Adversarial Examples Are Not Bugs, They Are Features

Andrew Ilyas* MIT ailyas@mit.edu
Shibani Santurkar* MIT shibani@mit.edu
Dimitris Tsipras* MIT tsipras@mit.edu
Logan Engstrom* MIT engstrom@mit.edu
Brandon Tran MIT btran115@mit.edu
Aleksander Mądry MIT madry@mit.edu

Abstract

Adversarial examples have attracted significant attention in machine learning, but the reasons for their existence and pervasiveness remain unclear. We demonstrate that adversarial examples can be directly attributed to the presence of *non-robust features*: features (derived from patterns in the data distribution) that are highly predictive, yet brittle and (thus) incomprehensible to humans. After capturing these features within a theoretical framework, we establish their widespread existence in standard datasets. Finally, we present a simple setting where we can rigorously tie the phenomena we observe in practice to a *misalignment* between the (human-specified) notion of robustness and the inherent geometry of the data.

1 Introduction

The pervasive brittleness of deep neural networks [Sze+14; Eng+19b; HD19; Ath+18] has attracted significant attention in recent years. Particularly worrisome is the phenomenon of *adversarial examples* [Big+13; Sze+14], imperceptibly perturbed natural inputs that induce erroneous predictions in state-of-the-art classifiers. Previous work has proposed a variety of explanations for this phenomenon, ranging from theoretical models [Sch+18; BPR18] to arguments based on concentration of measure in high-dimensions [Gil+18; MDM18; Sha+19a]. These theories, however, are often unable to fully capture behaviors we observe in practice (we discuss this further in Section 5).

More broadly, previous work in the field tends to view adversarial examples as aberrations arising either from the high dimensional nature of the input space or statistical fluctuations in the training data [Sze+14; GSS15; Gil+18]. From this point of view, it is natural to treat adversarial robustness as a goal that can be disentangled and pursued independently from maximizing accuracy [Mad+18; SHS19; Sug+19], either through improved standard regularization methods [TG16] or pre/post-processing of network inputs/outputs [Ues+18; CW17a; He+17].

In this work, we propose a new perspective on the phenomenon of adversarial examples. In contrast to the previous models, we cast adversarial vulnerability as a fundamental consequence of the dominant supervised learning paradigm. Specifically, we claim that:

Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.

Recall that we usually train classifiers to *solely* maximize (distributional) accuracy. Consequently, classifiers tend to use *any* available signal to do so, even those that look incomprehensible to humans. After all, the presence of "a tail" or "ears" is no more natural to a classifier than any other equally predictive feature. In fact, we find that standard ML datasets *do* admit highly predictive yet imperceptible features. We posit that

*Equal contribution

our models learn to rely on these "non-robust" features, leading to adversarial perturbations that exploit this dependence.[1]

Our hypothesis also suggests an explanation for *adversarial transferability*: the phenomenon that adversarial perturbations computed for one model often transfer to other, independently trained models. Since any two models are likely to learn similar non-robust features, perturbations that manipulate such features will apply to both. Finally, this perspective establishes adversarial vulnerability as a human-centric phenomenon, since, from the standard supervised learning point of view, non-robust features can be as important as robust ones. It also suggests that approaches aiming to enhance the interpretability of a given model by enforcing "priors" for its explanation [MV15; OMS17; Smi+17] actually hide features that are *"meaningful"* and *predictive* to standard models. As such, producing *human*-meaningful explanations that remain faithful to underlying models cannot be pursued independently from the training of the models themselves.

To corroborate our theory, we show that it is possible to disentangle robust from non-robust features in standard image classification datasets. Specifically, given any training dataset, we are able to construct:

1. **A "robustified" version for robust classification (Figure 1a)[2] .** We demonstrate that it is possible to effectively remove non-robust features from a dataset. Concretely, we create a training set (semantically similar to the original) on which *standard training* yields *good robust accuracy* on the *original, unmodified* test set. This finding establishes that adversarial vulnerability is not necessarily tied to the standard training framework, but is also a property of the dataset.

2. **A "non-robust" version for standard classification (Figure 1b)[2].** We are also able to construct a training dataset for which the inputs are nearly identical to the originals, but all appear incorrectly labeled. In fact, the inputs in the new training set are associated to their labels only through *small adversarial perturbations* (and hence utilize only non-robust features). Despite the lack of any predictive human-visible information, training on this dataset yields good accuracy on the *original, unmodified* test set. This demonstrates that adversarial perturbations can arise from flipping features in the data that are useful for classification of correct inputs (hence not being purely aberrations).

Finally, we present a concrete classification task where the connection between adversarial examples and non-robust features can be studied rigorously. This task consists of separating Gaussian distributions, and is loosely based on the model presented in Tsipras et al. [Tsi+19], while expanding upon it in a few ways. First, adversarial vulnerability in our setting can be precisely quantified as a difference between the intrinsic data geometry and that of the adversary's perturbation set. Second, robust training yields a classifier which utilizes a geometry corresponding to a combination of these two. Lastly, the gradients of standard models can be significantly more misaligned with the inter-class direction, capturing a phenomenon that has been observed in practice in more complex scenarios [Tsi+19].

2 The Robust Features Model

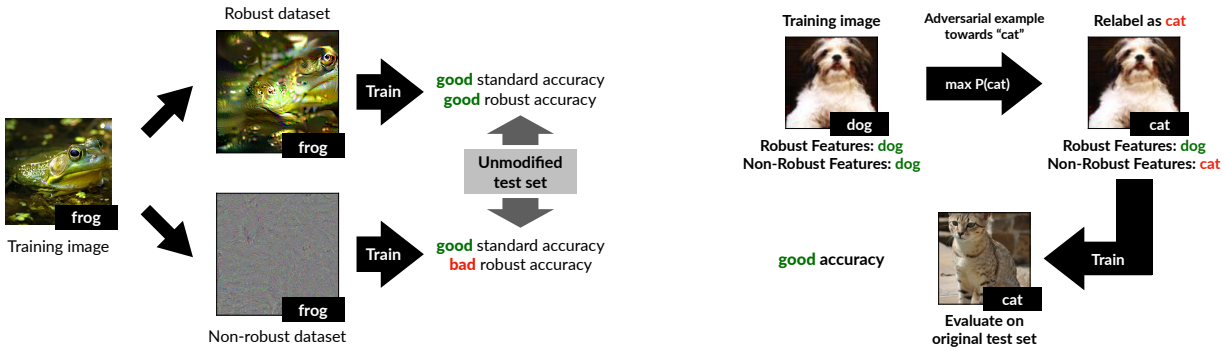
We begin by developing a framework, loosely based on the setting proposed by Tsipras et al. [Tsi+19], that enables us to rigorously refer to "robust" and "non-robust" features. In particular, we present a set of definitions which allow us to formally describe our setup, theoretical results, and empirical evidence.

Setup. We consider binary classification[3], where input-label pairs $(x, y) \in \mathcal{X} \times \{\pm 1\}$ are sampled from a (data) distribution $\mathcal{D}$; the goal is to learn a classifier $C : \mathcal{X} \to \{\pm 1\}$ which predicts a label y corresponding to a given input x.

[1] It is worth emphasizing that while our findings demonstrate that adversarial vulnerability *does* arise from non-robust features, they do not preclude the possibility of adversarial vulnerability also arising from other phenomena [TG16; Sch+18]. For example, Nakkiran [Nak19a] constructs adversarial examples that do not exploit non-robust features (and hence do not allow one to learn a generalizing model from them). Still, the mere existence of useful non-robust features suffices to establish that without explicitly discouraging models from utilizing these features, adversarial vulnerability will remain an issue.

[2] The corresponding datasets for CIFAR-10 are publicly available at `http://git.io/adv-datasets`.

[3] Our framework can be straightforwardly adapted though to the multi-class setting.



(a) (b)

Figure 1: A conceptual diagram of the experiments of Section 3. In (a) we disentangle features into combinations of robust/non-robust features (Section 3.1). In (b) we construct a dataset which appears mislabeled to humans (via adversarial examples) but results in good accuracy on the original test set (Section 3.2).

We define a *feature* to be a function mapping from the input space $\mathcal{X}$ to the real numbers, with the set of all features thus being $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$. For convenience, we assume that the features in $\mathcal{F}$ are shifted/scaled to be mean-zero and unit-variance (i.e., so that $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)] = 0$ and $\mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)^2] = 1$), in order to make the following definitions scale-invariant[4]. Note that this formal definition also captures what we abstractly think of as features (e.g., we can construct an f that captures how "furry" an image is).

Useful, robust, and non-robust features. We now define the key concepts required for formulating our framework. To this end, we categorize features in the following manner:

- **ρ-useful features**: For a given distribution $\mathcal{D}$, we call a feature f *ρ-useful* ($\rho > 0$) if it is correlated with the true label in expectation, that is if

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot f(x)] \geq \rho. \tag{1}$$

  We then define $\rho_{\mathcal{D}}(f)$ as the largest ρ for which feature f is ρ-useful under distribution $\mathcal{D}$. (Note that if a feature f is negatively correlated with the label, then $-f$ is useful instead.) Crucially, a linear classifier trained on ρ-useful features can attain non-trivial generalization performance.

- **γ-robustly useful features**: Suppose we have a ρ-useful feature f ($\rho_{\mathcal{D}}(f) > 0$). We refer to f as a *robust feature* (formally a γ-robustly useful feature for $\gamma > 0$) if, under adversarial perturbation (for some specified set of valid perturbations Δ), f remains γ-useful. Formally, if we have that

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\inf_{\delta\in\Delta(x)} y \cdot f(x+\delta)\right] \geq \gamma. \tag{2}$$

- **Useful, non-robust features**: A *useful, non-robust feature* is a feature which is ρ-useful for some ρ bounded away from zero, but is not a γ-robust feature for any $\gamma \geq 0$. These features help with classification in the standard setting, but may hinder accuracy in the adversarial setting, as the correlation with the label can be flipped.

Classification. In our framework, a classifier $C = (F, w, b)$ is comprised of a set of features $F \subseteq \mathcal{F}$, a weight vector w, and a scalar bias b. For a given input x, the classifier predicts the label y as

$$C(x) = \operatorname{sgn}\left(b + \sum_{f\in F} w_f \cdot f(x)\right).$$

For convenience, we denote the set of features learned by a classifier C as F_C.

[4]This restriction can be straightforwardly removed by simply shifting/scaling the definitions.

Standard Training. Training a classifier is performed by minimizing a loss function (via *empirical risk minimization* (ERM)) that decreases with the correlation between the weighted combination of the features and the label. The simplest example of such a loss is [5]

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}_\theta(x,y)\right] = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y\cdot\left(b+\sum_{f\in F} w_f\cdot f(x)\right)\right]. \tag{3}$$

When minimizing classification loss, *no distinction* exists between robust and non-robust features: the only distinguishing factor of a feature is its ρ-usefulness. Furthermore, the classifier will utilize *any* ρ-useful feature in F to decrease the loss of the classifier.

Robust training. In the presence of an *adversary*, any useful but non-robust features can be made *anti-correlated* with the true label, leading to adversarial vulnerability. Therefore, ERM is no longer sufficient to train classifiers that are robust, and we need to explicitly account for the effect of the adversary on the classifier. To do so, we use an *adversarial* loss function that can discern between robust and non-robust features [Mad+18]:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(x)}\mathcal{L}_\theta(x+\delta,y)\right], \tag{4}$$

for an appropriately defined set of perturbations Δ. Since the adversary can exploit non-robust features to degrade classification accuracy, minimizing this adversarial loss (as in adversarial training [GSS15; Mad+18]) can be viewed as explicitly preventing the classifier from learning a useful but non-robust combination of features.

Remark. We want to note that even though the framework above enables us to formally describe and predict the outcome of our experiments, it does not necessarily capture the notion of non-robust features exactly as we intuitively might think of them. For instance, in principle, our theoretical framework would allow for useful non-robust features to arise as combinations of useful robust features and useless non-robust features [Goh19b]. These types of constructions, however, are actually precluded by our experimental results (in particular, the classifiers trained in Section 3 would not generalize). This shows that our experimental findings capture a stronger, more fine-grained statement than our formal definitions are able to express. We view bridging this gap as an interesting direction for future work.

3 Finding Robust (and Non-Robust) Features

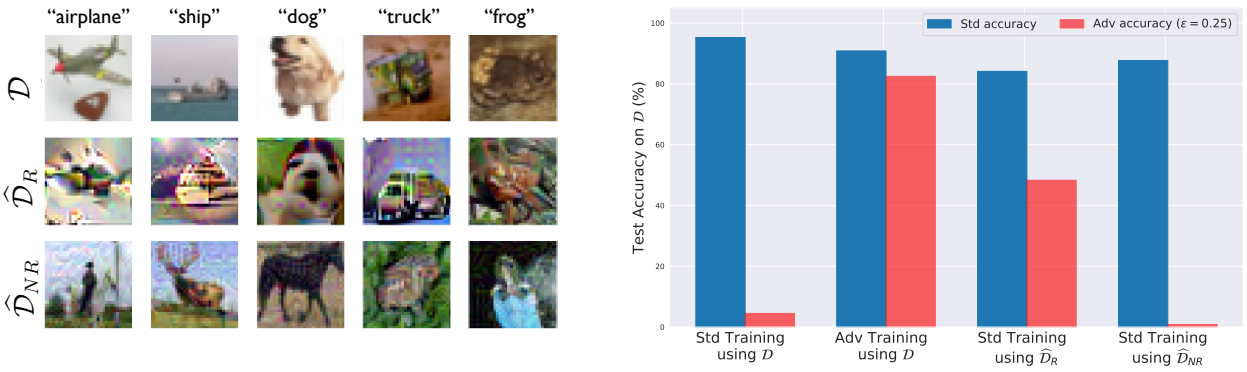
The central premise of our proposed framework is that there exist both robust and non-robust features that constitute useful signals for standard classification. We now provide evidence in support of this hypothesis by disentangling these two sets of features.

On one hand, we will construct a "robustified" dataset, consisting of samples that primarily contain robust features. Using such a dataset, we are able to train robust classifiers (with respect to the standard test set) using standard (i.e., non-robust) training. This demonstrates that robustness can arise by *removing* certain features from the dataset (as, overall, the new dataset contains less information about the original training set). Moreover, it provides evidence that adversarial vulnerability is caused by non-robust features and is not inherently tied to the standard training framework.

On the other hand, we will construct datasets where the input-label association is based purely on non-robust features (and thus the corresponding dataset appears *completely* mislabeled to humans). We show that this dataset suffices to train a classifier with good performance on the standard test set. This indicates that natural models use *non-robust features* to make predictions, even in the presence of robust features. These features *alone* are actually sufficient for non-trivial generalizations performance on natural images, which indicates that they are indeed valuable features, rather than artifacts of finite-sample overfitting.

A conceptual description of these experiments can be found in Figure 1.

[5] Just as for the other parts of this model, we use this loss for simplicity only—it is straightforward to generalize to more practical loss function such as logistic or hinge loss.



(a) (b)

Figure 2: **Left**: Random samples from our variants of the CIFAR-10 [Kri09] training set: the original training set; the *robust training set* $\widehat{\mathcal{D}}_R$, restricted to features used by a robust model; and the *non-robust training set* $\widehat{\mathcal{D}}_{NR}$, restricted to features relevant to a standard model (labels appear incorrect to humans). **Right**: Standard and robust accuracy on the CIFAR-10 test set ($\mathcal{D}$) for models trained with: (i) standard training (on $\mathcal{D}$) ; (ii) standard training on $\widehat{\mathcal{D}}_{NR}$; (iii) adversarial training (on $\mathcal{D}$); and (iv) standard training on $\widehat{\mathcal{D}}_R$. Models trained on $\widehat{\mathcal{D}}_R$ and $\widehat{\mathcal{D}}_{NR}$ reflect the original models used to create them: notably, standard training on $\widehat{\mathcal{D}}_R$ yields nontrivial robust accuracy. Results for Restricted-ImageNet [Tsi+19] are in D.8 Figure 12.

