surface behaves like a step-function, and gradient-free attacks succeed in finding adversarial examples where first-order methods failed.

fails to explain why first-order attacks appear to be optimal in the ℓ_∞-norm that the model was trained against.

A natural explanation for this discrepancy follows from an inspection of the robust model's first layer (as done in [25]). All kernels of the model's first convolutional layer have very small norm, except for three kernels that have a single large weight. This reduces the convolution to a thresholding filter, which we find to be of one of two forms: either $\text{ReLU}(\alpha \cdot (x - 0.3))$ or $\text{ReLU}(\alpha \cdot (x - 0.7))$ for constant $\alpha > 0$.[4] Thus, the model's first layer forms a piece-wise function with three distinct regimes, depending on the value of an input pixel x_i: (1) for $x_i \in [0, 0.3]$, the output is only influenced by the low-weight kernels. For $x_i \in [0.3, 1]$, the $\text{ReLU}(\alpha \cdot (x - 0.3))$ filters become active, and override the signal from the low-weight kernels. For $x_i \in [0.7, 1]$, the $\text{ReLU}(\alpha \cdot (x - 0.7))$ filters are also active.

As most MNIST pixels are in $\{0, 1\}$, ℓ_∞-attacks operate in a regime where most perturbed pixels are in $[0, 0.3] \cup [0.7, 1]$. The model's large-weight ReLUs thus never transition between active and inactive, which leads to a smooth, albeit flat loss that first-order methods navigate effectively.

For ℓ_1 and ℓ_2 attacks however, one would expect some of the ReLUs to be flipped as the attacks can make changes larger that 0.3 to some pixels. Yet, as most MNIST pixels are 0 (the digit's background), nearly all large-weight ReLUs start out inactive, with gradients equal to zero. A first-order adversary thus has no information on which pixels to focus the perturbation budget on.

Decision-based attacks sidestep this issue by disregarding gradients entirely. Figure 3 shows two examples of input points where a decision-based attack (Pointwise Attack for ℓ_1 [31] and Boundary Attack for ℓ_2 [3]) finds an adversarial example in a direction that is orthogonal to the one explored by PGD. The loss surface exhibits sharp thresholding steps, as predicted by our analysis.

When we explicitly train against first-order ℓ_1 or ℓ_2 adversaries (models Adv_1 and Adv_2 in Table 1, left), the resulting model is robust (at least empirically) to ℓ_1 or ℓ_2 attacks. Note that model Adv_∞ actually achieves higher robustness to ℓ_2-PGD attacks than Adv_2 (due to gradient-masking). Thus, the Adv_2 model converged to a *sub-optimal* local minima of its first-order adversarial training procedure (i.e., learning the same thresholding mechanism as Adv_∞ would yield lower loss). Yet, this sub-optimal local minima generalizes much better to other ℓ_2 attacks.

Models trained against ℓ_∞, ℓ_1 and ℓ_2 attacks (i.e., Adv_{all} and Adv_{max}) in Table 1, left) also learn to use thresholding to achieve robustness to ℓ_∞ attacks, while masking gradients for ℓ_1 and

[4] Specifically, for the "secret" model of Madry et al., the three thresholding filters are approximately $\text{ReLU}(0.6 \cdot (x - 0.3))$, $\text{ReLU}(1.34 \cdot (x - 0.3))$ and $\text{ReLU}(0.86 \cdot (x - 0.7))$.

ℓ_2 attacks.

E Examples of Affine Combinations of Perturbations

In Figure 4, we display examples of ℓ_1, ℓ_∞ and rotation-translation attacks on MNIST and CIFAR10, as well as affine attacks that interpolate between two attack types.

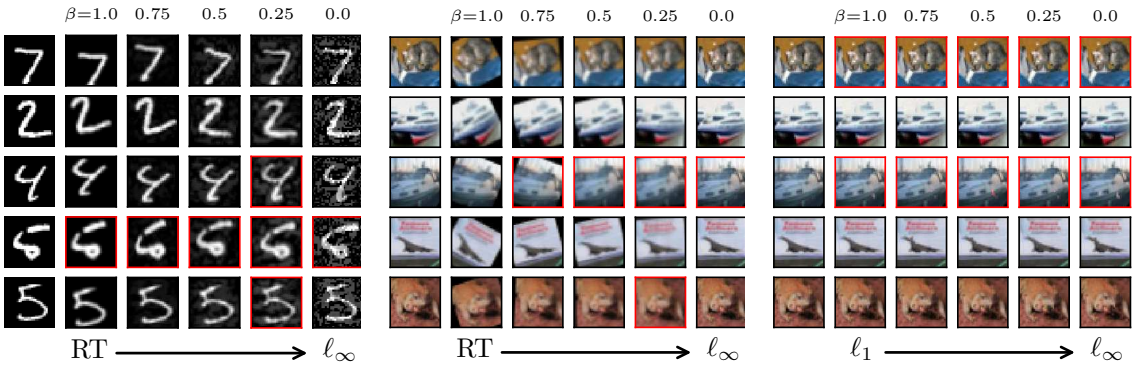


Figure 4: **Adversarial examples for ℓ_∞, ℓ_1 and rotation-translation (RT) attacks, and affine combinations thereof.** The first column in each subplot shows clean images. The following five images in each row linearly interpolate between two attack types, as described in Section 2.5. Images marked in red are mis-classified by a model trained against both types of perturbations. Note that there are examples for which combining a rotation-translation and ℓ_∞-attack is stronger than either perturbation type individually.