3.1 Disentangling robust and non-robust features

Recall that the features a classifier learns to rely on are based purely on how useful these features are for (standard) generalization. Thus, under our conceptual framework, if we can ensure that only robust features are useful, standard training should result in a robust classifier. Unfortunately, we cannot directly manipulate the features of very complex, high-dimensional datasets. Instead, we will leverage a robust model and modify our dataset to contain only the features that are relevant to that model.

In terms of our formal framework (Section 2), given a *robust* (i.e., adversarially trained [Mad+18]) model C we aim to construct a distribution $\widehat{\mathcal{D}}_R$ which satisfies:

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_R}[f(x)\cdot y] = \begin{cases} \mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)\cdot y] & \text{if } f \in F_C \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where F_C again represents the set of features utilized by C. Conceptually, we want features used by C to be as useful as they were on the original distribution $\mathcal{D}$ while ensuring that the rest of the features are not useful under $\widehat{\mathcal{D}}_{NR}$.

We will construct a training set for $\widehat{\mathcal{D}}_R$ via a one-to-one mapping $x \mapsto x_r$ from the original training set for $\mathcal{D}$. In the case of a deep neural network, F_C corresponds to exactly the set of activations in the penultimate layer (since these correspond to inputs to a linear classifier). To ensure that features used by the model are equally useful under both training sets, we (approximately) enforce all features in F_C to have similar values for both x and x_r through the following optimization:

$$\min_{x_r} \|g(x_r) - g(x)\|_2, \tag{6}$$

where x is the original input and g is the mapping from x to the representation layer. We optimize this objective using gradient descent in input space[6].

Since we don't have access to features outside F_C, there is no way to ensure that the expectation in (5) is zero for all $f \notin F_C$. To approximate this condition, we choose the starting point of gradient descent for the optimization in (6) to be an input x_0 which is drawn from $\mathcal{D}$ independently of the label of x (we also explore sampling x_0 from noise in Appendix D.1). This choice ensures that any feature present in that input will

[6]We follow [Mad+18] and normalize gradient steps during this optimization. Experimental details are provided in Appendix C.

not be useful since they are not correlated with the label in expectation over x_0. The underlying assumption here is that, when performing the optimization in (6), features that are not being directly optimized (i.e., features outside F_C) are not affected. We provide pseudocode for the construction in Figure 5 (Appendix C).

Given the new training set for $\widehat{\mathcal{D}}_R$ (a few random samples are visualized in Figure 2a), we train a classifier using standard (non-robust) training. We then test this classifier on the original test set (i.e. $\mathcal{D}$). The results (Figure 2b) indicate that the classifier learned using the new dataset attains good accuracy in *both standard and adversarial settings* [7] [8].

As a control, we repeat this methodology using a standard (non-robust) model for C in our construction of the dataset. Sample images from the resulting "non-robust dataset" $\widehat{\mathcal{D}}_{NR}$ are shown in Figure 2a—they tend to resemble more the source image of the optimization x_0 than the target image x. We find that training on this dataset leads to good standard accuracy, yet yields almost no robustness (Figure 2b). We also verify that this procedure is not simply a matter of encoding the weights of the original model—we get the same results for both $\widehat{\mathcal{D}}_R$ and $\widehat{\mathcal{D}}_{NR}$ if we train with different architectures than that of the original models.

Overall, our findings corroborate the hypothesis that adversarial examples can arise from (non-robust) features of the data itself. By filtering out non-robust features from the dataset (e.g. by restricting the set of available features to those used by a robust model), one can train a significantly more robust model using *standard training*.

3.2 Non-robust features suffice for standard classification

The results of the previous section show that by restricting the dataset to only contain features that are used by a robust model, standard training results in classifiers that are significantly more robust. This suggests that when training on the standard dataset, non-robust features take on a large role in the resulting learned classifier. Here we set out to show that this role is not merely incidental or due to finite-sample overfitting. In particular, we demonstrate that non-robust features *alone* suffice for standard generalization— i.e., a model trained solely on non-robust features can perform well on the *standard* test set.

To show this, we construct a dataset where the only features that are useful for classification are *non-robust* features (or in terms of our formal model from Section 2, all features f that are ρ-useful are non-robust). To accomplish this, we modify each input-label pair (x, y) as follows. We select a target class t either (a) uniformly at random among classes (hence features become uncorrelated with the labels) or (b) deterministically according to the source class (e.g. using a fixed permutation of labels). Then, we add a small adversarial perturbation to x in order to ensure it is classified as t by a standard model. Formally:

$$x_{adv} = \underset{\|x'-x\| \leq \varepsilon}{\arg\min}\ L_C(x', t), \tag{7}$$

where L_C is the loss under a standard (non-robust) classifier C and ε is a small constant. The resulting inputs are nearly indistinguishable from the originals (Appendix D Figure 9)—to a human observer, it thus appears that the label t assigned to the modified input is simply incorrect. The resulting input-label pairs (x_{adv}, t) make up the new training set (pseudocode in Appendix C Figure 6).

Now, since $\|x_{adv} - x\|$ is small, by definition the robust features of x_{adv} are still correlated with class y (and not t) in expectation over the dataset. After all, humans still recognize the original class. On the other hand, since every x_{adv} is strongly classified as t by a standard classifier, it must be that some of the non-robust features are now strongly correlated with t (in expectation).

In the case where t is chosen at random, the robust features are originally uncorrelated with the label t (in expectation), and after the adversarial perturbation can be only slightly correlated (hence being significantly

[7] In an attempt to explain the gap in accuracy between the model trained on $\widehat{\mathcal{D}}_R$ and the original robust classifier C, we test distributional shift, by reporting results on the "robustified" test set in Appendix D.3.

[8] In order to gain more confidence in the robustness of the resulting model, we attempt several diverse attacks in Appendix D.2.

less useful for classification than before) [9]. Formally, we aim to construct a dataset $\widehat{\mathcal{D}}_{rand}$ where [10] :

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_{rand}}\left[y \cdot f(x)\right] \begin{cases} > 0 & \text{if } f \text{ non-robustly useful under } \mathcal{D}, \\ \simeq 0 & \text{otherwise.} \end{cases} \tag{8}$$

In contrast, when t is chosen deterministically based on y, the robust features actually point *away* from the assigned label t. In particular, all of the inputs labeled with class t exhibit *non-robust features* correlated with t, but robust features correlated with the original class y. Thus, robust features on the original training set provide significant predictive power on the training set, but will actually hurt generalization on the standard test set. Viewing this case again using the formal model, our goal is to construct $\widehat{\mathcal{D}}_{det}$ such that

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_{det}}\left[y \cdot f(x)\right] \begin{cases} > 0 & \text{if } f \text{ non-robustly useful under } \mathcal{D}, \\ < 0 & \text{if } f \text{ robustly useful under } \mathcal{D} \\ \in \mathbb{R} & \text{otherwise } (f \text{ not useful under } \mathcal{D})^{11} \end{cases} \tag{9}$$

We find that standard training on these datasets actually generalizes to the *original* test set, as shown in Table 1). This indicates that non-robust features are indeed useful for classification in the standard setting. Remarkably, even training on $\widehat{\mathcal{D}}_{det}$ (where all the robust features are correlated with the wrong class), results in a well-generalizing classifier. This indicates that non-robust features can be picked up by models during standard training, even in the presence of *robust features* that are predictive [12][13].

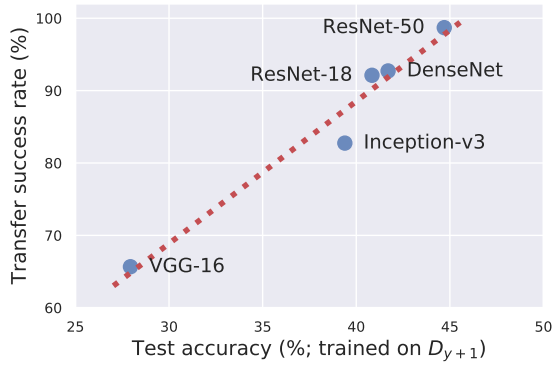


Figure 3: Transfer rate of adversarial examples from a ResNet-50 to different architectures alongside test set performance of these architecture when trained on the dataset generated in Section 3.2. Architectures more susceptible to transfer attacks also performed better on the standard test set supporting our hypothesis that adversarial transferability arises from utilizing similar *non-robust features*.

Source Dataset	**Dataset**	
	CIFAR-10	ImageNet$_R$
$\mathcal{D}$	95.3%	96.6%
$\widehat{\mathcal{D}}_{rand}$	63.3%	87.9%
$\widehat{\mathcal{D}}_{det}$	43.7%	64.4%

Table 1: Test accuracy (on $\mathcal{D}$) of classifiers trained on the $\mathcal{D}$, $\widehat{\mathcal{D}}_{rand}$, and $\widehat{\mathcal{D}}_{det}$ training sets created using a standard (non-robust) model. For both $\widehat{\mathcal{D}}_{rand}$ and $\widehat{\mathcal{D}}_{det}$, only non-robust features correspond to useful features on both the train set and $\mathcal{D}$. These datasets are constructed using adversarial perturbations of x towards a class t (random for $\widehat{\mathcal{D}}_{rand}$ and deterministic for $\widehat{\mathcal{D}}_{det}$); the resulting images are relabeled as t.

3.3 Transferability can arise from non-robust features

One of the most intriguing properties of adversarial examples is that they *transfer* across models with different architectures and independently sampled training sets [Sze+14; PMG16; CRP19]. Here, we show

[9] Goh [Goh19a] provides an approach to quantifying this "robust feature leakage" and finds that one can obtain a (small) amount of test accuracy by leveraging robust feature leakage on $\widehat{\mathcal{D}}_{rand}$.

[10] Note that the optimization procedure we describe aims to merely *approximate* this condition, where we once again use trained models to simulate access to robust and non-robust features.

[11] Note that regardless how useful a feature is on $\widehat{\mathcal{D}}_{det}$, since it is useless on $\mathcal{D}$ it cannot provide any generalization benefit on the unaltered test set.

[12] Additional results and analysis (e.g. training curves, generating $\widehat{\mathcal{D}}_{rand}$ and $\widehat{\mathcal{D}}_{det}$ with a robust model, etc.) are in App. D.6 and D.5

[13] We also show that the models trained on $\widehat{\mathcal{D}}_{rand}$ and $\widehat{\mathcal{D}}_{det}$ generalize to CIFAR-10.1 [Rec+19] in Appendix D.7.

that this phenomenon can in fact be viewed as a natural consequence of the existence of non-robust features. Recall that, according to our main thesis, adversarial examples can arise as a result of perturbing well-generalizing, yet brittle features. Given that such features are inherent to the data distribution, different classifiers trained on independent samples from that distribution are likely to utilize similar non-robust features. Consequently, an adversarial example constructed by exploiting the non-robust features learned by one classifier will transfer to any other classifier utilizing these features in a similar manner.

In order to illustrate and corroborate this hypothesis, we train five different architectures on the dataset generated in Section 3.2 (adversarial examples with deterministic labels) for a standard ResNet-50 [He+16]. Our hypothesis would suggest that architectures which learn better from this training set (in terms of performance on the standard test set) are more likely to learn similar non-robust features to the original classifier. Indeed, we find that the test accuracy of each architecture is predictive of how often adversarial examples transfer from the original model to standard classifiers with that architecture (Figure 3). In a similar vein, Nakkiran [Nak19a] constructs a set of adversarial perturbations that is explicitly non-transferable and finds that these perturbations cannot be used to learn a good classifier. These findings thus corroborate our hypothesis that adversarial transferability arises when models learn similar brittle features of the underlying dataset.

4 A Theoretical Framework for Studying (Non)-Robust Features

The experiments from the previous section demonstrate that the conceptual framework of robust and non-robust features is strongly predictive of the empirical behavior of state-of-the-art models on real-world datasets. In order to further strengthen our understanding of the phenomenon, we instantiate the framework in a concrete setting that allows us to theoretically study various properties of the corresponding model. Our model is similar to that of Tsipras et al. [Tsi+19] in the sense that it contains a dichotomy between robust and non-robust features, but extends upon it in a number of ways:

1. The adversarial vulnerability can be explicitly expressed as a difference between the inherent data metric and the ℓ_2 metric.
2. Robust learning corresponds exactly to learning a combination of these two metrics.
3. The gradients of adversarially trained models align better with the adversary's metric.

Setup. We study a simple problem of *maximum likelihood classification* between two Gaussian distributions. In particular, given samples (x, y) sampled from $\mathcal{D}$ according to

$$y \overset{\text{u.a.r.}}{\sim} \{-1, +1\}, \qquad x \sim \mathcal{N}(y \cdot \boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*), \tag{10}$$

our goal is to learn parameters $\Theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that

$$\Theta = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell(x; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma})\right], \tag{11}$$

where $\ell(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the Gaussian negative log-likelihood (NLL) function. Intuitively, we find the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ which maximize the likelihood of the sampled data under the given model. Classification under this model can be accomplished via likelihood test: given an unlabeled sample x, we predict y as

$$y = \arg\max_{y} \ell(x; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \text{sign}\left(x^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}\right).$$

In turn, the *robust analogue* of this problem arises from replacing $\ell(x; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the NLL under adversarial perturbation. The resulting robust parameters Θ_r can be written as

$$\Theta_r = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|_2 \leq \varepsilon} \ell(x + \delta; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma})\right], \tag{12}$$

A detailed analysis of this setting is in Appendix E—here we present a high-level overview of the results.

(1) Vulnerability from metric misalignment (non-robust features). Note that in this model, one can rigorously make reference to an *inner product* (and thus a metric) induced by the features. In particular, one can view the learned parameters of a Gaussian $\Theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as defining an inner product over the input space given by $\langle x, y \rangle_\Theta = (x - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (y - \boldsymbol{\mu})$. This in turn induces the Mahalanobis distance, which represents how a change in the input affects the features learned by the classifier. This metric is not necessarily aligned with the metric in which the adversary is constrained, the ℓ_2-norm. Actually, we show that adversarial vulnerability arises exactly as a *misalignment* of these two metrics.

Theorem 1 (Adversarial vulnerability from misalignment)**.** *Consider an adversary whose perturbation is determined by the "Lagrangian penalty" form of* (12)*, i.e.*

$$\max_\delta \ell(x + \delta; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma}) - C \cdot \|\delta\|_2,$$

where $C \geq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma}_*)}$ is a constant trading off NLL minimization and the adversarial constraint[14]*. Then, the adversarial loss $\mathcal{L}_{adv}$ incurred by the non-robustly learned $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by:*

$$\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) = tr\left[\left(I + (C \cdot \boldsymbol{\Sigma}_* - I)^{-1}\right)^2\right] - d,$$

and, for a fixed $tr(\boldsymbol{\Sigma}_*) = k$ the above is minimized by $\boldsymbol{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$.

In fact, note that such a misalignment corresponds precisely to the existence of *non-robust features*, as it indicates that "small" changes in the adversary's metric along certain directions can cause large changes under the data-dependent notion of distance established by the parameters. This is illustrated in Figure 4, where misalignment in the feature-induced metric is responsible for the presence of a non-robust feature in the corresponding classification problem.

(2) Robust Learning. The optimal (non-robust) maximum likelihood estimate is $\Theta = \Theta^*$, and thus the vulnerability for the standard MLE estimate is governed entirely by the true data distribution. The following theorem characterizes the behaviour of the learned parameters in the robust problem. [15]. In fact, we can prove (Section E.3.4) that performing (sub)gradient descent on the inner maximization (also known as *adversarial training* [GSS15; Mad+18]) yields exactly Θ_r. We find that as the perturbation budget ε is increased, the metric induced by the learned features *mixes* ℓ_2 and the metric induced by the features.

Theorem 2 (Robustly Learned Parameters)**.** *Just as in the non-robust case, $\boldsymbol{\mu}_r = \boldsymbol{\mu}^*$, i.e. the true mean is learned. For the robust covariance $\boldsymbol{\Sigma}_r$, there exists an $\varepsilon_0 > 0$, such that for any $\varepsilon \in [0, \varepsilon_0)$,*

$$\boldsymbol{\Sigma}_r = \frac{1}{2}\boldsymbol{\Sigma}_* + \frac{1}{\lambda} \cdot \boldsymbol{I} + \sqrt{\frac{1}{\lambda} \cdot \boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}, \qquad \textit{where} \qquad \Omega\left(\frac{1 + \varepsilon^{1/2}}{\varepsilon^{1/2} + \varepsilon^{3/2}}\right) \leq \lambda \leq O\left(\frac{1 + \varepsilon^{1/2}}{\varepsilon^{1/2}}\right).$$

The effect of robust optimization under an ℓ_2-constrained adversary is visualized in Figure 4. As ϵ grows, the learned covariance becomes more aligned with identity. For instance, we can see that the classifier learns to be less sensitive in certain directions, despite their usefulness for natural classification.