F Proof of Theorem 1 (Robustness trade-off between ℓ_∞ and ℓ_1- norms)

Our proof follows a similar structure to the proof of Theorem 2.1 in [41], although the analysis is slightly simplified in our case as we are comparing two perturbation models, an ℓ_∞-bounded one and an ℓ_1-bounded one, that are essentially orthogonal to each other. With a perturbation of size $\epsilon = 2\eta$, the ℓ_∞-bounded noise can "flip" the distribution of the features $x_1, \ldots, x_d$ to reflect the opposite label, and thus destroy any information that a classifier might extract from those features. On the other side, an ℓ_1-bounded perturbation with $\epsilon = 2$ can flip the distribution of x_0. By sacrificing some features, a classifier can thus achieve some robustness to either ℓ_∞ *or* ℓ_1 noise, but never to both simultaneously.

For $y \in \{-1, +1\}$, let $\mathcal{G}^y$ be the distribution over feature x_0 conditioned on the value of y. Similarly, let $\mathcal{H}^y$ be the conditional distribution over features $x_1, \ldots, x_d$. Consider the following perturbations: $\boldsymbol{r}_\infty = [0, -2y\eta, \ldots, -2y\eta]$ has small ℓ_∞-norm, and $\boldsymbol{r}_1 = [-2x_0, 0, \ldots, 0]$ has small ℓ_1-norm. The ℓ_∞ perturbation can change $\mathcal{H}^y$ to $\mathcal{H}^{-y}$, while the ℓ_1 perturbation can change $\mathcal{G}^y$ to $\mathcal{G}^{-y}$.

Let $f(\boldsymbol{x})$ be any classifier from $\mathbb{R}^{d+1}$ to $\{-1, +1\}$ and define:

$$p_{+-} = \Pr_{\boldsymbol{x} \sim (\mathcal{G}^{+1}, \mathcal{H}^{-1})}[f(\boldsymbol{x}) = +1] \,, \qquad p_{-+} = \Pr_{\boldsymbol{x} \sim (\mathcal{G}^{-1}, \mathcal{H}^{+1})}[f(\boldsymbol{x}) = +1] \,.$$

The accuracy of f against the $\boldsymbol{r}_\infty$ perturbation is given by:

$$\Pr[f(\boldsymbol{x} + \boldsymbol{r}_\infty) = y] = \Pr[y = +1] \cdot p_{+-} + \Pr[y = -1] \cdot (1 - p_{-+}) = \frac{1}{2} \cdot (1 + p_{+-} - p_{-+}) \,.$$

Similarly, the accuracy of f against the $\boldsymbol{r}_1$ perturbation is:

$$\Pr[f(\boldsymbol{x}+\boldsymbol{r}_1)=y] = \Pr[y=+1]\cdot p_{-+} + \Pr[y=-1]\cdot(1-p_{+-}) = \frac{1}{2}\cdot(1+p_{-+}-p_{+-})\ .$$

Combining these, we get $\Pr[f(\boldsymbol{x}+\boldsymbol{r}_\infty)=y] + \Pr[f(\boldsymbol{x}+\boldsymbol{r}_1)=y] = 1$.

As $\boldsymbol{r}_\infty$ and $\boldsymbol{r}_1$ are two specific ℓ_∞- and ℓ_1-bounded perturbations, the above is an upper-bound on the accuracy that f achieves against worst-case perturbation within the prescribed noise models, which concludes the proof.

□

G Proof of Theorem 2 (Robustness trade-off between ℓ_∞ and spatial perturbations)

The proof of this theorem follows a similar blueprint to the proof of Theorem 1. Recall that an ℓ_∞ perturbation with $\epsilon = 2\eta$ can flip the distribution of the features $x_1,\ldots,x_n$ to reflect an opposite label y. The tricky part of the proof is to show that a small rotation or translation can flip the distribution of x_0 to the opposite label, without affecting the marginal distribution of the other features too much.

Recall that we model rotations and translations as picking a permutation π from some fixed set Π of permutations over the indices in $\boldsymbol{x}$, with the constraint that feature x_0 be moved to at most N different positions for all $\pi\in\Pi$.

We again define $\mathcal{G}^y$ as the distribution of x_0 conditioned on y, and $\mathcal{H}^y$ for the distribution of $x_1,\ldots,x_d$. We know that a small ℓ_∞-perturbation can transform $\mathcal{H}^y$ into $\mathcal{H}^{-y}$. Our goal is to show that a rotation-translation adversary can change $(\mathcal{G}^y,\mathcal{H}^y)$ into a distribution that is very close to $(\mathcal{G}^{-y},\mathcal{H}^y)$. The result of the theorem then follows by arguing that no binary classifier f can distinguish, with high accuracy, between ℓ_∞-perturbed examples with label y and rotated examples with label $-y$ (and vice versa).

We first describe our proof idea at a high level. We define an intermediate "hybrid" distribution $\mathcal{Z}^y$ where all $d+1$ features are i.i.d $N(y\eta,1)$ (that is, x_0 now has the same distribution as the other weakly-correlated features). The main step in the proof is to show that for samples from either $(\mathcal{G}^y,\mathcal{H}^y)$ or $(\mathcal{G}^{-y},\mathcal{H}^y)$, a random rotation-translation yields a distribution that is very close (in total variation) to $\mathcal{Z}^y$. From this, we then show that there exists an adversary that applies two rotations or translations in a row, to first transform samples from $(\mathcal{G}^y,\mathcal{H}^y)$ into samples close to $\mathcal{Z}^y$, and then transform those samples into ones that are close to $(\mathcal{G}^{-y},\mathcal{H}^y)$.

We will need a standard version of the Berry-Esseen theorem, stated hereafter for completeness.