(3) Gradient Interpretability. Tsipras et al. [Tsi+19] observe that gradients of robust models tend to look more semantically meaningful. It turns out that under our model, this behaviour arises as a natural consequence of Theorem 2. In particular, we show that the resulting robustly learned parameters cause the gradient of the linear classifier and the vector connecting the means of the two distributions to better align (in a worst-case sense) under the ℓ_2 inner product.

Theorem 3 (Gradient alignment)**.** *Let $f(x)$ and $f_r(x)$ be monotonic classifiers based on the linear separator induced by standard and ℓ_2-robust maximum likelihood classification, respectively. The maximum angle formed between the gradient of the classifier (wrt input) and the vector connecting the classes can be smaller for the robust model:*

$$\min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f_r(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f_r(x)\|} > \min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f(x)\|}.$$

[14]The constraint on C is to ensure the problem is concave.

[15]Note: as discussed in Appendix E.3.3, we study a slight relaxation of (12) that approaches exactness exponentially fast as $d \to \infty$

Figure 4 illustrates this phenomenon in the two-dimensional case. With ℓ_2-bounded adversarial training the gradient direction (perpendicular to the decision boundary) becomes increasingly aligned under the ℓ_2 inner product with the vector between the means ($\boldsymbol{\mu}$).

Discussion. Our theoretical analysis suggests that rather than offering any quantitative classification benefits, a natural way to view the role of robust optimization is as enforcing a *prior* over the features learned by the classifier. In particular, training with an ℓ_2-bounded adversary prevents the classifier from relying heavily on features which induce a metric dissimilar to the ℓ_2 metric. The strength of the adversary then allows for a trade-off between the enforced prior, and the data-dependent features.

Robustness and accuracy. Note that in the setting described so far, robustness *can* be at odds with accuracy since robust training prevents us from learning the most accurate classifier (a similar conclusion is drawn in [Tsi+19]). However, we note that there are very similar settings where non-robust features manifest themselves in the same way, yet a classifier with perfect robustness and accuracy is still attainable. Concretely, consider the distributions pictured in Figure 14 in Appendix D.10. It is straightforward to show that while there are many perfectly accurate classifiers, any standard loss function will learn an accurate yet non-robust classifier. Only when robust training is employed does the classifier learn a perfectly accurate and perfectly robust decision boundary.

5 Related Work

Several models for explaining adversarial examples have been proposed in prior work, utilizing ideas ranging from finite-sample overfitting to high-dimensional statistical phenomena [Gil+18; FFF18; For+19; TG16; Sha+19a; MDM18; Sha+19b; GSS15; BPR18]. The key differentiating aspect of our model is that adversarial perturbations arise as *well-generalizing, yet brittle, features*, rather than statistical anomalies or effects of poor statistical concentration. In particular, adversarial vulnerability does not stem from using a specific model class or a specific training method, since standard training on the "robustified" data distribution of Section 3.1 leads to robust models. At the same time, as shown in Section 3.2, these non-robust features are sufficient to learn a good standard classifier. We discuss the connection between our model and others in detail in Appendix A. We discuss additional related work in Appendix B.

6 Conclusion

In this work, we cast the phenomenon of adversarial examples as a natural consequence of the presence of *highly predictive but non-robust features* in standard ML datasets. We provide support for this hypothesis by

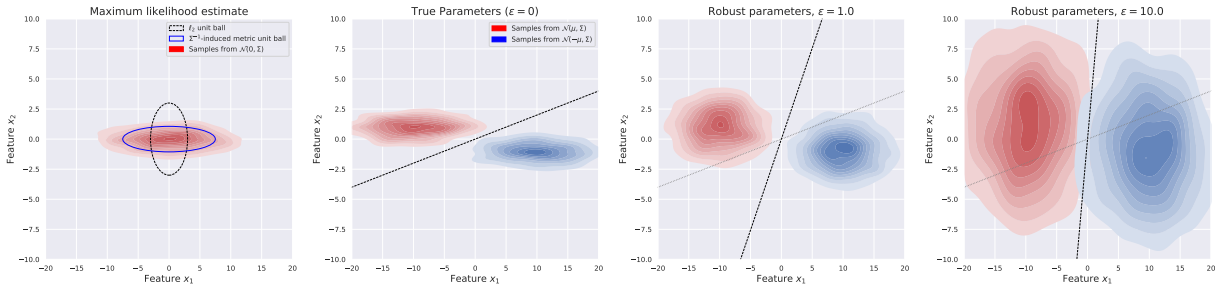


Figure 4: An empirical demonstration of the effect illustrated by Theorem 2—as the adversarial perturbation budget ε is increased, the learned mean $\boldsymbol{\mu}$ remains constant, but the learned covariance "blends" with the identity matrix, effectively adding more and more uncertainty onto the non-robust feature.

explicitly disentangling robust and non-robust features in standard datasets, as well as showing that non-robust features alone are sufficient for good generalization. Finally, we study these phenomena in more detail in a theoretical setting where we can rigorously study adversarial vulnerability, robust training, and gradient alignment.

Our findings prompt us to view adversarial examples as a fundamentally *human* phenomenon. In particular, we should not be surprised that classifiers exploit highly predictive features that happen to be non-robust under a human-selected notion of similarity, given such features exist in real-world datasets. In the same manner, from the perspective of interpretability, as long as models rely on these non-robust features, we cannot expect to have model explanations that are both human-meaningful and faithful to the models themselves. Overall, attaining models that are robust and interpretable will require explicitly encoding *human priors* into the training process.

7 Acknowledgements

We thank Preetum Nakkiran for suggesting the experiment of Appendix D.9 (i.e. replicating Figure 3 but with targeted attacks). We also are grateful to the authors of Engstrom et al. [Eng+19a] (Chris Olah, Dan Hendrycks, Justin Gilmer, Reiichiro Nakano, Preetum Nakkiran, Gabriel Goh, Eric Wallace)—for their insights and efforts replicating, extending, and discussing our experimental results.

Work supported in part by the NSF grants CCF-1553428, CCF-1563880, CNS-1413920, CNS-1815221, IIS-1447786, IIS-1607189, the Microsoft Corporation, the Intel Corporation, the MIT-IBM Watson AI Lab research grant, and an Analog Devices Fellowship.

References

[ACW18] Anish Athalye, Nicholas Carlini, and David A. Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *International Conference on Machine Learning (ICML)*. 2018.

[Ath+18] Anish Athalye et al. "Synthesizing Robust Adversarial Examples". In: *International Conference on Machine Learning (ICML)*. 2018.

[BCN06] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *International Conference on Knowledge Discovery and Data Mining (KDD)*. 2006.

[Big+13] Battista Biggio et al. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases (ECML-KDD)*. 2013.

[BPR18] Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. "Adversarial examples from computational constraints". In: *arXiv preprint arXiv:1805.10204*. 2018.

[Car+19] Nicholas Carlini et al. "On Evaluating Adversarial Robustness". In: *ArXiv preprint arXiv:1902.06705*. 2019.

[CRK19] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. "Certified adversarial robustness via randomized smoothing". In: *arXiv preprint arXiv:1902.02918*. 2019.

[CRP19] Zachary Charles, Harrison Rosenberg, and Dimitris Papailiopoulos. "A Geometric Perspective on the Transferability of Adversarial Directions". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2019.

[CW17a] Nicholas Carlini and David Wagner. "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods". In: *Workshop on Artificial Intelligence and Security (AISec)*. 2017.

[CW17b] Nicholas Carlini and David Wagner. "Towards evaluating the robustness of neural networks". In: *Symposium on Security and Privacy (SP)*. 2017.

[Dan67] John M. Danskin. *The Theory of Max-Min and its Application to Weapons Allocation Problems*. 1967.

[Das+19] Constantinos Daskalakis et al. "Efficient Statistics, in High Dimensions, from Truncated Samples". In: *Foundations of Computer Science (FOCS)*. 2019.

[Din+19] Gavin Weiguang Ding et al. "On the Sensitivity of Adversarial Robustness to Input Data Distributions". In: *International Conference on Learning Representations*. 2019.

[Eng+19a] Logan Engstrom et al. "A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features'". In: *Distill* (2019). https://distill.pub/2019/advex-bugs-discussion. DOI: 10.23915/distill.00019.

[Eng+19b] Logan Engstrom et al. "A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations". In: *International Conference on Machine Learning (ICML)*. 2019.

[FFF18] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. "Adversarial vulnerability for any classifier". In: *Advances in Neural Information Processing Systems (NeuRIPS)*. 2018.

[FMF16] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. "Robustness of classifiers: from adversarial to random noise". In: *Advances in Neural Information Processing Systems*. 2016.

[For+19] Nic Ford et al. "Adversarial Examples Are a Natural Consequence of Test Error in Noise". In: *arXiv preprint arXiv:1901.10513*. 2019.

[Fur+18] Tommaso Furlanello et al. "Born-Again Neural Networks". In: *International Conference on Machine Learning (ICML)*. 2018.

[Gei+19] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." In: *International Conference on Learning Representations*. 2019.

[Gil+18] Justin Gilmer et al. "Adversarial spheres". In: *Workshop of International Conference on Learning Representations (ICLR)*. 2018.

[Goh19a] Gabriel Goh. "A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Robust Feature Leakage". In: *Distill* (2019). https://distill.pub/2019/advex-bugs-discussion/response-2. DOI: 10.23915/distill.00019.2.

[Goh19b] Gabriel Goh. "A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Two Examples of Useful, Non-Robust Features". In: *Distill* (2019). https://distill.pub/2019/advex-bugs-discussion/response-3. DOI: 10.23915/distill.00019.3.

[GSS15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations (ICLR)*. 2015.

[HD19] Dan Hendrycks and Thomas G. Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations". In: *International Conference on Learning Representations (ICLR)*. 2019.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[He+17] Warren He et al. "Adversarial example defense: Ensembles of weak defenses are not strong". In: *USENIX Workshop on Offensive Technologies (WOOT)*. 2017.

[HVD14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network". In: *Neural Information Processing Systems (NeurIPS) Deep Learning Workshop*. 2014.

[JLT18] Saumya Jetley, Nicholas Lord, and Philip Torr. "With friends like these, who needs adversaries?" In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[Kri09] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: *Technical report*. 2009.

[KSJ19] Beomsu Kim, Junghoon Seo, and Taegyun Jeon. "Bridging Adversarial Robustness and Gradient Interpretability". In: *International Conference on Learning Representations Workshop on Safe Machine Learning (ICLR SafeML)*. 2019.

[Lec+19] Mathias Lecuyer et al. "Certified robustness to adversarial examples with differential privacy". In: *Symposium on Security and Privacy (SP)*. 2019.

[Liu+17] Yanpei Liu et al. "Delving into Transferable Adversarial Examples and Black-box Attacks". In: *International Conference on Learning Representations (ICLR)*. 2017.

[LM00] Beatrice Laurent and Pascal Massart. "Adaptive estimation of a quadratic functional by model selection". In: *Annals of Statistics*. 2000.

[Mad+18] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *International Conference on Learning Representations (ICLR)*. 2018.

[MDM18] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. "The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure". In: *AAAI Conference on Artificial Intelligence (AAAI)*. 2018.

[Moo+17] Seyed-Mohsen Moosavi-Dezfooli et al. "Universal adversarial perturbations". In: *conference on computer vision and pattern recognition (CVPR)*. 2017.

[MV15] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them". In: *computer vision and pattern recognition (CVPR)*. 2015.

[Nak19a] Preetum Nakkiran. "A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarial Examples are Just Bugs, Too". In: *Distill* (2019). https://distill.pub/2019/advex-bugs-discussion/response-5. DOI: 10.23915/distill.00019.5.

[Nak19b] Preetum Nakkiran. "Adversarial robustness may be at odds with simplicity". In: *arXiv preprint arXiv:1901.00532*. 2019.

[OMS17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature Visualization". In: *Distill*. 2017.

[Pap+17] Nicolas Papernot et al. "Practical black-box attacks against machine learning". In: *Asia Conference on Computer and Communications Security*. 2017.

[PMG16] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. "Transferability in Machine Learning: from Phenomena to Black-box Attacks using Adversarial Samples". In: *ArXiv preprint arXiv:1605.07277*. 2016.

[Rec+19] Benjamin Recht et al. "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" In: *International Conference on Machine Learning (ICML)*. 2019.

[RSL18] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. "Certified defenses against adversarial examples". In: *International Conference on Learning Representations (ICLR)*. 2018.

[Rus+15] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)*. 2015.

[Sch+18] Ludwig Schmidt et al. "Adversarially Robust Generalization Requires More Data". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[Sha+19a] Ali Shafahi et al. "Are adversarial examples inevitable?" In: *International Conference on Learning Representations (ICLR)*. 2019.

[Sha+19b] Adi Shamir et al. "A Simple Explanation for the Existence of Adversarial Examples with Small Hamming Distance". In: *arXiv preprint arXiv:1901.10861*. 2019.

[SHS19] David Stutz, Matthias Hein, and Bernt Schiele. "Disentangling Adversarial Robustness and Generalization". In: *Computer Vision and Pattern Recognition (CVPR)*. 2019.

[Smi+17] D. Smilkov et al. "SmoothGrad: removing noise by adding noise". In: *ICML workshop on visualization for deep learning*. 2017.

[Sug+19] Arun Sai Suggala et al. "Revisiting Adversarial Risk". In: *Conference on Artificial Intelligence and Statistics (AISTATS)*. 2019.

[Sze+14] Christian Szegedy et al. "Intriguing properties of neural networks". In: *International Conference on Learning Representations (ICLR)*. 2014.

[TG16] Thomas Tanay and Lewis Griffin. "A Boundary Tilting Perspective on the Phenomenon of Adversarial Examples". In: *ArXiv preprint arXiv:1608.07690*. 2016.

[Tra+17] Florian Tramer et al. “The Space of Transferable Adversarial Examples”. In: *ArXiv preprint arXiv:1704.03453*. 2017.

[Tsi+19] Dimitris Tsipras et al. “Robustness May Be at Odds with Accuracy”. In: *International Conference on Learning Representations (ICLR)*. 2019.

[Ues+18] Jonathan Uesato et al. “Adversarial Risk and the Dangers of Evaluating Against Weak Attacks”. In: *International Conference on Machine Learning (ICML)*. 2018.

[Wan+18] Tongzhou Wang et al. “Dataset Distillation”. In: *ArXiv preprint arXiv:1811.10959*. 2018.

[WK18] Eric Wong and J Zico Kolter. “Provable defenses against adversarial examples via the convex outer adversarial polytope”. In: *International Conference on Machine Learning (ICML)*. 2018.

[Xia+19] Kai Y. Xiao et al. “Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability”. In: *International Conference on Learning Representations (ICLR)*. 2019.

[Zou+18] Haosheng Zou et al. “Geometric Universality of Adversarial Examples in Deep Learning”. In: *Geometry in Machine Learning ICML Workshop (GIML)*. 2018.

A Connections to and Disambiguation from Other Models

Here, we describe other models for adversarial examples and how they relate to the model presented in this paper.

Concentration of measure in high-dimensions. An orthogonal line of work [Gil+18; FFF18; MDM18; Sha+19a], argues that the high dimensionality of the input space can present fundamental barriers on classifier robustness. At a high level, one can show that, for certain data distributions, any decision boundary will be close to a large fraction of inputs and hence no classifier can be robust against small perturbations. While there might exist such fundamental barriers to robustly classifying standard datasets, this model cannot fully explain the situation observed in practice, where one can train (reasonably) robust classifiers on standard datasets [Mad+18; RSL18; WK18; Xia+19; CRK19].

Insufficient data. Schmidt et al. [Sch+18] propose a theoretical model under which a single sample is sufficient to learn a good, yet non-robust classifier, whereas learning a good robust classifier requires $O(\sqrt{d})$ samples. Under this model, adversarial examples arise due to insufficient information about the true data distribution. However, unless the adversary is strong enough (in which case no robust classifier exists), adversarial inputs cannot be utilized as inputs of the opposite class (as done in our experiments in Section 3.2). We note that our model does not explicitly contradict the main thesis of Schmidt et al. [Sch+18]. In fact, this thesis can be viewed as a natural consequence of our conceptual framework. In particular, since training models robustly reduces the effective amount of information in the training data (as non-robust features are discarded), more samples should be required to generalize robustly.

Boundary Tilting. Tanay and Griffin [TG16] introduce the "boundary tilting" model for adversarial examples, and suggest that adversarial examples are a product of over-fitting. In particular, the model conjectures that "adversarial examples are possible because the class boundary extends beyond the submanifold of sample data and can be—under certain circumstances—lying close to it." Consequently, the authors suggest that mitigating adversarial examples may be a matter of regularization and preventing finite-sample overfitting. In contrast, our empirical results in Section 3.2 suggest that adversarial inputs consist of features inherent to the data distribution, since they can encode generalizing information about the target class.

Inspired by this hypothesis and concurrently to our work, Kim, Seo, and Jeon [KSJ19] present a simple classification task comprised of two Gaussian distributions in two dimensions. They experimentally show that the decision boundary tends to better align with the vector between the two means for robust models. This is a special case of our theoretical results in Section 4. (Note that this exact statement is not true beyond two dimensions, as discussed in Section 4.)

Test Error in Noise. Fawzi, Moosavi-Dezfooli, and Frossard [FMF16] and Ford et al. [For+19] argue that the adversarial robustness of a classifier can be directly connected to its robustness under (appropriately scaled) random noise. While this constitutes a natural explanation of adversarial vulnerability given the classifier robustness to noise, these works do not attempt to justify the source of the latter.

At the same time, recent work [Lec+19; CRK19; For+19] utilizes random noise during training or testing to construct adversarially robust classifiers. In the context of our framework, we can expect the added noise to disproportionately affect non-robust features and thus hinder the model's reliance on them.

Local Linearity. Goodfellow, Shlens, and Szegedy [GSS15] suggest that the local linearity of DNNs is largely responsible for the existence of small adversarial perturbations. While this conjecture is supported by the effectiveness of adversarial attacks exploiting local linearity (e.g., FGSM [GSS15]), it is not sufficient to fully characterize the phenomena observed in practice. In particular, there exist adversarial examples that violate the local linearity of the classifier [Mad+18], while classifiers that are less linear do not exhibit greater robustness [ACW18].

Piecewise-linear decision boundaries. Shamir et al. [Sha+19b] prove that the geometric structure of the classifier's decision boundaries can lead to sparse adversarial perturbations. However, this result does not take into account the distance to the decision boundary along these direction or feasibility constraints on the input domain. As a result, it cannot meaningfully distinguish between classifiers that are brittle to small adversarial perturbations and classifiers that are moderately robust.

Theoretical constructions which incidentally exploit non-robust features. Bubeck, Price, and Razenshteyn [BPR18] and Nakkiran [Nak19b] propose theoretical models where the barrier to learning robust classifiers is, respectively, due to computational constraints or model complexity. In order to construct distributions that admit accurate yet non-robust classifiers they (implicitly) utilize the concept of non-robust features. Namely, they add a low-magnitude signal to each input that encodes the true label. This allows a classifier to achieve perfect standard accuracy, but cannot be utilized in an adversarial setting as this signal is susceptible to small adversarial perturbations.

B Additional Related Work

We describe previously proposed models for the existence of adversarial examples in the previous section. Here we discuss other work that is methodologically or conceptually similar to ours.

Distillation. The experiments performed in Section 3.1 can be seen as a form of *distillation*. There is a line of work, known as model distillation [HVD14; Fur+18; BCN06], where the goal is to train a new model to mimic another already trained model. This is typically achieved by adding some regularization terms to the loss in order to encourage the two models to be similar, often replacing training labels with some other target based on the already trained model. While it might be possible to successfully distill a robust model using these methods, our goal was to achieve it by *only* modifying the training set (leaving the training process unchanged), hence demonstrating that adversarial vulnerability is mainly a property of the dataset. Closer to our work is dataset distillation [Wan+18] which considers the problem of reconstructing a classifier from an alternate dataset much smaller than the original training set. This method aims to produce inputs that directly encode the weights of the already trained model by ensuring that the classifier's gradient with respect to these inputs approximates the desired weights. (As a result, the inputs constructed do not resemble natural inputs.) This approach is orthogonal to our goal since we are not interested in encoding the particular weights into the dataset but rather in imposing a structure to its features.

Adversarial Transferabiliy. In our work, we posit that a potentially natural consequence of the existence of non-robust features is *adversarial transferability* [Pap+17; Liu+17; PMG16]. A recent line of work has considered this phenomenon from a theoretical perspective, confined to simple models, or unbounded perturbations [CRP19; Zou+18]. Tramer et al. [Tra+17] study transferability empirically, by finding *adversarial subspaces*, (orthogonal vectors whose linear combinations are adversarial perturbations). The authors find that there is a significant overlap in the adversarial subspaces between different models, and identify this as a source of transferability. In our work, we provide a potential reason for this overlap—these directions correspond to non-robust features utilized by models in a similar manner.

Universal Adversarial Perturbations Moosavi-Dezfooli et al. [Moo+17] construct perturbations that can cause misclassification when applied to multiple different inputs. More recently, Jetley, Lord, and Torr [JLT18] discover input patterns that are meaningless to humans and can induce misclassification, while at the same time being essential for standard classification. These findings can be naturally cast into our framework by considering these patterns as non-robust features, providing further evidence about their pervasiveness.

Manipulating dataset features Ding et al. [Din+19] perform synthetic transformations on the dataset (e.g., image saturation) and study the performance of models on the transformed dataset under standard and robust training. While this can be seen as a method of restricting the features available to the model during

training, it is unclear how well these models would perform on the standard test set. Geirhos et al. [Gei+19] aim to quantify the relative dependence of standard models on shape and texture information of the input. They introduce a version of ImageNet where texture information has been removed and observe an improvement to certain corruptions.

C Experimental Setup

C.1 Datasets

For our experimental analysis, we use the CIFAR-10 [Kri09] and (restricted) ImageNet [Rus+15] datasets. Attaining robust models for the complete ImageNet dataset is known to be a challenging problem, both due to the hardness of the learning problem itself, as well as the computational complexity. We thus restrict our focus to a subset of the dataset which we denote as restricted ImageNet. To this end, we group together semantically similar classes from ImageNet into 9 super-classes shown in Table 2. We train and evaluate only on examples corresponding to these classes.

Class	**Corresponding ImageNet Classes**
"Dog"	151 to 268
"Cat"	281 to 285
"Frog"	30 to 32
"Turtle"	33 to 37
"Bird"	80 to 100
"Primate"	365 to 382
"Fish"	389 to 397
"Crab"	118 to 121
"Insect"	300 to 319

Table 2: Classes used in the Restricted ImageNet model. The class ranges are inclusive.

C.2 Models

We use the ResNet-50 architecture for our baseline standard and adversarially trained classifiers on CIFAR-10 and restricted ImageNet. For each model, we grid search over three learning rates (0.1, 0.01, 0.05), two batch sizes (128, 256) including/not including a learning rate drop (a single order of magnitude) and data augmentation. We use the standard training parameters for the remaining parameters. The hyperparameters used for each model are given in Table 3.

Dataset	LR	Batch Size	LR Drop	Data Aug.	Momentum	Weight Decay
$\widehat{\mathcal{D}}_R$ (CIFAR)	0.1	128	Yes	Yes	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_R$ (Restricted ImageNet)	0.01	128	No	Yes	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_{NR}$ (CIFAR)	0.1	128	Yes	Yes	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_{rand}$ (CIFAR)	0.01	128	Yes	Yes	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_{rand}$ (Restricted ImageNet)	0.01	256	No	No	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_{det}$ (CIFAR)	0.1	128	Yes	No	0.9	$5 \cdot 10^{-4}$
$\widehat{\mathcal{D}}_{det}$ (Restricted ImageNet)	0.05	256	No	No	0.9	$5 \cdot 10^{-4}$

Table 3: Hyperparameters for the models trained in the main paper. All hyperparameters were obtained through a grid search.

C.3 Adversarial training

To obtain robust classifiers, we employ the adversarial training methodology proposed in [Mad+18]. Specifically, we train against a projected gradient descent (PGD) adversary constrained in ℓ_2-norm starting from the original image. Following Madry et al. [Mad+18] we normalize the gradient at each step of PGD to ensure that we move a fixed distance in ℓ_2-norm per step. Unless otherwise specified, we use the values of ϵ provided in Table 4 to train/evaluate our models. We used 7 steps of PGD with a step size of $\varepsilon/5$.

Adversary	**CIFAR-10**	**Restricted Imagenet**
ℓ_2	0.5	3

Table 4: Value of ϵ used for ℓ_2 adversarial training/evaluation of each dataset.

C.4 Constructing a Robust Dataset

In Section 3.1, we describe a procedure to construct a dataset that contains features relevant only to a given (standard/robust) model. To do so, we optimize the training objective in (6). Unless otherwise specified, we initialize x_r as a different randomly chosen sample from the training set. (For the sake of completeness, we also try initializing with a Gaussian noise instead as shown in Table 7.) We then perform normalized gradient descent (ℓ_2-norm of gradient is fixed to be constant at each step). At each step we clip the input x_r to in the $[0, 1]$ range so as to ensure that it is a valid image. Details on the optimization procedure are shown in Table 5. We provide the pseudocode for the construction in Figure 5.

GETROBUSTDATASET(D)

1. $C_R \leftarrow$ ADVERSARIALTRAINING(D)
   $g_R \leftarrow$ mapping learned by C_R from the input to the representation layer
2. $D_R \leftarrow \{\}$
3. For $(x, y) \in D$
   $x' \sim D$
   $x_R \leftarrow \arg\min_{z \in [0,1]^d} \|g_R(z) - g_R(x)\|_2$ # Solved using ℓ_2-PGD starting from x'
   $D_R \leftarrow D_R \bigcup \{(x_R, y)\}$
4. Return D_R

Figure 5: Algorithm to construct a "robust" dataset, by restricting to features used by a robust model.

	CIFAR-10	**Restricted Imagenet**
step size	0.1	1
iterations	1000	2000

Table 5: Parameters used for optimization procedure to construct dataset in Section 3.1.

C.5 Non-robust features suffice for standard classification

To construct the dataset as described in Section 3.2, we use the standard projected gradient descent (PGD) procedure described in [Mad+18] to construct an adversarial example for a given input from the dataset (7). Perturbations are constrained in ℓ_2-norm while each PGD step is normalized to a fixed step size. The details for our PGD setup are described in Table 6. We provide pseudocode in Figure 6.

GETNONROBUSTDATASET(D, ε)

1. $D_{NR} \leftarrow \{\}$
2. $C \leftarrow$ STANDARDTRAINING(D)
3. For $(x, y) \in D$
   - $t \overset{\text{uar}}{\sim} [C]$ # or $t \leftarrow (y+1) \mod C$
   - $x_{NR} \leftarrow \min_{||x'-x|| \leq \varepsilon} L_C(x', t)$ # Solved using ℓ_2 PGD
   - $D_{NR} \leftarrow D_{NR} \bigcup \{(x_{NR}, t)\}$
4. Return D_{NR}

Figure 6: Algorithm to construct a dataset where input-label association is based entirely on non-robust features.

Attack Parameters	CIFAR-10	Restricted Imagenet
ε	0.5	3
step size	0.1	0.1
iterations	100	100

Table 6: Projected gradient descent parameters used to construct constrained adversarial examples in Section 3.2.

D Omitted Experiments and Figures

D.1 Detailed evaluation of models trained on "robust" dataset

In Section 3.1, we generate a "robust" training set by restricting the dataset to only contain features relevant to a robust model (robust dataset) or a standard model (non-robust dataset). This is performed by choosing either a random input from the training set or random noise[16] and then performing the optimization procedure described in (6). The performance of these classifiers along with various baselines is shown in Table 7. We observe that while the robust dataset constructed from noise resembles the original, the corresponding non-robust does not (Figure 7). This also leads to suboptimal performance of classifiers trained on this dataset (only 46% standard accuracy) potentially due to a distributional shift.

		Robust Accuracy	
Model	**Accuracy**	$\varepsilon = 0.25$	$\varepsilon = 0.5$
Standard Training	95.25 %	4.49%	0.0%
Robust Training	90.83%	82.48%	70.90%
Trained on non-robust dataset (constructed from images)	**87.68%**	0.82%	0.0%
Trained on non-robust dataset (constructed from noise)	**45.60%**	1.50%	0.0%
Trained on robust dataset (constructed from images)	85.40%	**48.20 %**	21.85%
Trained on robust dataset (constructed from noise)	84.10%	**48.27 %**	29.40%

Table 7: Standard and robust classification performance on the CIFAR-10 test set of: an (i) ERM classifier; (ii) ERM classifier trained on a dataset obtained by distilling features relevant to ERM classifier in (i); (iii) adversarially trained classifier ($\varepsilon = 0.5$); (iv) ERM classifier trained on dataset obtained by distilling features used by robust classifier in (iii). Simply restricting the set of available features during ERM to features used by a standard model yields non-trivial robust accuracy.

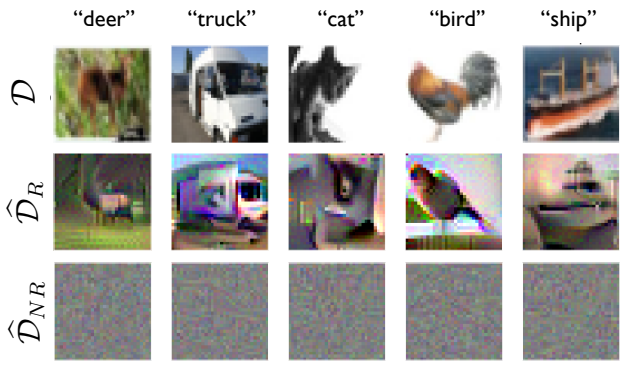


Figure 7: Robust and non-robust datasets for CIFAR-10 when the process starts from noise (as opposed to random images as in Figure 2a).

[16] We use 10k steps to construct the dataset from noise, instead to using 1k steps done when the input is a different training set image (cf. Table 5).

D.2 Adversarial evaluation

To verify the robustness of our classifiers trained on the 'robust" dataset, we evaluate them with strong attacks [Car+19]. In particular, we try up to 2500 steps of projected gradient descent (PGD), increasing steps until the accuracy plateaus, and also try the CW-ℓ_2 loss function [CW17b] with 1000 steps. For each attack we search over step size. We find that over all attacks and step sizes, the accuracy of the model does not drop by more than 2%, and plateaus at 48.27% for both PGD and CW-ℓ_2 (the value given in Figure 2). We show a plot of accuracy in terms of the number of PGD steps used in Figure 8.

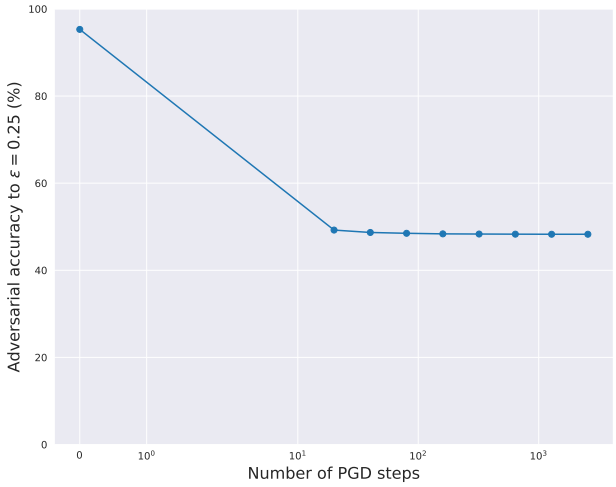


Figure 8: Robust accuracy as a function of the number of PGD steps used to generate the attack. The accuracy plateaus at 48.27%.