Theorem 5 (Berry-Esseen [2])**.** *Let $X_1,\ldots,X_n$ be independent random variables with $\mathbb{E}[X_i]=\mu_i$, $\mathbb{E}[X_i^2]=\sigma_i^2>0$, and $\mathbb{E}[|X_i|^3]=\rho_i<\infty$, where the μ_i,σ_i and ρ_i are constants independent of n. Let $S_n = X_1+\cdots+X_n$, with $F_n(x)$ the CDF of S_n and $\Phi(x)$ the CDF of the standard normal distribution. Then,*

$$\sup_{x\in\mathbb{R}}\left|F_n(x) - \Phi\left(\frac{x-\mathbb{E}[S_n]}{\sqrt{\mathrm{Var}\,[S_n]}}\right)\right| = O(1/\sqrt{n})\ .$$

For distributions $\mathcal{P},\mathcal{Q}$, let $\Delta_{\mathrm{TV}}(\mathcal{P},\mathcal{Q})$ denote their total-variation distance. The below lemma is the main technical result we need, and bounds the total variation between a multivariate Gaussian $\mathcal{P}$ and a special mixture of multivariate Gaussians $\mathcal{Q}$.

Lemma 6. *For $k>1$, let $\mathcal{P}$ be a k-dimensional Gaussians with mean $\boldsymbol{\mu}_P=[\lambda_P,\ldots,\lambda_P]$ and identity covariance. For all $i\in[k]$, let $\mathcal{Q}_i$ be a multivariate Gaussian with mean $\boldsymbol{\mu}_i$ and diagonal covariance $\boldsymbol{\Sigma}_i$ where $(\boldsymbol{\mu}_i)_j = \begin{cases}\lambda_Q & \text{if } i=j\\ \lambda_P & \text{otherwise}\end{cases}$ and $(\boldsymbol{\Sigma}_i)_{(j,j)} = \begin{cases}\sigma_Q^2 & \text{if } i=j\\ 1 & \text{otherwise}\end{cases}$. Define $\mathcal{Q}$ as a mixture distribution of the $\mathcal{Q}_1,\ldots,\mathcal{Q}_k$ with probabilities $1/k$. Assuming that $\lambda_P,\lambda_Q,\sigma_Q$ are constants independent of k, we have $\Delta_{TV}(\mathcal{P},\mathcal{Q}) = O(1/\sqrt{k})$.*

Proof. [5] Let $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ denote, respectively, the pdfs of $\mathcal{P}$ and $\mathcal{Q}$. Note that $q(\boldsymbol{x}) = \sum_{i=1}^{k} \frac{1}{k} q_i(\boldsymbol{x})$, where $q_i(\boldsymbol{x})$ is the pdf of $\mathcal{Q}_i$. We first compute:

$$\begin{aligned} q(\boldsymbol{x}) &= \sum_{i=1}^{k} \frac{1}{k} \frac{1}{\sqrt{(2\pi)^k \cdot |\boldsymbol{\Sigma}_i|}} \cdot e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)} \\ &= \frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_P)^T(\boldsymbol{x}-\boldsymbol{\mu}_P)}}{\sqrt{(2\pi)^k}} \cdot \frac{1}{k \cdot \sigma_Q^2} \cdot \sum_{i=1}^{k} e^{-\frac{1}{2}t(x_i)} \\ &= p(\boldsymbol{x}) \cdot \frac{1}{k \cdot \sigma_Q^2} \cdot \sum_{i=1}^{k} e^{-\frac{1}{2}t(x_i)} \ , \end{aligned}$$

where

$$t(x_i) \coloneqq (\sigma_Q^{-2} - 1)x_i^2 - (2\lambda_Q \sigma_Q^{-2} - 2\lambda_P)x_i + (\lambda_Q^2 \sigma_Q^{-2} - \lambda_P^2) \ . \tag{3}$$

Thus we have that

$$q(\boldsymbol{x}) < p(\boldsymbol{x}) \quad \iff \quad \frac{1}{k \cdot \sigma_Q^2} \cdot \sum_{i=1}^{k} e^{-\frac{1}{2}t(x_i)} < 1 \ .$$

The total-variation distance between $\mathcal{P}$ and $\mathcal{Q}$ is then $\Delta_{\mathrm{TV}}(\mathcal{P}, \mathcal{Q}) = p_1 - p_2$, where

$$\begin{gathered} p_1 \coloneqq \Pr\left[S_k < k \cdot \sigma_Q^2\right] \ , \quad p_2 \coloneqq \Pr\left[T_k < k \cdot \sigma_Q^2\right] \ , \\ S_k \coloneqq \sum_{i=1}^{k} U_i \ , \quad T_k \coloneqq S_{k-1} + V_k \ , \quad U_i \coloneqq e^{-\frac{1}{2}t(Z_i)} \ , \quad V_n \coloneqq e^{-\frac{1}{2}t(W_n)} \ , \end{gathered} \tag{4}$$

and the $Z_i \sim \mathcal{N}(\lambda_P, 1)$, $W_n \sim \mathcal{N}(\lambda_Q, \sigma_Q^2)$ and all the Z_i and W_n are mutually independent.

It is easy to verify that $\mathbb{E}[U_i] = \sigma_Q^2$, $\mathrm{Var}[U_i] = O(1)$, $\mathbb{E}[U_i^3] = O(1)$, $\mathbb{E}[W_n] = O(1)$, $\mathrm{Var}[W_n] = O(1), \mathbb{E}[W_n^3] = O(1)$. Then, applying the Berry-Esseen theorem, we get:

$$\begin{aligned} p_1 &= \Pr\left[S_k < k \cdot \sigma_Q^2\right] = \Phi(0) + O\left(\frac{1}{\sqrt{k}}\right) = \frac{1}{2} + O\left(\frac{1}{\sqrt{k}}\right) \ , \\ p_2 &= \Pr\left[T_k < k \cdot \sigma_Q^2\right] = \Phi\left(\frac{k \cdot \sigma_Q^2 - \mathbb{E}[T_k]}{\sqrt{\mathrm{Var}[T_k]}}\right) + O\left(\frac{1}{\sqrt{k}}\right) = \Phi\left(O\left(\frac{1}{\sqrt{k}}\right)\right) + O\left(\frac{1}{\sqrt{k}}\right) \\ &= \frac{1}{2} + O\left(\frac{1}{\sqrt{k}}\right) \ . \end{aligned}$$