D.3 Performance of "robust" training and test set

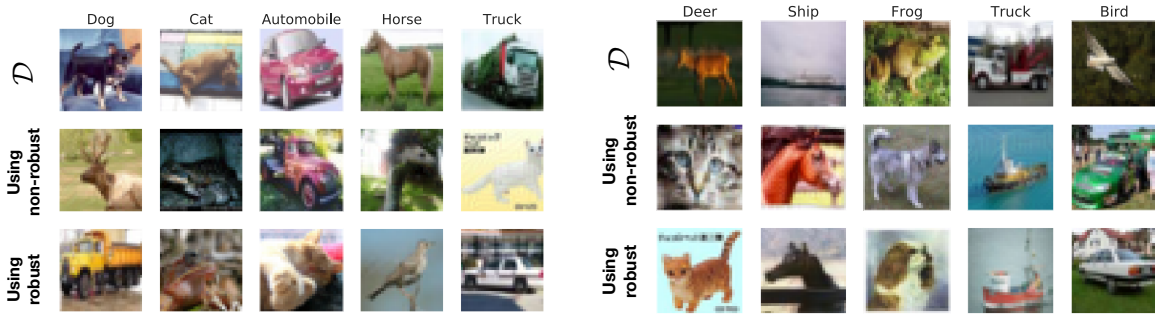
In Section 3.1, we observe that an ERM classifier trained on a "robust" training dataset $\widehat{\mathcal{D}}_R$ (obtained by restricting features to those relevant to a robust model) attains non-trivial robustness (cf. Figure 1 and Table 7). In Table 8, we evaluate the adversarial accuracy of the model on the corresponding robust training set (the samples which the classifier was trained on) and test set (unseen samples from $\widehat{\mathcal{D}}_R$, based on the test set). We find that the drop in robustness comes from a combination of generalization gap (the robustness on the $\widehat{\mathcal{D}}_R$ test set is worse than it is on the robust training set) and distributional shift (the model performs better on the robust test set consisting of unseen samples from $\widehat{\mathcal{D}}_R$ than on the standard test set containing unseen samples from $\mathcal{D}$).

Dataset	Robust Accuracy
Robust training set	77.33%
Robust test set	62.49%
Standard test set	48.27%

Table 8: Performance of model trained on the *robust dataset* on the robust training and test sets as well as the standard CIFAR-10 test set. We observe that the drop in robust accuracy stems from a combination of generalization gap and distributional shift. The adversary is constrained to $\varepsilon = 0.25$ in ℓ_2-norm.

D.4 Classification based on non-robust features

Figure 9 shows sample images from $\mathcal{D}$, $\widehat{\mathcal{D}}_{rand}$ and $\widehat{\mathcal{D}}_{det}$ constructed using a standard (non-robust) ERM classifier, and an adversarially trained (robust) classifier.



(a) $\widehat{\mathcal{D}}_{rand}$ (b) $\widehat{\mathcal{D}}_{det}$

Figure 9: Random samples from datasets where the input-label correlation is entirely based on non-robust features. Samples are generated by performing small adversarial perturbations using either random ($\widehat{\mathcal{D}}_{rand}$) or deterministic ($\widehat{\mathcal{D}}_{det}$) label-target mappings for every sample in the training set. Each image shows: *top*: original; *middle*: adversarial perturbations using a standard ERM-trained classifier; *bottom*: adversarial perturbations using a robust classifier (adversarially trained against $\varepsilon = 0.5$).

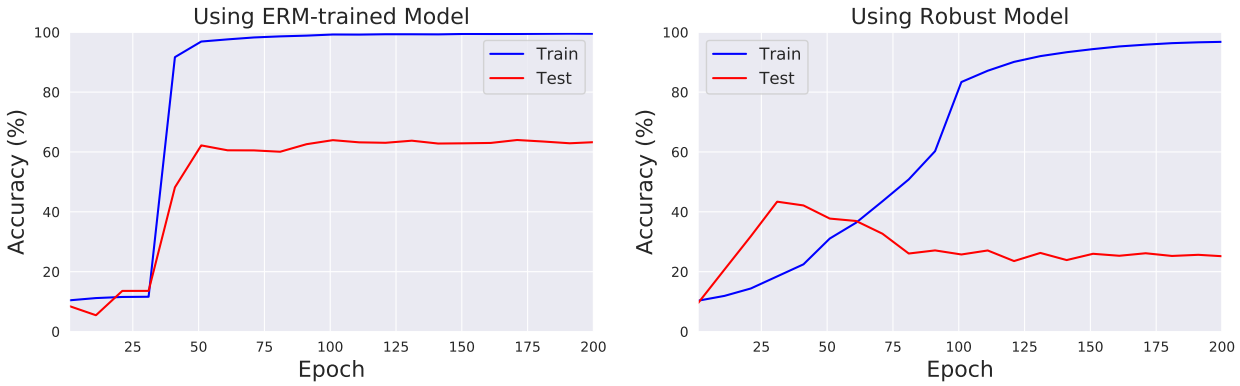
In Table 9, we repeat the experiments in Table 1 based on datasets constructed using a robust model. Note that using a robust model to generate the $\widehat{\mathcal{D}}_{det}$ and $\widehat{\mathcal{D}}_{rand}$ datasets will not result in non-robust features that are strongly predictive of t (since the prediction of the classifier will not change). Thus, training a model on these datasets leads to poor accuracy on the standard test set from $\mathcal{D}$.

Observe from Figure 10 that models trained on datasets derived from the robust model show a decline in test accuracy as training progresses. In Table 9, the accuracy numbers reported correspond to the *last* iteration, and not the *best* performance. This is because we have no way to cross-validate in a meaningful way as the validation set itself comes from $\widehat{\mathcal{D}}_{rand}$ or $\widehat{\mathcal{D}}_{det}$, and not from the true data distribution D. Thus, validation accuracy will not be predictive of the true test accuracy, and thus will not help determine when to early stop.

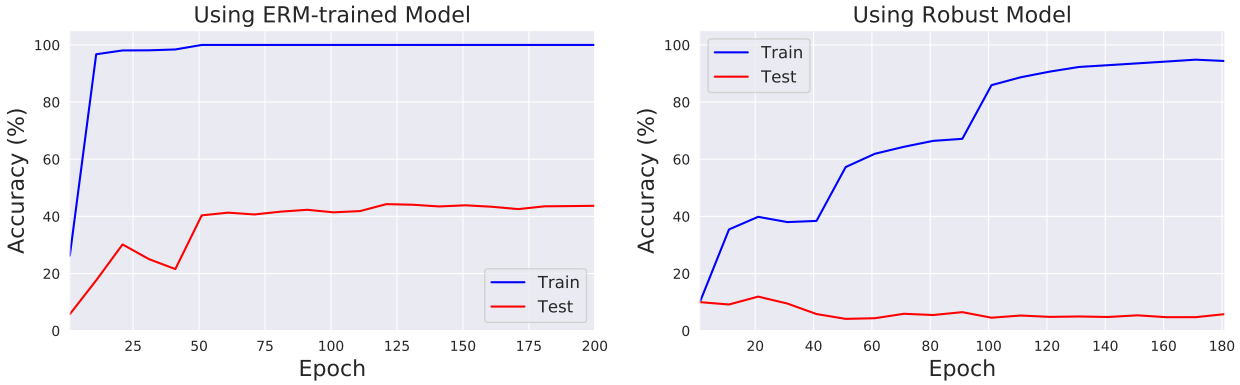
Model used to construct dataset	**Dataset used in training**		
	$\mathcal{D}$	$\widehat{\mathcal{D}}_{rand}$	$\widehat{\mathcal{D}}_{det}$
Robust	95.3%	25.2 %	5.8%
Standard	95.3%	63.3 %	43.7%

Table 9: Repeating the experiments of Table 1 using a robust model to construct the datasets $\mathcal{D}$, $\widehat{\mathcal{D}}_{rand}$ and $\widehat{\mathcal{D}}_{det}$. Results in Table 1 are reiterated for comparison.

D.5 Accuracy curves



(a) Trained using $\widehat{\mathcal{D}}_{rand}$ training set



(b) Trained using $\widehat{\mathcal{D}}_{det}$ training set

Figure 10: Test accuracy on $\mathcal{D}$ of standard classifiers trained on datasets where input-label correlation is based solely on non-robust features as in Section 3.2. The datasets are constructed using either a non-robust/standard model (*left column*) or a robust model (*right column*). The labels used are either random ($\widehat{\mathcal{D}}_{rand}$; *top row*) or correspond to a deterministic permutation ($\widehat{\mathcal{D}}_{det}$; *bottom row*).

D.6 Performance of ERM classifiers on relabeled test set

In Table 10), we evaluate the performance of classifiers trained on $\widehat{\mathcal{D}}_{det}$ on both the original test set drawn from $\mathcal{D}$, and the test set relabelled using $t(y) = (y+1) \mod C$. Observe that the classifier trained on $\widehat{\mathcal{D}}_{det}$ constructed using a robust model actually ends up learning permuted labels based on robust features (indicated by high test accuracy on the relabelled test set).

Model used to construct training dataset for $\widehat{\mathcal{D}}_{det}$	**Dataset used in testing**	
	$\mathcal{D}$	relabelled-$\mathcal{D}$
Standard	43.7%	16.2%
Robust	5.8%	65.5%

Table 10: Performance of classifiers trained using $\widehat{\mathcal{D}}_{det}$ training set constructed using either standard or robust models. The classifiers are evaluated both on the standard test set from $\mathcal{D}$ and the test set relabeled using $t(y) = (y+1) \mod C$. We observe that using a robust model for the construction results in a model that largely predicts the permutation of labels, indicating that the dataset does not have strongly predictive non-robust features.

D.7 Generalization to CIFAR-10.1

Recht et al. [Rec+19] have constructed an unseen but distribution-shifted test set for CIFAR-10. They show that for many previously proposed models, accuracy on the CIFAR-10.1 test set can be predicted as a linear function of performance on the CIFAR-10 test set.

As a sanity check (and a safeguard against any potential adaptive overfitting to the test set via hyper-parameters, historical test set reuse, etc.) we note that the classifiers trained on $\widehat{\mathcal{D}}_{det}$ and $\widehat{\mathcal{D}}_{rand}$ achieve 44% and 55% generalization on the CIFAR-10.1 test set, respectively. This demonstrates non-trivial generalization, and actually perform better than the linear fit would predict (given their accuracies on the CIFAR-10 test set).

D.8 Omitted Results for Restricted ImageNet

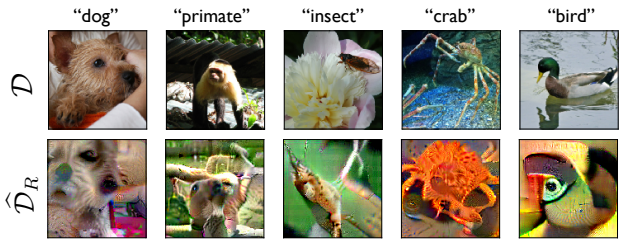


Figure 11: Repeating the experiments shown in Figure 2 for the Restricted ImageNet dataset. Sample images from the resulting dataset.

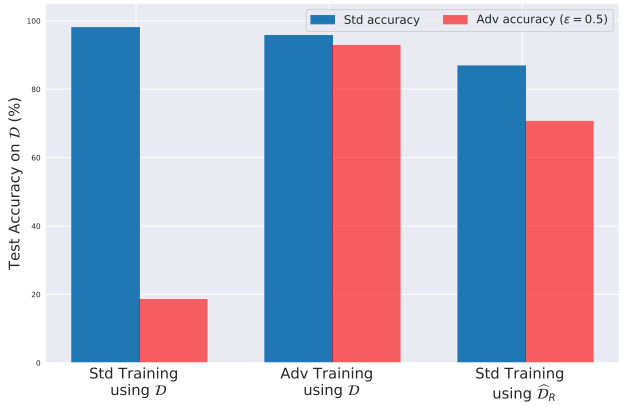


Figure 12: Repeating the experiments shown in Figure 2 for the Restricted ImageNet dataset. Standard and robust accuracy of models trained on these datasets.

D.9 Targeted Transferability

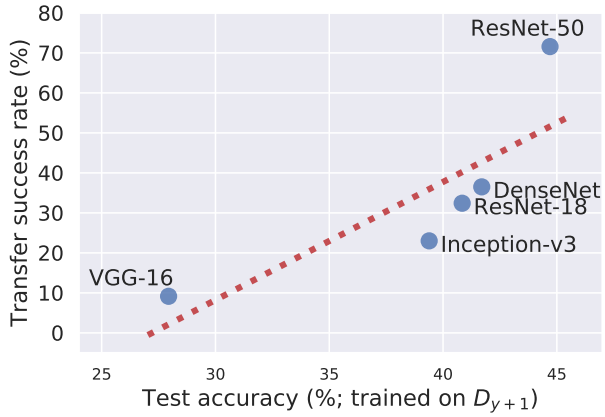


Figure 13: Transfer rate of *targeted* adversarial examples (measured in terms of attack success rate, not just misclassification) from a ResNet-50 to different architectures alongside test set performance of these architecture when trained on the dataset generated in Section 3.2. Architectures more susceptible to transfer attacks also performed better on the standard test set supporting our hypothesis that adversarial transferability arises from utilizing similar *non-robust features*.

D.10 Robustness vs. Accuracy

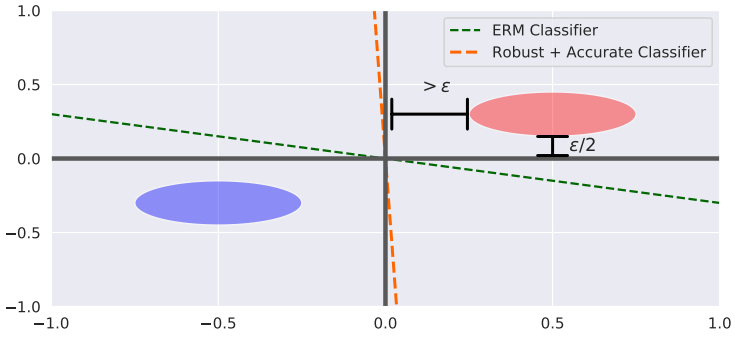


Figure 14: An example where adversarial vulnerability can arise from ERM training on any standard loss function due to non-robust features (the green line shows the ERM-learned decision boundary). There exists, however, a classifier that is both perfectly robust *and* accurate, resulting from robust training, which forces the classifier to ignore the x_2 feature despite its predictiveness.

E Gaussian MLE under Adversarial Perturbation

In this section, we develop a framework for studying non-robust features by studying the problem of *maximum likelihood classification* between two Gaussian distributions. We first recall the setup of the problem, then present the main theorems from Section 4. First we build the techniques necessary for their proofs.

E.1 Setup

We consider the setup where a learner receives labeled samples from two distributions, $\mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, and $\mathcal{N}(-\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$. The learner's goal is to be able to classify new samples as being drawn from $\mathcal{D}_1$ or $\mathcal{D}_2$ according to a maximum likelihood (MLE) rule.

A simple coupling argument demonstrates that this problem can actually be reduced to learning the parameters $\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}$ of a single Gaussian $\mathcal{N}(-\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$, and then employing a linear classifier with weight $\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}$. In the standard setting, maximum likelihoods estimation learns the true parameters, $\boldsymbol{\mu}_*$ and $\boldsymbol{\Sigma}_*$, and thus the learned classification rule is $C(x) = \mathbb{1}\{x^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} > 0\}$.

In this work, we consider the problem of *adversarially robust* maximum likelihood estimation. In particular, rather than simply being asked to classify samples, the learner will be asked to classify *adversarially perturbed* samples $x + \delta$, where $\delta \in \Delta$ is chosen to maximize the loss of the learner. Our goal is to derive the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ corresponding to an adversarially robust maximum likelihood estimate of the parameters of $\mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$. Note that since we have access to $\boldsymbol{\Sigma}_*$ (indeed, the learner can just run non-robust MLE to get access), we work in the space where $\boldsymbol{\Sigma}^*$ is a diagonal matrix, and we restrict the learned covariance $\boldsymbol{\Sigma}$ to the set of diagonal matrices.

Notation. We denote the parameters of the sampled Gaussian by $\boldsymbol{\mu}_* \in \mathbb{R}^d$, and $\boldsymbol{\Sigma}_* \in \{\text{diag}(\boldsymbol{u}) | \boldsymbol{u} \in \mathbb{R}^d\}$. We use $\sigma_{min}(X)$ to represent the smallest eigenvalue of a square matrix X, and $\ell(\cdot; x)$ to represent the Gaussian negative log-likelihood for a single sample x. For convenience, we often use $\boldsymbol{v} = x - \boldsymbol{\mu}$, and $R = \|\boldsymbol{\mu}_*\|$. We also define the $\diagdown$ operator to represent the vectorization of the diagonal of a matrix. In particular, for a matrix $X \in \mathbb{R}^{d\times d}$, we have that $X_{\diagdown} = v \in \mathbb{R}^d$ if $v_i = X_{ii}$.

E.2 Outline and Key Results

We focus on the case where $\Delta = \mathcal{B}_2(\epsilon)$ for some $\epsilon > 0$, i.e. the ℓ_2 ball, corresponding to the following minimax problem:

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)} \left[\max_{\delta:\|\delta\|=\varepsilon} \ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}; x + \delta) \right] \tag{13}$$

We first derive the optimal adversarial perturbation for this setting (Section E.3.1), and prove Theorem 1 (Section E.3.2). We then propose an alternate problem, in which the adversary picks a linear operator to be applied to a fixed vector, rather than picking a specific perturbation vector (Section E.3.3). We argue via Gaussian concentration that the alternate problem is indeed reflective of the original model (and in particular, the two become equivalent as $d \to \infty$). In particular, we propose studying the following in place of (13):

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \max_{M\in\mathcal{M}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)} \left[\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}; x + M(x - \boldsymbol{\mu}))\right] \tag{14}$$

$$\text{where } \mathcal{M} = \left\{ M \in \mathbb{R}^{d\times d} : \; M_{ij} = 0 \; \forall \; i \neq j, \; \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)} \left[\|M\boldsymbol{v}\|_2^2 \right] = \epsilon^2 \right\}.$$

Our goal is to characterize the behavior of the robustly learned covariance $\boldsymbol{\Sigma}$ in terms of the true covariance matrix $\boldsymbol{\Sigma}_*$ and the perturbation budget ε. The proof is through Danskin's Theorem, which allows us to use any maximizer of the inner problem M^* in computing the subgradient of the inner minimization. After showing the applicability of Danskin's Theorem (Section E.3.4) and then applying it (Section E.3.5) to prove our main results (Section E.3.7). Our three main results, which we prove in the following section, are presented below.

First, we consider a simplified version of (13), in which the adversary solves a maximization with a fixed Lagrangian penalty, rather than a hard ℓ_2 constraint. In this setting, we show that the loss contributed by

the adversary corresponds to a misalignment between the data metric (the Mahalanobis distance, induced by $\mathbf{\Sigma}^{-1}$), and the ℓ_2 metric:

Theorem 1 (Adversarial vulnerability from misalignment)**.** *Consider an adversary whose perturbation is determined by the "Lagrangian penalty" form of* (12)*, i.e.*

$$\max_{\delta} \ell(x+\delta; y\cdot\boldsymbol{\mu}, \mathbf{\Sigma}) - C\cdot\|\delta\|_2,$$

where $C \geq \frac{1}{\sigma_{min}(\mathbf{\Sigma}_*)}$ *is a constant trading off NLL minimization and the adversarial constraint*[17]*. Then, the adversarial loss* $\mathcal{L}_{adv}$ *incurred by the non-robustly learned* $(\boldsymbol{\mu}, \mathbf{\Sigma})$ *is given by:*

$$\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) = tr\left[\left(I + (C\cdot\mathbf{\Sigma}_* - I)^{-1}\right)^2\right] - d,$$

and, for a fixed $tr(\mathbf{\Sigma}_*) = k$ *the above is minimized by* $\mathbf{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$*.*

We then return to studying (14), where we provide upper and lower bounds on the learned robust covariance matrix $\mathbf{\Sigma}$:

Theorem 2 (Robustly Learned Parameters)**.** *Just as in the non-robust case,* $\boldsymbol{\mu}_r = \boldsymbol{\mu}^*$*, i.e. the true mean is learned. For the robust covariance* $\mathbf{\Sigma}_r$*, there exists an* $\varepsilon_0 > 0$*, such that for any* $\varepsilon \in [0, \varepsilon_0)$*,*

$$\mathbf{\Sigma}_r = \frac{1}{2}\mathbf{\Sigma}_* + \frac{1}{\lambda}\cdot\boldsymbol{I} + \sqrt{\frac{1}{\lambda}\cdot\mathbf{\Sigma}_* + \frac{1}{4}\mathbf{\Sigma}_*^2}, \qquad \textit{where} \qquad \Omega\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}+\varepsilon^{3/2}}\right) \leq \lambda \leq O\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}}\right).$$

Finally, we show that in the worst case over mean vectors $\boldsymbol{\mu}_*$, the gradient of the adversarial robust classifier aligns more with the inter-class vector:

Theorem 3 (Gradient alignment)**.** *Let* $f(x)$ *and* $f_r(x)$ *be monotonic classifiers based on the linear separator induced by standard and* ℓ_2*-robust maximum likelihood classification, respectively. The maximum angle formed between the gradient of the classifier (wrt input) and the vector connecting the classes can be smaller for the robust model:*

$$\min_{\boldsymbol{\mu}} \frac{\langle\boldsymbol{\mu}, \nabla_x f_r(x)\rangle}{\|\boldsymbol{\mu}\|\cdot\|\nabla_x f_r(x)\|} > \min_{\boldsymbol{\mu}} \frac{\langle\boldsymbol{\mu}, \nabla_x f(x)\rangle}{\|\boldsymbol{\mu}\|\cdot\|\nabla_x f(x)\|}.$$

E.3 Proofs

In the first section, we have shown that the classification between two Gaussian distributions with identical covariance matrices centered at $\boldsymbol{\mu}^*$ and $-\boldsymbol{\mu}^*$ can in fact be reduced to learning the parameters of a single one of these distributions.

Thus, in the standard setting, our goal is to solve the following problem:

$$\min_{\boldsymbol{\mu},\mathbf{\Sigma}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\mathbf{\Sigma}^*)}\left[\ell(\boldsymbol{\mu}, \mathbf{\Sigma}; x)\right] := \min_{\boldsymbol{\mu},\mathbf{\Sigma}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\mathbf{\Sigma}^*)}\left[-\log\left(\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}; x)\right)\right].$$

Note that in this setting, one can simply find differentiate ℓ with respect to both $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, and obtain closed forms for both (indeed, these closed forms are, unsurprisingly, $\boldsymbol{\mu}^*$ and $\mathbf{\Sigma}^*$). Here, we consider the existence of a *malicious adversary* who is allowed to perturb each sample point x by some δ. The goal of the adversary is to *maximize* the same loss that the learner is minimizing.

E.3.1 Motivating example: ℓ_2-constrained adversary

We first consider, as a motivating example, an ℓ_2-constrained adversary. That is, the adversary is allowed to perturb each sampled point by $\delta : \|\delta\|_2 = \varepsilon$. In this case, the minimax problem being solved is the following:

$$\min_{\boldsymbol{\mu},\mathbf{\Sigma}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\mathbf{\Sigma}^*)}\left[\max_{\|\delta\|=\varepsilon} \ell(\boldsymbol{\mu}, \mathbf{\Sigma}; x+\delta)\right]. \tag{15}$$

The following Lemma captures the optimal behaviour of the adversary:

[17] The constraint on C is to ensure the problem is concave.

Lemma 1. *In the minimax problem captured in* (15) *(and earlier in* (13)*), the optimal adversarial perturbation δ^* is given by*

$$\delta^* = \left(\lambda \boldsymbol{I} - \boldsymbol{\Sigma}^{-1}\right)^{-1} \boldsymbol{\Sigma}^{-1} \boldsymbol{v} = \left(\lambda \boldsymbol{\Sigma} - \boldsymbol{I}\right)^{-1} \boldsymbol{v}, \tag{16}$$

where $\boldsymbol{v} = x - \boldsymbol{\mu}$, and λ is set such that $\|\delta^*\|_2 = \varepsilon$.

Proof. In this context, we can solve the inner maximization problem with Lagrange multipliers. In the following we write $\Delta = \mathcal{B}_2(\varepsilon)$ for brevity, and discard terms not containing δ as well as constant factors freely:

$$\begin{aligned}
\arg\max_{\delta \in \Delta} \ \ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}; x + \delta) - &= \arg\max_{\delta \in \Delta} \left(x + \delta - \boldsymbol{\mu}\right)^\top \boldsymbol{\Sigma}^{-1} \left(x + \delta - \boldsymbol{\mu}\right) \\
&= \arg\max_{\delta \in \Delta} \ (x - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) + 2\delta^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) + \delta^\top \boldsymbol{\Sigma}^{-1} \delta \\
&= \arg\max_{\delta \in \Delta} \ \delta^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) + \frac{1}{2} \delta^\top \boldsymbol{\Sigma}^{-1} \delta.
\end{aligned} \tag{17}$$

Now we can solve (17) using the aforementioned Lagrange multipliers. In particular, note that the maximum of (17) is attained at the boundary of the ℓ_2 ball Δ. Thus, we can solve the following system of two equations to find δ, rewriting the norm constraint as $\frac{1}{2}\|\delta\|_2^2 = \frac{1}{2}\varepsilon^2$:

$$\begin{cases} \nabla_\delta \left(\delta^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) + \frac{1}{2} \delta^\top \boldsymbol{\Sigma}^{-1} \delta\right) = \lambda \nabla_\delta \left(\|\delta\|_2^2 - \varepsilon^2\right) \implies \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) + \boldsymbol{\Sigma}^{-1} \delta = \lambda \delta \\ \|\delta\|_2^2 = \varepsilon^2. \end{cases} \tag{18}$$

For clarity, we write $\boldsymbol{v} = x - \boldsymbol{\mu}$: then, combining the above, we have that

$$\delta^* = \left(\lambda \boldsymbol{I} - \boldsymbol{\Sigma}^{-1}\right)^{-1} \boldsymbol{\Sigma}^{-1} \boldsymbol{v} = \left(\lambda \boldsymbol{\Sigma} - \boldsymbol{I}\right)^{-1} \boldsymbol{v}, \tag{19}$$

our final result for the maximizer of the inner problem, where λ is set according to the norm constraint. □

E.3.2 Variant with Fixed Lagrangian (Theorem 1)

To simplify the analysis of Theorem 1, we consider a version of (15) with a fixed Lagrangian penalty, rather than a norm constraint:

$$\max \ell(x + \delta; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma}) - C \cdot \|\delta\|_2.$$

Note then, that by Lemma 1, the optimal perturbation δ^* is given by

$$\delta^* = \left(C\boldsymbol{\Sigma} - \boldsymbol{I}\right)^{-1}.$$

We now proceed to the proof of Theorem 1.

Theorem 1 (Adversarial vulnerability from misalignment)**.** *Consider an adversary whose perturbation is determined by the "Lagrangian penalty" form of* (12)*, i.e.*

$$\max_\delta \ell(x + \delta; y \cdot \boldsymbol{\mu}, \boldsymbol{\Sigma}) - C \cdot \|\delta\|_2,$$

where $C \geq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma}_*)}$ is a constant trading off NLL minimization and the adversarial constraint[18]*. Then, the adversarial loss $\mathcal{L}_{adv}$ incurred by the non-robustly learned $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by:*

$$\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) = tr\left[\left(I + \left(C \cdot \boldsymbol{\Sigma}_* - I\right)^{-1}\right)^2\right] - d,$$

and, for a fixed $tr(\boldsymbol{\Sigma}_*) = k$ the above is minimized by $\boldsymbol{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$.

[18] The constraint on C is to ensure the problem is concave.

Proof. We begin by expanding the Gaussian negative log-likelihood for the relaxed problem:

$$\begin{aligned}\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) &= \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[2\cdot\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}-\boldsymbol{I}\right)^{-\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{v}+\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}-\boldsymbol{I}\right)^{-\top}\boldsymbol{\Sigma}^{-1}\left(C\cdot\boldsymbol{\Sigma}-\boldsymbol{I}\right)^{-1}\boldsymbol{v}\right]\\ &= \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[2\cdot\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}\boldsymbol{\Sigma}-\boldsymbol{\Sigma}\right)^{-1}\boldsymbol{v}+\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}-\boldsymbol{I}\right)^{-\top}\boldsymbol{\Sigma}^{-1}\left(C\cdot\boldsymbol{\Sigma}-\boldsymbol{I}\right)^{-1}\boldsymbol{v}\right]\end{aligned}$$

Recall that we are considering the vulnerability at the MLE parameters $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$:

$$\begin{aligned}\mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) &= \mathbb{E}_{v\sim\mathcal{N}(0,I)}\Big[2\cdot\boldsymbol{v}^\top\boldsymbol{\Sigma}_*^{1/2}\left(C\cdot\boldsymbol{\Sigma}_*^2-\boldsymbol{\Sigma}_*\right)^{-1}\boldsymbol{\Sigma}_*^{1/2}\boldsymbol{v}\\ &\qquad+\boldsymbol{v}^\top\boldsymbol{\Sigma}_*^{1/2}\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-\top}\boldsymbol{\Sigma}_*^{-1}\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\boldsymbol{\Sigma}_*^{1/2}\boldsymbol{v}\Big]\\ &= \mathbb{E}_{v\sim\mathcal{N}(0,I)}\left[2\cdot\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\boldsymbol{v}+\boldsymbol{v}^\top\boldsymbol{\Sigma}_*^{1/2}\left(C^2\boldsymbol{\Sigma}_*^3-2C\cdot\boldsymbol{\Sigma}_*^2+\boldsymbol{\Sigma}_*\right)^{-1}\boldsymbol{\Sigma}_*^{1/2}\boldsymbol{v}\right]\\ &= \mathbb{E}_{v\sim\mathcal{N}(0,I)}\left[2\cdot\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\boldsymbol{v}+\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-2}\boldsymbol{v}\right]\\ &= \mathbb{E}_{v\sim\mathcal{N}(0,I)}\left[-\|v\|_2^2+\boldsymbol{v}^\top\boldsymbol{I}\boldsymbol{v}+2\cdot\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\boldsymbol{v}+\boldsymbol{v}^\top\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-2}\boldsymbol{v}\right]\\ &= \mathbb{E}_{v\sim\mathcal{N}(0,I)}\left[-\|v\|_2^2+\boldsymbol{v}^\top\left(\boldsymbol{I}+\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\right)^2\boldsymbol{v}\right]\\ &= \mathrm{tr}\left[\left(\boldsymbol{I}+\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\right)^2\right]-d\end{aligned}$$

This shows the first part of the theorem. It remains to show that for a fixed $k = \mathrm{tr}(\boldsymbol{\Sigma}_*)$, the adversarial risk is minimized by $\boldsymbol{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$:

$$\begin{aligned}\min_{\boldsymbol{\Sigma}_*}\ \mathcal{L}_{adv}(\Theta) - \mathcal{L}(\Theta) &= \min_{\boldsymbol{\Sigma}_*}\ \mathrm{tr}\left[\left(\boldsymbol{I}+\left(C\cdot\boldsymbol{\Sigma}_*-\boldsymbol{I}\right)^{-1}\right)^2\right]\\ &= \min_{\{\sigma_i\}}\ \sum_{i=1}^{d}\left(1+\frac{1}{C\cdot\sigma_i-1}\right)^2,\end{aligned}$$

where $\{\sigma_i\}$ are the eigenvalues of $\boldsymbol{\Sigma}_*$. Now, we have that $\sum\sigma_i = k$ by assumption, so by optimality conditions, we have that $\boldsymbol{\Sigma}_*$ minimizes the above if $\nabla_{\{\sigma_i\}} \propto \vec{1}$, i.e. if $\nabla_{\sigma_i} = \nabla_{\sigma_j}$ for all i, j. Now,

$$\begin{aligned}\nabla_{\sigma_i} &= -2\cdot\left(1+\frac{1}{C\cdot\sigma_i-1}\right)\cdot\frac{C}{(C\cdot\sigma_i-1)^2}\\ &= -2\cdot\frac{C^2\cdot\sigma_i}{(C\cdot\sigma_i-1)^3}.\end{aligned}$$

Then, by solving analytically, we find that

$$-2\cdot\frac{C^2\cdot\sigma_i}{(C\cdot\sigma_i-1)^3} = -2\cdot\frac{C^2\cdot\sigma_j}{(C\cdot\sigma_j-1)^3}$$

admits only one real solution, $\sigma_i = \sigma_j$. Thus, $\boldsymbol{\Sigma}_* \propto \boldsymbol{I}$. Scaling to satisfy the trace constraint yields $\boldsymbol{\Sigma}_* = \frac{k}{d}\boldsymbol{I}$, which concludes the proof. □

E.3.3 Real objective

Our motivating example (Section E.3.1) demonstrates that the optimal perturbation for the adversary in the ℓ_2-constrained case is actually a linear function of $\boldsymbol{v}$, and in particular, that the optimal perturbation can be expressed as $D\boldsymbol{v}$ for a diagonal matrix D. Note, however, that the problem posed in (15) is not actually

a minimax problem, due to the presence of the expectation between the outer minimization and the inner maximization. Motivated by this and (19), we define the following robust problem:

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \max_{M\in\mathcal{M}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)} \left[\ell(\boldsymbol{\mu},\boldsymbol{\Sigma}; x + M\boldsymbol{v})\right], \tag{20}$$
$$\text{where } \mathcal{M} = \left\{M \in \mathbb{R}^{d\times d} :\ M_{ij} = 0\ \forall\, i \neq j,\ \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\|M\boldsymbol{v}\|_2^2\right] = \epsilon^2\right\}.$$

First, note that this objective is slightly different from that of (15). In the motivating example, δ is constrained to *always* have ε-norm, and thus is normalizer on a per-sample basis inside of the expectation. In contrast, here the classifier is concerned with being robust to perturbations that are linear in $\boldsymbol{v}$, and of ε^2 squared norm *in expectation*.