And thus,

$$\Delta_{\mathrm{TV}}(\mathcal{P}, \mathcal{Q}) = p_1 - p_2 = O(1/\sqrt{k}) \ . \tag{5}$$

□

We now define a rotation-translation adversary $\mathcal{A}$ with a budget of N. It samples a random permutation from the set Π of permutations that switch position 0 with a position in $[0, N-1]$ and leave all other positions fixed (note that $|\Pi| = N$). Let $\mathcal{A}(\mathcal{G}^y, \mathcal{H}^y)$ denote the distribution resulting from applying $\mathcal{A}$ to $(\mathcal{G}^y, \mathcal{H}^y)$ and define $\mathcal{A}(\mathcal{G}^{-y}, \mathcal{H}^y)$ similarly. Recall that $\mathcal{Z}^y$ is a hybrid distribution which has all features distributed as $\mathcal{N}(y\eta, 1)$.

Claim 7. *$\Delta_{TV}(\mathcal{A}(\mathcal{G}^y, \mathcal{H}^y), \mathcal{Z}^y) = O(1/\sqrt{N})$ and $\Delta_{TV}(\mathcal{A}(\mathcal{G}^{-y}, \mathcal{H}^y), \mathcal{Z}^y) = O(1/\sqrt{N})$*

Proof. For the first N features, samples output by $\mathcal{A}$ follow exactly the distribution $\mathcal{Q}$ from Lemma (6), for $k = N$ and $\lambda_P = y \cdot \eta, \lambda_Q = y, \sigma_Q^2 = \alpha^{-2}$. Note that in this case, the distribution $\mathcal{P}$ has each feature distributed as in $\mathcal{Z}^y$. Thus, Lemma (6) tells us that the distribution of the

[5] We thank Iosif Pinelis for his help with this proof (`https://mathoverflow.net/questions/325409/`).

first N features is the same as in $\mathcal{Z}^y$, up to a total-variation distance of $O(1/\sqrt{N})$. As features $x_N \ldots, x_d$ are unaffected by $\mathcal{A}$ and thus remain distributed as in $\mathcal{Z}^y$, we conclude that the total-variation distance between $\mathcal{A}$'s outputs and $\mathcal{Z}^y$ is $O(1/\sqrt{N})$.

The proof for $\mathcal{A}(\mathcal{G}^{-y}, \mathcal{H}^y)$ is similar, except that we apply Lemma (6) with $\lambda_Q = -y$. □

Let $\tilde{\mathcal{Z}}^y$ be the true distribution $\mathcal{A}(\mathcal{G}^{-y}, \mathcal{H}^y)$, which we have shown to be close to $\mathcal{Z}^y$. Consider the following "inverse" adversary $\mathcal{A}^{-1}$. This adversary samples $\boldsymbol{z} \sim \tilde{\mathcal{Z}}^y$ and returns $\pi^{-1}(\boldsymbol{z})$, for $\pi \in \mathit{\Pi}$, with probability

$$\frac{1}{|\mathit{\Pi}|} \cdot \frac{f_{(\mathcal{G}^{-y}, \mathcal{H}^y)}(\pi^{-1}(\boldsymbol{z}))}{f_{\tilde{\mathcal{Z}}^y}(\boldsymbol{z})} ,$$

where $f_{(\mathcal{G}^{-y}, \mathcal{H}^y)}$ and $f_{\tilde{\mathcal{Z}}^y}$ are the probability density functions for $(\mathcal{G}^{-y}, \mathcal{H}^y)$ and for $\tilde{\mathcal{Z}}^y$.

Claim 8. *$\mathcal{A}^{-1}$ is a RT adversary with budget N that transforms $\tilde{\mathcal{Z}}^y$ into $(\mathcal{G}^{-y}, \mathcal{H}^y)$.*

Proof. Note that $\mathcal{A}^{-1}$ always applies the inverse of a perturbation in $\mathit{\Pi}$. So feature x_0 gets sent to at most N positions when perturbed by $\mathcal{A}^{-1}$.

Let Z be a random variable distributed as $\tilde{\mathcal{Z}}^y$ and let h be the density function of the distribution obtained by applying $\mathcal{A}^{-1}$ to Z. We compute:

$$\begin{aligned} h(\boldsymbol{x}) &= \sum_{\pi \in \mathit{\Pi}} f_{\tilde{\mathcal{Z}}^y}(\pi(\boldsymbol{x})) \cdot \Pr\big[\mathcal{A}^{-1} \text{ picks permutation } \pi \mid Z = \pi(\boldsymbol{x})\big] \\ &= \sum_{\pi \in \mathit{\Pi}} f_{\tilde{\mathcal{Z}}^y}(\pi(\boldsymbol{x})) \cdot \frac{1}{|\mathit{\Pi}|} \cdot \frac{f_{(\mathcal{G}^{-y}, \mathcal{H}^y)}(\pi(\pi^{-1}(\boldsymbol{x})))}{f_{\tilde{\mathcal{Z}}^y}(\pi(\boldsymbol{x}))} = \sum_{\pi \in \mathit{\Pi}} \frac{1}{|\mathit{\Pi}|} \cdot f_{(\mathcal{G}^{-y}, \mathcal{H}^y)}(\boldsymbol{x}) \\ &= f_{(\mathcal{G}^{-y}, \mathcal{H}^y)}(\boldsymbol{x}) , \end{aligned}$$

so applying $\mathcal{A}^{-1}$ to $\tilde{\mathcal{Z}}^y$ does yield the distribution $(\mathcal{G}^{-y}, \mathcal{H}^y)$. □

We can now finally define our main rotation-translation adversary, $\mathcal{A}^*$. The adversary first applies $\mathcal{A}$ to samples from $(\mathcal{G}^y, \mathcal{H}^y)$, and then applies $\mathcal{A}^{-1}$ to the resulting samples from $\tilde{\mathcal{Z}}^y$.