Note, however, that via the result of Laurent and Massart [LM00] showing strong concentration for the norms of Gaussian random variables, in high dimensions this bound on expectation has a corresponding high-probability bound on the norm. In particular, this implies that as $d \to \infty$, $\|M\boldsymbol{v}\|_2 = \varepsilon$ almost surely, and thus the problem becomes identical to that of (15). We now derive the optimal M for a given $(\boldsymbol{\mu},\boldsymbol{\Sigma})$:

Lemma 2. *Consider the minimax problem described by* (20)*, i.e.*

$$\min_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \max_{M\in\mathcal{M}} \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)} \left[\ell(\boldsymbol{\mu},\boldsymbol{\Sigma}; x + M\boldsymbol{v})\right].$$

Then, the optimal action M^* of the inner maximization problem is given by

$$M = (\lambda\boldsymbol{\Sigma} - \boldsymbol{I})^{-1}, \tag{21}$$

where again λ is set so that $M \in \mathcal{M}$.

Proof. We accomplish this in a similar fashion to what was done for δ^*, using Lagrange multipliers:

$$\begin{aligned}
\nabla_M \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\boldsymbol{v}^\top M\boldsymbol{\Sigma}^{-1}\boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^\top M\boldsymbol{\Sigma}^{-1}M\boldsymbol{v}\right] &= \lambda\nabla_M \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\|M\boldsymbol{v}\|_2^2 - \varepsilon^2\right] \\
\mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{v}\boldsymbol{v}^\top + \boldsymbol{\Sigma}^{-1}M\boldsymbol{v}\boldsymbol{v}^\top\right] &= \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\lambda M\boldsymbol{v}\boldsymbol{v}^\top\right] \\
\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^* + \boldsymbol{\Sigma}^{-1}M\boldsymbol{\Sigma}^* &= \lambda M\boldsymbol{\Sigma}^* \\
M &= (\lambda\boldsymbol{\Sigma} - \boldsymbol{I})^{-1},
\end{aligned}$$

where λ is a constant depending on $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ enforcing the expected squared-norm constraint. □

Indeed, note that the optimal M for the adversary takes a near-identical form to the optimal δ (19), with the exception that λ is not sample-dependent but rather varies only with the parameters.

E.3.4 Danskin's Theorem

The main tool in proving our key results is Danskin's Theorem [Dan67], a powerful theorem from minimax optimization which contains the following key result:

Theorem 4 (Danskin's Theorem)**.** *Suppose $\phi(x,z) : \mathbb{R}\times Z \to \mathbb{R}$ is a continuous function of two arguments, where $Z \subset \mathbb{R}^m$ is compact. Define $f(x) = \max_{z\in Z}\phi(x,z)$. Then, if for every $z \in Z$, $\phi(x,z)$ is convex and differentiable in x, and $\frac{\partial\phi}{\partial x}$ is continuous:*

The subdifferential of $f(x)$ is given by

$$\partial f(x) = \operatorname{conv}\left\{\frac{\partial\phi(x,z)}{\partial x} : z \in Z_0(x)\right\},$$

where $\operatorname{conv}(\cdot)$ *represents the convex hull operation, and Z_0 is the set of maximizers defined as*

$$Z_0(x) = \left\{\overline{z} : \phi(x,\overline{z}) = \max_{z\in Z}\phi(x,z)\right\}.$$

In short, given a minimax problem of the form $\min_x \max_{y \in C} f(x, y)$ where C is a compact set, if $f(\cdot, y)$ is convex for all values of y, then rather than compute the gradient of $g(x) := \max_{y \in C} f(x, y)$, we can simply find a maximizer y^* for the current parameter x; Theorem 4 ensures that $\nabla_x f(x, y^*) \in \partial_x g(x)$. Note that $\mathcal{M}$ is trivially compact (by the Heine-Borel theorem), and differentiability/continuity follow rather straightforwardly from our reparameterization (c.f. (22)), and so it remains to show that the outer minimization is convex for any fixed M.

Convexity of the outer minimization. Note that even in the standard case (i.e. non-adversarial), the Gaussian negative log-likelihood is not convex with respect to $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus, rather than proving convexity of this function directly, we employ the parameterization used by [Das+19]: in particular, we write the problem in terms of $\boldsymbol{T} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{m} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$. Under this parameterization, we show that the robust problem is convex for any fixed M.

Lemma 3. *Under the aforementioned parameterization of $\boldsymbol{T} = \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{m} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, the following "Gaussian robust negative log-likelihood" is convex:*

$$\mathbb{E}_{x \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)} \left[\ell(\boldsymbol{m}, \boldsymbol{T}; x + Mv) \right].$$

Proof. To prove this, we show that the likelihood is convex even with respect to a single sample x; the result follows, since a convex combination of convex functions remains convex. We begin by looking at the likelihood of a single sample $x \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; x + M(x - \boldsymbol{\mu})) &= \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (x - \boldsymbol{\mu})^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) \right) \\
&= \frac{\frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (x - \boldsymbol{\mu})^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) \right)}{\int \frac{1}{\sqrt{(2\pi)^k |(\boldsymbol{I} + \boldsymbol{M})^{-2} \boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (x - \boldsymbol{\mu})^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}) \right)} \\
&= \frac{|I + M|^{-1} \exp\left(-\frac{1}{2} x^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} x + \boldsymbol{\mu}^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} x \right)}{\int \exp\left(-\frac{1}{2} x^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} x + \boldsymbol{\mu}^\top (I + M)^2 \boldsymbol{\Sigma}^{-1} x \right)}
\end{aligned}$$

In terms of the aforementioned $\boldsymbol{T}$ and $\boldsymbol{m}$, and for convenience defining $A = (I + M)^2$:

$$\begin{aligned}
\ell(x) &= |A|^{-1/2} + \left(\frac{1}{2} x^\top A \boldsymbol{T} x - \boldsymbol{m}^\top A x \right) - \log\left(\int \exp\left(\frac{1}{2} x^\top A \boldsymbol{T} x - \boldsymbol{m}^\top A x \right) \right) \\
\nabla \ell(x) &= \begin{bmatrix} \frac{1}{2}(A x x^\top)_{\diamond} \\ -Ax \end{bmatrix} - \frac{\int \begin{bmatrix} \frac{1}{2}(A x x^\top)_{\diamond} \\ -Ax \end{bmatrix} \exp\left(\frac{1}{2} x^\top A \boldsymbol{T} x - \boldsymbol{m}^\top A x \right)}{\int \exp\left(\frac{1}{2} x^\top A \boldsymbol{T} x - \boldsymbol{m}^\top A x \right)} \\
&= \begin{bmatrix} \frac{1}{2}(A x x^\top)_{\diamond} \\ -Ax \end{bmatrix} - \mathbb{E}_{z \sim \mathcal{N}(\boldsymbol{T}^{-1}\boldsymbol{m}, (\boldsymbol{A}\boldsymbol{T})^{-1})} \begin{bmatrix} \frac{1}{2}(A z z^\top)_{\diamond} \\ -Az \end{bmatrix}. \qquad (22)
\end{aligned}$$

From here, following an identical argument to [Das+19] Equation (3.7), we find that

$$\boldsymbol{H}_\ell = \mathrm{Cov}_{z \sim \mathcal{N}\left(\boldsymbol{T}^{-1}\boldsymbol{m}, (A\boldsymbol{T})^{-1}\right)} \left[\begin{pmatrix} \left(-\frac{1}{2} A z z^T \right)_{\diamond} \\ z \end{pmatrix}, \begin{pmatrix} \left(-\frac{1}{2} A z z^T \right)_{\diamond} \\ z \end{pmatrix} \right] \succcurlyeq \mathbf{0},$$

i.e. that the log-likelihood is indeed convex with respect to $\begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{m} \end{bmatrix}$, as desired. □

E.3.5 Applying Danskin's Theorem

The previous two parts show that we can indeed apply Danskin's theorem to the outer minimization, and in particular that the gradient of f at $M = M^*$ is in the subdifferential of the outer minimization problem. We proceed by writing out this gradient explicitly, and then setting it to zero (note that since we have shown f is convex for all choices of perturbation, we can use the fact that a convex function is globally minimized $\iff$ its subgradient contains zero). We continue from above, plugging in (21) for M and using (22) to write the gradients of ℓ with respect to $\boldsymbol{T}$ and $\boldsymbol{m}$.

$$\begin{aligned}
0 = \nabla_{\begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{m}\end{bmatrix}} \ell &= \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\left[\begin{bmatrix}\frac{1}{2}(Axx^\top)^\vee\\ -Ax\end{bmatrix}\right] - \mathbb{E}_{z\sim\mathcal{N}(\boldsymbol{T}^{-1}\boldsymbol{m},(\boldsymbol{A}\boldsymbol{T})^{-1})}\left[\begin{bmatrix}\frac{1}{2}(Azz^\top)^\vee\\ -Az\end{bmatrix}\right]\\
&= \mathbb{E}_{x\sim\mathcal{N}(\boldsymbol{\mu}^*,\boldsymbol{\Sigma}^*)}\begin{bmatrix}\frac{1}{2}(Axx^\top)^\vee\\ -Ax\end{bmatrix} - \mathbb{E}_{z\sim\mathcal{N}(\boldsymbol{T}^{-1}\boldsymbol{m},(\boldsymbol{A}\boldsymbol{T})^{-1})}\begin{bmatrix}\frac{1}{2}(Azz^\top)^\vee\\ -Az\end{bmatrix}\\
&= \begin{bmatrix}\frac{1}{2}(A\boldsymbol{\Sigma}_*)^\vee\\ -A\boldsymbol{\mu}_*\end{bmatrix} - \mathbb{E}_{z\sim\mathcal{N}(\boldsymbol{T}^{-1}\boldsymbol{m},(\boldsymbol{A}\boldsymbol{T})^{-1})}\begin{bmatrix}\frac{1}{2}(A(A\boldsymbol{T})^{-1})^\vee\\ -A\boldsymbol{T}^{-1}\boldsymbol{m}\end{bmatrix}\\
&= \begin{bmatrix}\frac{1}{2}A\boldsymbol{\Sigma}_*\\ -A\boldsymbol{\mu}_*\end{bmatrix} - \begin{bmatrix}\frac{1}{2}A(A\boldsymbol{T})^{-1}\\ -A\boldsymbol{T}^{-1}\boldsymbol{m}\end{bmatrix}\\
&= \begin{bmatrix}\frac{1}{2}A\boldsymbol{\Sigma}_* - \frac{1}{2}\boldsymbol{T}^{-1}\\ A\boldsymbol{T}^{-1}\boldsymbol{m} - A\boldsymbol{\mu}_*\end{bmatrix}
\end{aligned} \tag{23}$$

Using this fact, we derive an *implicit* expression for the robust covariance matrix $\boldsymbol{\Sigma}$. Note that for the sake of brevity, we now use M to denote the optimal adversarial perturbation (previously defined as M^* in (21)). This implicit formulation forms the foundation of the bounds given by our main results.

Lemma 4. *The minimax problem discussed throughout this work admits the following (implicit) form of solution:*

$$\boldsymbol{\Sigma} = \frac{1}{\lambda}I + \frac{1}{2}\boldsymbol{\Sigma}_* + \sqrt{\frac{1}{\lambda}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2},$$

where λ is such that $M \in \mathcal{M}$, and is thus dependent on $\boldsymbol{\Sigma}$.

Proof. Rewriting (23) in the standard parameterization (with respect to $\boldsymbol{\mu}, \boldsymbol{\Sigma}$) and re-expanding $A = (I + M)^2$ yields:

$$0 = \nabla_{\begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{m}\end{bmatrix}} \ell = \begin{bmatrix}\frac{1}{2}(I+M)^2\boldsymbol{\Sigma}_* - \frac{1}{2}\boldsymbol{\Sigma}\\ (I+M)^2\boldsymbol{\mu} - (I+M)^2\boldsymbol{\mu}_*\end{bmatrix}$$

Now, note that the equations involving $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are completely independent, and thus can be solved separately. In terms of $\boldsymbol{\mu}$, the relevant system of equations is $A\boldsymbol{\mu} - A\boldsymbol{\mu}_* = 0$, where multiplying by the inverse A gives that

$$\boldsymbol{\mu} = \boldsymbol{\mu}_*. \tag{24}$$

This tells us that the mean learned via ℓ_2-robust maximum likelihood estimation is precisely the true mean of the distribution.

Now, in the same way, we set out to find $\boldsymbol{\Sigma}$ by solving the relevant system of equations:

$$\boldsymbol{\Sigma}_*^{-1} = \boldsymbol{\Sigma}^{-1}(M+I)^2. \tag{25}$$

Now, we make use of the Woodbury Matrix Identity in order to write $(I + M)$ as

$$I + (\lambda\boldsymbol{\Sigma} - I)^{-1} = I + \left(-I - \left(\frac{1}{\lambda}\boldsymbol{\Sigma}^{-1} - I\right)^{-1}\right) = -\left(\frac{1}{\lambda}\boldsymbol{\Sigma}^{-1} - I\right)^{-1}.$$

Thus, we can revisit (25) as follows:

$$\begin{aligned}
\boldsymbol{\Sigma}_*^{-1} &= \boldsymbol{\Sigma}^{-1}\left(\frac{1}{\lambda}\boldsymbol{\Sigma}^{-1} - I\right)^{-2} \\
\frac{1}{\lambda^2}\boldsymbol{\Sigma}_*^{-1}\boldsymbol{\Sigma}^{-2} - \left(\frac{2}{\lambda}\boldsymbol{\Sigma}_*^{-1} + I\right)\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}_*^{-1} &= 0 \\
\frac{1}{\lambda^2}\boldsymbol{\Sigma}_*^{-1} - \left(\frac{2}{\lambda}\boldsymbol{\Sigma}_*^{-1} + I\right)\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_*^{-1}\boldsymbol{\Sigma}^2 &= 0
\end{aligned}$$

We now apply the quadratic formula to get an implicit expression for $\boldsymbol{\Sigma}$ (implicit since technically λ depends on $\boldsymbol{\Sigma}$):

$$\begin{aligned}
\boldsymbol{\Sigma} &= \left(\frac{2}{\lambda}\boldsymbol{\Sigma}_*^{-1} + I \pm \sqrt{\frac{4}{\lambda}\boldsymbol{\Sigma}_*^{-1} + I}\right)\frac{1}{2}\boldsymbol{\Sigma}_* \\
&= \frac{1}{\lambda}I + \frac{1}{2}\boldsymbol{\Sigma}_* + \sqrt{\frac{1}{\lambda}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}. \qquad (26)
\end{aligned}$$

This concludes the proof. □

E.3.6 Bounding λ

We now attempt to characterize the shape of λ as a function of ε. First, we use the fact that $\mathbb{E}[\|Xv\|^2] = \mathrm{tr}(X^2)$ for standard normally-drawn v. Thus, λ is set such that $\mathrm{tr}(\boldsymbol{\Sigma}_* M^2) = \varepsilon$, i.e:

$$\sum_{i=0} \frac{\boldsymbol{\Sigma}_{ii}^*}{(\lambda\boldsymbol{\Sigma}_{ii} - 1)^2} = \varepsilon \qquad (27)$$

Now, consider ε^2 as a function of λ. Observe that for $\lambda \geq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma})}$, we have that M must be positive semi-definite, and thus ε^2 decays smoothly from ∞ (at $\lambda = \frac{1}{\sigma_{min}}$) to zero (at $\lambda = \infty$). Similarly, for $\lambda \leq \frac{1}{\sigma_{max}(\boldsymbol{\Sigma})}$, ε decays smoothly as λ *decreases*. Note, however, that such values of λ would necessarily make M *negative semi-definite*, which would actually *help* the log-likelihood. Thus, we can exclude this case; in particular, for the remainder of the proofs, we can assume $\lambda \geq \frac{1}{\sigma_{max}(\boldsymbol{\Sigma})}$.

Also observe that the zeros of ε in terms of λ are only at $\lambda = \pm\infty$. Using this, we can show that there exists some ε_0 for which, for all $\varepsilon < \varepsilon_0$, the only corresponding possible valid value of λ is where $\lambda \geq \frac{1}{\sigma_{min}}$. This idea is formalized in the following Lemma.

Lemma 5. *For every $\boldsymbol{\Sigma}_*$, there exists some $\varepsilon_0 > 0$ for which, for all $\varepsilon \in [0, \varepsilon_0)$ the only admissible value of λ is such that $\lambda \geq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma})}$, and thus such that M is positive semi-definite.*

Proof. We prove the existence of such an ε_0 by lower bounding ε (in terms of λ) for any finite $\lambda > 0$ that does not make M PSD. Providing such a lower bound shows that for small enough ε (in particular, less than this lower bound), the only corresponding values of λ are as desired in the statement[19].

In particular, if M is not PSD, then there must exist at least one index k such that $\lambda\boldsymbol{\Sigma}_{kk} < 1$, and thus $(\lambda\boldsymbol{\Sigma}_{kk} - 1)^2 \leq 1$ for all $\lambda > 0$. We can thus lower bound (27) as:

$$\varepsilon = \sum_{i=0} \frac{\boldsymbol{\Sigma}_{ii}^*}{(\lambda\boldsymbol{\Sigma}_{ii} - 1)^2} \geq \frac{\boldsymbol{\Sigma}_{kk}^*}{(\lambda\boldsymbol{\Sigma}_{kk} - 1)^2} \geq \boldsymbol{\Sigma}_{kk}^* \geq \sigma_{min}(\boldsymbol{\Sigma}^*) > 0 \qquad (28)$$

By contradiction, it follows that for any $\varepsilon < \sigma_{min}(\boldsymbol{\Sigma}_*)^2$, the only admissible λ is such that M is PSD, i.e. according to the statement of the Lemma. □

[19] Since our only goal is existence, we lose many factors from the analysis that would give a tighter bound on ε_0.

In the regime $\varepsilon \in [0, \varepsilon_0)$, note that λ is inversely proportional to ε (i.e. as ε grows, λ decreases). This allows us to get a qualitative view of (26): as the allowed perturbation value increases, the robust covariance $\boldsymbol{\Sigma}$ resembles the identity matrix more and more, and thus assigns more and more variance on initially low-variance features. The $\sqrt{\boldsymbol{\Sigma}_*}$ term indicates that the robust model also adds uncertainty proportional to the square root of the initial variance—thus, low-variance features will have (relatively) more uncertainty in the robust case. Indeed, our main result actually follows as a (somewhat loose) formalization of this intuition.

E.3.7 Proof of main theorems

First, we give a proof of Theorem 2, providing lower and upper bounds on the learned robust covariance $\boldsymbol{\Sigma}$ in the regime $\varepsilon \in [0, \varepsilon_0)$.

Theorem 2 (Robustly Learned Parameters)**.** *Just as in the non-robust case, $\boldsymbol{\mu}_r = \boldsymbol{\mu}^*$, i.e. the true mean is learned. For the robust covariance $\boldsymbol{\Sigma}_r$, there exists an $\varepsilon_0 > 0$, such that for any $\varepsilon \in [0, \varepsilon_0)$,*

$$\boldsymbol{\Sigma}_r = \frac{1}{2}\boldsymbol{\Sigma}_* + \frac{1}{\lambda} \cdot \boldsymbol{I} + \sqrt{\frac{1}{\lambda} \cdot \boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}, \qquad \textit{where} \qquad \Omega\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}+\varepsilon^{3/2}}\right) \leq \lambda \leq O\left(\frac{1+\varepsilon^{1/2}}{\varepsilon^{1/2}}\right).$$

Proof. We have already shown that $\boldsymbol{\mu} = \boldsymbol{\mu}_*$ in the robust case (c.f. (24)). We choose ε_0 to be as described, i.e. the largest ε for which the set $\{\lambda : \mathrm{tr}(\boldsymbol{\Sigma}_*^2 M) = \varepsilon, \lambda \geq 1/\sigma_{\max}(\boldsymbol{\Sigma})\}$ has only one element λ (which, as we argued, must not be less than $1/\sigma_{min}(\boldsymbol{\Sigma})$). We have argued that such an ε_0 must exist.

We prove the result by combining our early derivation (in particular, (25) and (26)) with upper and lower bound on λ, which we can compute based on properties of the trace operator. We begin by deriving a lower bound on λ. By linear algebraic manipulation (given in Appendix E.3.8), we get the following bound:

$$\lambda \geq \frac{d}{\mathrm{tr}(\boldsymbol{\Sigma})}\left(1 + \sqrt{\frac{d \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)}{\varepsilon}}\right) \tag{29}$$

Now, we can use (25) in order to remove the dependency of λ on $\boldsymbol{\Sigma}$:

$$\begin{aligned}
\boldsymbol{\Sigma} &= \boldsymbol{\Sigma}_*(M+I)^2 \\
\mathrm{tr}(\boldsymbol{\Sigma}) &= \mathrm{tr}\left[(\boldsymbol{\Sigma}_*^{1/2}M + \boldsymbol{\Sigma}_*^{1/2})^2\right] \\
&\leq 2 \cdot \mathrm{tr}\left[(\boldsymbol{\Sigma}_*^{1/2}M)^2 + (\boldsymbol{\Sigma}_*^{1/2})^2\right] \\
&\leq 2 \cdot (\varepsilon + \mathrm{tr}(\boldsymbol{\Sigma}_*)).
\end{aligned}$$

Applying this to (29) yields:

$$\lambda \geq \frac{d/2}{\varepsilon + \mathrm{tr}(\boldsymbol{\Sigma}_*)}\left(1 + \sqrt{\frac{d \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)}{\varepsilon}}\right).$$

Note that we can simplify this bound significantly by writing $\varepsilon = d \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)\varepsilon' \leq \mathrm{tr}(\boldsymbol{\Sigma}_*)\varepsilon'$, which does not affect the result (beyond rescaling the valid regime $(0, \varepsilon_0)$), and gives:

$$\lambda \geq \frac{d/2}{(1+\varepsilon')\mathrm{tr}(\boldsymbol{\Sigma}_*)}\left(1 + \frac{1}{\sqrt{\varepsilon'}}\right) \geq \frac{d \cdot (1+\sqrt{\varepsilon'})}{2\sqrt{\varepsilon'}(1+\varepsilon')\mathrm{tr}(\boldsymbol{\Sigma}_*)}$$

Next, we follow a similar methodology (Appendix E.3.8) in order to upper bound λ:

$$\lambda \leq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma})}\left(\sqrt{\frac{\|\boldsymbol{\Sigma}_*\|_F \cdot d}{\varepsilon}} + 1\right).$$

Note that by (25) and positive semi-definiteness of M, it must be that $\sigma_{min}(\boldsymbol{\Sigma}) \geq \sigma_{min}(\boldsymbol{\Sigma}_*)$. Thus, we can simplify the previous expression, also substituting $\varepsilon = d \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)\varepsilon'$:

$$\lambda \leq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma}_*)}\left(\sqrt{\frac{\|\boldsymbol{\Sigma}_*\|_F}{\sigma_{min}(\boldsymbol{\Sigma}_*)\varepsilon'}} + 1\right) = \frac{\|\boldsymbol{\Sigma}_*\|_F + \sqrt{\varepsilon \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)}}{\sigma_{min}(\boldsymbol{\Sigma}_*)^{3/2}\sqrt{\varepsilon}}$$

These bounds can be straightforwardly combined with Lemma 4, which concludes the proof. □

Using this theorem, we can now show Theorem 3:

Theorem 3 (Gradient alignment)**.** *Let $f(x)$ and $f_r(x)$ be monotonic classifiers based on the linear separator induced by standard and ℓ_2-robust maximum likelihood classification, respectively. The maximum angle formed between the gradient of the classifier (wrt input) and the vector connecting the classes can be smaller for the robust model:*

$$\min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f_r(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f_r(x)\|} > \min_{\boldsymbol{\mu}} \frac{\langle \boldsymbol{\mu}, \nabla_x f(x) \rangle}{\|\boldsymbol{\mu}\| \cdot \|\nabla_x f(x)\|}.$$

Proof. To prove this, we make use of the following Lemmas:

Lemma 6. *For two positive definite matrices A and B with $\kappa(A) > \kappa(B)$, we have that $\kappa(A+B) \leq \max\{\kappa(A), \kappa(B)\}$.*

Proof. We proceed by contradiction:

$$\begin{aligned}
\kappa(A+B) &= \frac{\lambda_{max}(A) + \lambda_{max}(B)}{\lambda_{min}(A) + \lambda_{min}(B)} \\
\kappa(A) &= \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \\
\kappa(A) &\geq \kappa(A+B) \\
\iff \lambda_{max}(A)\left(\lambda_{min}(A) + \lambda_{min}(B)\right) &\geq \lambda_{min}(A)\left(\lambda_{max}(A) + \lambda_{max}(B)\right) \\
\iff \lambda_{max}(A)\lambda_{min}(B) &\geq \lambda_{min}(A)\lambda_{max}(B) \\
\iff \frac{\lambda_{max}(A)}{\lambda_{min}(A)} &\geq \frac{\lambda_{min}(A)}{\lambda_{max}(B)},
\end{aligned}$$

which is false by assumption. This concludes the proof. □

Lemma 7 (Straightforward)**.** *For a positive definite matrix A and $k > 0$, we have that*

$$\kappa(A + k \cdot I) < \kappa(A) \qquad \kappa(A + k \cdot \sqrt{A}) \leq \kappa(A).$$

Lemma 8 (Angle induced by positive definite matrix; folklore)**.** [20] *For a positive definite matrix $A \succ 0$ with condition number κ, we have that*

$$\min_x \frac{x^\top A x}{\|Ax\|_2 \cdot \|x\|_2} = \frac{2\sqrt{\kappa}}{1+\kappa}. \tag{30}$$

These two results can be combined to prove the theorem. First, we show that $\kappa(\boldsymbol{\Sigma}) \leq \kappa(\boldsymbol{\Sigma}_*)$:

$$\begin{aligned}
\kappa(\boldsymbol{\Sigma}) &= \kappa\left(\frac{1}{\lambda} I + \frac{1}{2}\boldsymbol{\Sigma}_* + \sqrt{\frac{1}{\lambda}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}\right) \\
&< \max\left\{\kappa\left(\frac{1}{\lambda} I + \frac{1}{2}\boldsymbol{\Sigma}_*\right), \kappa\left(\sqrt{\frac{1}{\lambda}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2}\right)\right\} \\
&< \max\left\{\kappa\left(\boldsymbol{\Sigma}_*\right), \sqrt{\kappa\left(\frac{1}{\lambda}\boldsymbol{\Sigma}_* + \frac{1}{4}\boldsymbol{\Sigma}_*^2\right)}\right\} \\
&= \max\left\{\kappa\left(\boldsymbol{\Sigma}_*\right), \sqrt{\kappa\left(\frac{2}{\lambda}\sqrt{\frac{1}{4}\boldsymbol{\Sigma}_*^2} + \frac{1}{4}\boldsymbol{\Sigma}_*^2\right)}\right\} \\
&\leq \kappa\left(\boldsymbol{\Sigma}_*\right).
\end{aligned}$$

Finally, note that (30) is a strictly decreasing function in κ, and as such, we have shown the theorem. □

[20] A proof can be found in `https://bit.ly/2L6jdAT`

E.3.8 Bounds for λ

Lower bound.

$$
\begin{aligned}
\varepsilon &= \operatorname{tr}(\boldsymbol{\Sigma}_* M^2) \\
&\geq \sigma_{min}(\boldsymbol{\Sigma}_*) \cdot \operatorname{tr}(M^2) && \text{by the definition of } \operatorname{tr}(\cdot) \\
&\geq \frac{\sigma_{min}(\boldsymbol{\Sigma}_*)}{d} \cdot \operatorname{tr}(M)^2 && \text{by Cauchy-Schwarz} \\
&\geq \frac{\sigma_{min}(\boldsymbol{\Sigma}_*)}{d} \cdot \left[\operatorname{tr}\left((\lambda\boldsymbol{\Sigma} - \boldsymbol{I})^{-1}\right)\right]^2 && \text{Expanding } M \text{ (21)} \\
&\geq \frac{\sigma_{min}(\boldsymbol{\Sigma}_*)}{d} \cdot \left[\operatorname{tr}\left(\lambda\boldsymbol{\Sigma} - \boldsymbol{I}\right)^{-1} \cdot d^2\right]^2 && \text{AM-HM inequality} \\
&\geq d^3 \cdot \sigma_{min}(\boldsymbol{\Sigma}_*) \cdot [\lambda \cdot \operatorname{tr}(\boldsymbol{\Sigma}) - d]^{-2} \\
[\lambda \cdot \operatorname{tr}(\boldsymbol{\Sigma}) - d]^2 &\geq \frac{d^3 \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)}{\varepsilon} \\
\lambda \cdot \operatorname{tr}(\boldsymbol{\Sigma}) - d &\geq \frac{d^{3/2} \cdot \sqrt{\sigma_{min}(\boldsymbol{\Sigma}_*)}}{\sqrt{\varepsilon}} && \text{since } M \text{ is PSD} \\
\lambda &\geq \frac{d}{\operatorname{tr}(\boldsymbol{\Sigma})}\left(1 + \sqrt{\frac{d \cdot \sigma_{min}(\boldsymbol{\Sigma}_*)}{\varepsilon}}\right)
\end{aligned}
$$

Upper bound

$$
\begin{aligned}
\varepsilon &= \operatorname{tr}(\boldsymbol{\Sigma}_* M^2) \\
&\leq \|\boldsymbol{\Sigma}_*\|_F \cdot d \cdot \sigma_{max}(M)^2 \\
&\leq \|\boldsymbol{\Sigma}_*\|_F \cdot d \cdot \sigma_{min}(M)^{-2} \\
\lambda \cdot \sigma_{min}(\boldsymbol{\Sigma}) - 1 &\leq \sqrt{\frac{\|\boldsymbol{\Sigma}_*\|_F \cdot d}{\varepsilon}} \\
\lambda &\leq \frac{1}{\sigma_{min}(\boldsymbol{\Sigma})}\left(\sqrt{\frac{\|\boldsymbol{\Sigma}_*\|_F \cdot d}{\varepsilon}} + 1\right).
\end{aligned}
$$