Claim 9. *The adversary $\mathcal{A}^*$ is a rotation-translation adversary with budget N. Moreover,*

$$\mathit{\Delta}_{TV}\big(\mathcal{A}^*(\mathcal{G}^y, \mathcal{H}^y), (\mathcal{G}^{-y}, \mathcal{H}^y)\big) = O(1/\sqrt{N}) .$$

Proof. The adversary $\mathcal{A}^*$ first switches x_0 with some random position in $[0, N-1]$ by applying $\mathcal{A}$. Then, $\mathcal{A}^{-1}$ either switches x_0 back into its original position or leaves it untouched. Thus, $\mathcal{A}^*$ always moves x_0 into one of N positions. The total-variation bound follows by the triangular inequality:

$$\begin{aligned} &\mathit{\Delta}_{\mathrm{TV}}\big(\mathcal{A}^*(\mathcal{G}^y, \mathcal{H}^y), (\mathcal{G}^{-y}, \mathcal{H}^y)\big) \\ &\quad = \mathit{\Delta}_{\mathrm{TV}}\big(\mathcal{A}^{-1}(\mathcal{A}(\mathcal{G}^y, \mathcal{H}^y)), (\mathcal{G}^{-y}, \mathcal{H}^y)\big) \\ &\quad \le \mathit{\Delta}_{\mathrm{TV}}\big(\mathcal{A}^{-1}(\mathcal{Z}^y), (\mathcal{G}^{-y}, \mathcal{H}^y)\big) + \mathit{\Delta}_{\mathrm{TV}}\left(\mathcal{Z}^y, \mathcal{A}(\mathcal{G}^y, \mathcal{H}^y)\right) \\ &\quad \le \underbrace{\mathit{\Delta}_{\mathrm{TV}}\Big(\mathcal{A}^{-1}(\tilde{\mathcal{Z}}^y), (\mathcal{G}^{-y}, \mathcal{H}^y)\Big)}_{0} + \underbrace{\mathit{\Delta}_{\mathrm{TV}}\Big(\tilde{\mathcal{Z}}^y, (\mathcal{G}^{-y}, \mathcal{H}^y)\Big)}_{O(1/\sqrt{N})} + \underbrace{\mathit{\Delta}_{\mathrm{TV}}\left(\mathcal{Z}^y, \mathcal{A}(\mathcal{G}^y, \mathcal{H}^y)\right)}_{O(1/\sqrt{N})} \\ &\quad = O(1/\sqrt{N}) . \end{aligned}$$

□

To conclude the proof, we define:

$$\begin{aligned} p_{+-} &= \Pr_{\boldsymbol{x} \sim (\mathcal{G}^{+1}, \mathcal{H}^{-1})}[f(\boldsymbol{x}) = +1] , \qquad p_{-+} = \Pr_{\boldsymbol{x} \sim (\mathcal{G}^{-1}, \mathcal{H}^{+1})}[f(\boldsymbol{x}) = +1] , \\ \tilde{p}_{-+} &= \Pr_{\boldsymbol{x} \sim \mathcal{A}^*(\mathcal{G}^{+1}, \mathcal{H}^{+1})}[f(\boldsymbol{x}) = +1] , \qquad \tilde{p}_{+-} = \Pr_{\boldsymbol{x} \sim (\mathcal{G}^{-1}, \mathcal{H}^{-1})}[f(\boldsymbol{x}) = +1] . \end{aligned}$$

Then,

$$\begin{aligned}\Pr[f(\boldsymbol{x}+\boldsymbol{r}_\infty)=y]+\Pr[f(A^*(\boldsymbol{x}))=y] &= \frac{1}{2}p_{+-}+\frac{1}{2}(1-p_{-+})+\frac{1}{2}\tilde{p}_{-+}+\frac{1}{2}(1-\tilde{p}_{+-})\\ &= 1+\frac{1}{2}\left(p_{+-}-\tilde{p}_{+-}\right)+\frac{1}{2}\left(p_{-+}-\tilde{p}_{-+}\right)\\ &\leq 1-O(1/\sqrt{N})\ .\end{aligned}$$

□

G.1 Numerical Estimates for the Robustness Trade-off in Theorem 2

While the robustness trade-off we proved in Theorem 2 is asymptotic in N (the budget of the RT adversary), we can provide tight numerical estimates for this trade-off for concrete parameter settings:

Remark 10. Let $d \geq 200$, $\alpha = 2$ and $N = 49$ (e.g., translations by ± 3 pixels). Then, there exists a classifier with $\mathcal{R}_{\text{adv}}(f; S_\infty) < 10\%$, as well as a (distinct) classifier with $\mathcal{R}_{\text{adv}}(f; S_{\text{RT}}) < 10\%$. Yet, any single classifier satisfies $\mathcal{R}^{\text{avg}}_{\text{adv}}(f; S_\infty, S_{\text{RT}}) \gtrapprox 0.425$.

We first show the existence of classifiers with $\mathcal{R}_{\text{adv}} < 10\%$ for the given ℓ_∞ and RT attacks.

Let $f(\boldsymbol{x}) = \text{sign}(x_0)$ and let $\boldsymbol{r} = [-y\epsilon, 0, \ldots, 0]$ be the worst-case perturbation with $\|\boldsymbol{r}\| \leq \epsilon$. Recall that $\epsilon = 2\eta = 4/\sqrt{d}$. We have

$$\Pr[f(\boldsymbol{x}+\boldsymbol{r}) \neq y] = \Pr\Big[\mathcal{N}(1, 1/4) - 4/\sqrt{d} < 0\Big] \leq \Pr\Big[\mathcal{N}(1 - 4/\sqrt{200}, 1/4) < 0\Big] \leq 8\%\ .$$

Thus, f achieves $\mathcal{R}_{\text{adv}} < 10\%$ against the ℓ_∞-perturbations.

Let $g(\boldsymbol{x}) = \text{sign}(\sum_{i=N}^{d} x_i)$ be a classifier that ignores all feature positions that a RT adversary $\mathcal{A}$ may affect. We have

$$\begin{aligned}\Pr[g(\mathcal{A}(\boldsymbol{x})) \neq y] = \Pr[g(\boldsymbol{x}) \neq y] &= \Pr\left[\mathcal{N}\left((d-N+1)\cdot\eta, d-N+1\right) < 0\right]\\ &\leq \Pr\Big[\mathcal{N}(2\sqrt{d-48}/\sqrt{d}, 1) < 0\Big] \leq 5\%\ .\end{aligned}$$

Thus, g achieves $\mathcal{R}_{\text{adv}} < 10\%$ against RT perturbations.

We upper-bound the adversarial risk that any classifier must incur against both attacks by numerically estimating the total-variation distance between the distributions induced by the RT and ℓ_∞ adversaries for inputs of opposing labels y. Specifically, we generate 100,000 samples from the distributions $\mathcal{G}^{+1}, \mathcal{G}^{-1}$ and $\mathcal{H}^{+1}$ as defined in the proof of Theorem 2, and obtain an estimate of the total-variation distance in Lemma (9). For this, we numerically estimate p_1 and p_2 as defined in Equation (4).

H Proof of Claim 3 (Affine combinations of ℓ_p- perturbations do not affect linear models)

Let

$$\max_{\boldsymbol{r}\in S_{\text{U}}} \boldsymbol{w}^T\boldsymbol{r} = v_{\max}, \quad \text{and} \quad \min_{\boldsymbol{r}\in S_{\text{U}}} \boldsymbol{w}^T\boldsymbol{r} = v_{\min}\ .$$

Let $S_{\text{U}} \coloneqq S_p \cup S_q$. Note that any $\boldsymbol{r} \in S_{\text{affine}}$ is of the form $\beta\boldsymbol{r}_1 + (1-\beta)\boldsymbol{r}_2$ for $\beta \in [0,1]$. Moreover, we have $\boldsymbol{r}_1 \in S_p \subset S_{\text{U}}$ and $\boldsymbol{r}_2 \in S_q \subset S_{\text{U}}$. Thus,

$$\max_{\boldsymbol{r}\in S_{\text{affine}}} \boldsymbol{w}^T\boldsymbol{r} = v_{\max}, \quad \text{and} \quad \min_{\boldsymbol{r}\in S_{\text{affine}}} \boldsymbol{w}^T\boldsymbol{r} = v_{\min}\ .$$

Let $h(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$, so that $f(\boldsymbol{x}) = \text{sign}(h(\boldsymbol{x}))$. Then, we get

$$\begin{aligned}
\Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{affine}} : f(\boldsymbol{x}+\boldsymbol{r}) \neq y\right] &= \frac{1}{2}\Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{affine}} : \boldsymbol{w}^T\boldsymbol{r} < -h(\boldsymbol{x}) \mid y = +1\right] \\
&\quad + \frac{1}{2}\Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{affine}} : \boldsymbol{w}^T\boldsymbol{r} > h(\boldsymbol{x}) \mid y = -1\right] \\
&= \frac{1}{2}\Pr_{\mathcal{D}}\left[v_{\min} < -h(\boldsymbol{x}) \mid y = +1\right] + \frac{1}{2}\Pr_{\mathcal{D}}\left[v_{\max} > h(\boldsymbol{x}) \mid y = -1\right] \\
&= \frac{1}{2}\Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{U}} : \boldsymbol{w}^T\boldsymbol{r} < -h(\boldsymbol{x}) \mid y = +1\right] \\
&\quad + \frac{1}{2}\Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{U}} : \boldsymbol{w}^T\boldsymbol{r} > h(\boldsymbol{x}) \mid y = -1\right] \\
&= \Pr_{\mathcal{D}}\left[\exists \boldsymbol{r} \in S_{\text{U}} : f(\boldsymbol{x}+\boldsymbol{r}) \neq y\right] \ .
\end{aligned}$$

□

I Affine combinations of ℓ_p- perturbations can affect non-linear models

In Section 2.5, we showed that for linear models, robustness to a union of ℓ_p-perturbations implies robustness to an affine adversary that interpolates between perturbation types. We show that this need not be the case when the model is non-linear. In particular, we can show that for the distribution $\mathcal{D}$ introduced in Section 2, non-linearity is necessary to achieve robustness to a union of ℓ_∞ and ℓ_1-perturbations (with different parameter settings than for Theorem 1), but that at the same time, robustness to affine combinations of these perturbations is unattainable by any model.

Theorem 11. *Consider the distribution $\mathcal{D}$ with $d \geq 200$, $\alpha = 2$ and $p_0 = 1 - \Phi(-2)$. Let S_∞ be the set of ℓ_∞-bounded perturbation with $\epsilon = (3/2)\eta = 3/\sqrt{d}$ and let S_1 be the set of ℓ_1-bounded perturbations with $\epsilon = 3$. Define S_{affine} as in Section 2.5. Then, there exists a non-linear classifier g that achieves $\mathcal{R}_{adv}^{max}(g; S_\infty, S_1) \leq 35\%$. Yet, for all classifiers f we have $\mathcal{R}_{adv}(f; S_{affine}) \geq 50\%$.*

Proof. We first prove that no classifier can achieve accuracy above 50% (which is achieved by the constant classifier) against S_{affine}. The proof is very similar to the one of Theorem 1.

Let $\beta = 2/3$, so the affine attacker gets to compose an ℓ_∞-budget of $2/\sqrt{d}$ and an ℓ_1-budget of 1. Specifically, for a point $(\boldsymbol{x}, y) \sim \mathcal{D}$, the affine adversary will apply the perturbation

$$\boldsymbol{r} = [-x_0, -y\frac{2}{\sqrt{d}}, \ldots, -y\frac{2}{\sqrt{d}}] = [-x_0, -y\eta, \ldots, -y\eta] \ .$$

Let $\mathcal{G}^{0,0}$ be the following distribution:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \quad x_0 = 0, \quad x_1, \ldots, x_d \overset{i.i.d}{\sim} \mathcal{N}(0, 1) \ .$$

Note that in $\mathcal{G}^{0,0}$, $\boldsymbol{x}$ is independent of y so no classifier can achieve more than 50% accuracy on $\mathcal{G}^{0,0}$. Yet, note that the affine adversary's perturbation $\boldsymbol{r}$ transforms any $(\boldsymbol{x}, y) \sim \mathcal{D}$ into $(\boldsymbol{x}, y) \sim \mathcal{G}^{0,0}$.

We now show that there exists a classifier that achieves non-trivial robustness against the set of perturbations $S_\infty \cup S_1$, i.e., the union of ℓ_∞-noise with $\epsilon = 3/\sqrt{d}$ and ℓ_1-noise with $\epsilon = 3$. Note that by Claim 3, this classifier must be *non-linear*. We define

$$f(\boldsymbol{x}) = \text{sign}\left(3 \cdot \text{sign}(x_0) + \sum_{i=1}^{d} \frac{2}{\sqrt{d}} \cdot x_i\right) \ .$$

The reader might notice that $f(\boldsymbol{x})$ closely resembles the *Bayes optimal classifier* for $\mathcal{D}$ (which would be a linear classifier). The non-linearity in f comes from the sign function applied to x_0.

Intuitively, this limits the damage caused by the ℓ_1-noise, as $\text{sign}(x_0)$ cannot change by more than ± 2 under any perturbation of x_0. This forces the ℓ_1 perturbation budget to be "wasted" on the other features $x_1, \ldots, x_d$, which are very robust to ℓ_1 attacks.

As a warm-up, we compute the classifier's natural accuracy on $\mathcal{D}$. For $(\boldsymbol{x}, y) \sim \mathcal{D}$, let $X = y \cdot \sum_{i=1}^{d} \frac{2}{\sqrt{d}} \cdot x_i$ be a random variable. Recall that $\eta = 2/\sqrt{d}$. Note that X is distributed as

$$y \cdot \sum_{i=1}^{d} \frac{2}{\sqrt{d}} \cdot \mathcal{N}(y\eta, 1) = \sum_{i=1}^{d} \frac{2}{\sqrt{d}} \cdot \mathcal{N}\left(\frac{2}{\sqrt{d}}, 1\right) = \sum_{i=1}^{d} \mathcal{N}\left(\frac{4}{d}, \frac{4}{d}\right) = \mathcal{N}(4, 4) \ .$$

Recall that $x_0 = y$ with probability $p_0 = 1 - \Phi(-2) \approx 0.977$. We get:

$$\begin{aligned}
\Pr_{\mathcal{D}}[f(\boldsymbol{x}) = y] &= \Pr_{\mathcal{D}}\left[y \cdot \left(3 \cdot \text{sign}(x_0) + \sum_{i=1}^{d} \frac{2}{\sqrt{d}} \cdot x_i\right) > 0\right] \\
&= \Pr_{\mathcal{D}}[x_0 = y] \cdot \Pr_{\mathcal{D}}\left[3 \cdot y \cdot \text{sign}(x_0) + X > 0 \mid x_0 = y\right] \\
&\quad + \Pr_{\mathcal{D}}[x_0 \neq y] \cdot \Pr_{\mathcal{D}}\left[3 \cdot y \cdot \text{sign}(x_0) + X > 0 \mid x_0 \neq y\right] \\
&= p \cdot \Pr\left[3 + \mathcal{N}(4, 4) > 0\right] + (1 - p) \cdot \Pr\left[-3 + \mathcal{N}(4, 4) > 0\right] \approx 99\% \ .
\end{aligned}$$

We now consider an adversary that picks either an ℓ_∞-perturbation with $\epsilon = 3/\sqrt{d}$ or an ℓ_1-perturbation with $\epsilon = 3$. It will suffice to consider the case where $x_0 = y$. Note that the ℓ_∞ classifier cannot meaningfully perturb x_0, and the best perturbation is always $\boldsymbol{r}_\infty = [0, -y3/\sqrt{d}, \ldots, -y3/\sqrt{d}]$. Moreover, the best ℓ_1-bounded perturbation is $\boldsymbol{r}_1 = [-2y, -y, 0, \ldots, 0]$. We have $f(\boldsymbol{x} + \boldsymbol{r}_\infty) = \text{sign}(y \cdot (3 + X - 6))$ and $f(\boldsymbol{x} + \boldsymbol{r}_1) = \text{sign}(y \cdot (-3 + X - 2/\sqrt{d}))$. We now lower-bound the classifier's accuracy under the union $S_{\text{U}} \coloneqq S_\infty \cup S_1$ of these two perturbation models:

$$\begin{aligned}
\Pr_{\mathcal{D}}[f(\boldsymbol{x} + \boldsymbol{r}) = y, \forall \boldsymbol{r} \in S_{\text{U}}] &\geq \Pr_{\mathcal{D}}[x_0 = y] \cdot \Pr_{\mathcal{D}}[f(\boldsymbol{x} + \boldsymbol{r}) = y, \forall \boldsymbol{r} \in S_{\text{U}} \mid x_0 = y] \\
&\geq p \cdot \Pr_{\mathcal{D}}\left[(3 + X - 6 > 0) \wedge (-3 + X - 2/\sqrt{d}) > 0)\right] \\
&= p \cdot \Pr\left[\mathcal{N}(4, 4) > 3 + 2/\sqrt{d}\right] \geq 65\% \quad (\text{for } d \geq 200) \ .
\end{aligned}$$

□

J Proof of Theorem 4 (Affine combinations of ℓ_∞- and spatial perturbations can affect linear models)

Note that our definition of affine perturbation allows for a different weighting parameter β to be chosen for each input. Thus, the adversary that selects perturbations from S_{affine} is at least as powerful as the one that selects perturbations from $S_\infty \cup S_{\text{RT}}$. All we need to show to complete the proof is that there exists some input $\boldsymbol{x}$ that the affine adversary can perturb, while the adversary limited to the union of spatial and ℓ_∞ perturbations cannot.

Without loss of generality, assume that the RT adversary picks a permutation that switches x_0 with a position in $[0, N-1]$, and leaves all other indices untouched. The main idea is that for any input $\boldsymbol{x}$ where the RT adversary moves x_0 to position $j < N - 1$, the RT adversary with budget N is no more powerful than one with budget $j + 1$. The affine adversary can thus limit its rotation-translation budget and use the remaining budget on an extra ℓ_∞ perturbation.

We now construct an input $\boldsymbol{x}$ such that: (1) $\boldsymbol{x}$ cannot be successfully attacked by an RT adversary (with budget N) or by an ℓ_∞-adversary (with budget ϵ); (2) $\boldsymbol{x}$ can be attacked by an affine adversary.

Without loss of generality, assume that $w_1 = \min\{w_1, \ldots, w_{N-1}\}$, i.e., among all the features that x_0 can be switched with, x_1 has the smallest weight. Let $y = +1$, and let $x_1, \ldots, x_{N-1}$ be

chosen such that $\arg\min\{x_1, \ldots, x_{N-1}\} = 1$. We set

$$x_0 \coloneqq \frac{\epsilon \cdot \|\boldsymbol{w}\|_1}{w_0 - w_1} + x_1 \ .$$

Moreover, set $x_N, \ldots, x_d$ such that

$$\boldsymbol{w}^T \boldsymbol{x} + b = 1.1 \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 \ .$$

Note that constructing such an $\boldsymbol{x}$ is always possible as we assumed $w_0 > w_i > 0$ for all $1 \leq i \leq d$.

We now have an input $(\boldsymbol{x}, y)$ that has non-zero support under $\mathcal{D}$. Let $\boldsymbol{r}$ be a perturbation with $\|\boldsymbol{r}\|_\infty \leq \epsilon$. We have:

$$\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{r}) + b \geq \boldsymbol{w}^T \boldsymbol{x} + b - \epsilon \cdot \|\boldsymbol{w}\|_1 = 0.1 \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 > 0 \ ,$$

so $f(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{r}) + b) = y$, i.e., $\boldsymbol{x}$ cannot be attacked by any ϵ-bounded ℓ_∞-perturbation.

Define $\hat{\boldsymbol{x}}_i$ as the input $\boldsymbol{x}$ with features x_0 and x_i switched, for some $0 \leq i < N$. Then,

$$\begin{aligned}
\boldsymbol{w}^T \hat{\boldsymbol{x}}_i + b &= \boldsymbol{w}^T \boldsymbol{x} + b - (w_0 - w_i) \cdot (x_0 - x_i) \\
&\geq \boldsymbol{w}^T \boldsymbol{x} + b - (w_0 - w_1) \cdot (x_0 - x_1) \\
&= \boldsymbol{w}^T \boldsymbol{x} + b - \epsilon \cdot \|\boldsymbol{w}\|_1 = 0.1 \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 > 0 \ .
\end{aligned}$$

Thus, the RT adversary cannot change the sign of $f(\boldsymbol{x})$ either. This means that an adversary that chooses from $S_\infty \cup S_{\mathrm{RT}}$ cannot successfully perturb $\boldsymbol{x}$.

Now, consider the affine adversary, with $\beta = 2/N$ that first applies an RT perturbation with budget $\frac{2}{N} \cdot N = 2$ (i.e., the adversary can only flip x_0 with x_1), followed by an ℓ_∞-perturbation with budget $(1 - \frac{2}{N}) \cdot \epsilon$. Specifically, the adversary flips x_0 and x_1 and then adds noise $\boldsymbol{r} = -(1 - \frac{2}{N}) \cdot \epsilon \cdot \mathrm{sign}(\boldsymbol{w})$. Let this adversarial example by $\hat{\boldsymbol{x}}_{\mathrm{affine}}$. We have

$$\begin{aligned}
\boldsymbol{w}^T \hat{\boldsymbol{x}}_{\mathrm{affine}} + b &= \boldsymbol{w}^T \boldsymbol{x} + b - (w_0 - w_1) \cdot (x_0 - x_1) - \left(1 - \frac{2}{N}\right) \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 \\
&= 1.1 \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 - \epsilon \cdot \|\boldsymbol{w}\|_1 - \left(1 - \frac{2}{N}\right) \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 \\
&= -\left(0.9 - \frac{2}{N}\right) \cdot \epsilon \cdot \|\boldsymbol{w}\|_1 \\
&< 0 \ .
\end{aligned}$$

Thus, $f(\hat{\boldsymbol{x}}_{\mathrm{affine}}) = -1 \neq y$, so the affine adversary is strictly stronger that the adversary that is restricted to RT or ℓ_∞ perturbations. □