ON THE HUMAN-RECOGNIZABILITY PHENOMENON OF ADVERSARIALLY TRAINED DEEP IMAGE CLASSIFIERS

Jonathan Helland* **Nathan VanHoudnos**
Software Engineering Institute
Carnegie Mellon University
{jwhelland, nmvanhoudnos}@sei.cmu.edu

Abstract

In this work, we investigate the phenomenon that robust image classifiers have human-recognizable features – often referred to as interpretability – as revealed through the input gradients of their score functions and their subsequent adversarial perturbations. In particular, we demonstrate that state-of-the-art methods for adversarial training incorporate two terms – one that orients the decision boundary via minimizing the expected loss, and another that induces smoothness of the classifier's decision surface by penalizing the local Lipschitz constant. Through this demonstration, we provide a unified discussion of gradient and Jacobian-based regularizers that have been used to encourage adversarial robustness in prior works. Following this discussion, we give qualitative evidence that the coupling of smoothness and orientation of the decision boundary is sufficient to induce the aforementioned human-recoginzability phenomenon.

1. Introduction

An adversarial example is often defined as "an input to a ML model that is intentionally designed by an attacker to fool the model into producing an incorrect output" (Goodfellow and Papernot, 2017). Tsipras et al. (2019) observed that the adversarial examples for adversarially trained image classifiers were clearly human-recognizable as particular classes in the training data – a phenomenon that does not generally occur for non-adversarially trained image classifiers.

Figure 1.1 illustrates this human-recognizability phenomenon by comparing the adversarial perturbations for two pre-activation ResNet18 models (He et al., 2016) trained adversarially (orange, "PGD") and non-adversarially (blue, "standard") on the CIFAR-10 dataset (Krizhevsky et al., 2009). The images show targeted adversarial perturbations towards the *ship* class of an image taken from the *bird* class of the test-set, where each perturbation budget value indicates a separate perturbation starting from the same bird image (outlined in black). The adversarially trained model produces perturbations that gradually become human-recognizable as the target class, whereas the non-adversarially trained model remains human-recognizable as a noisy version of the original class.

In this paper, we study adversarial examples from the perspective of human-recognizability. We specifically ask the following question:

> *What is a sufficient condition for the training method of a Convolutional Neural Network (CNN) image classifier such that adversarial perturbations against that CNN model are recognizable to humans?*

In answering this question, we make two contributions:

*Corresponding author.

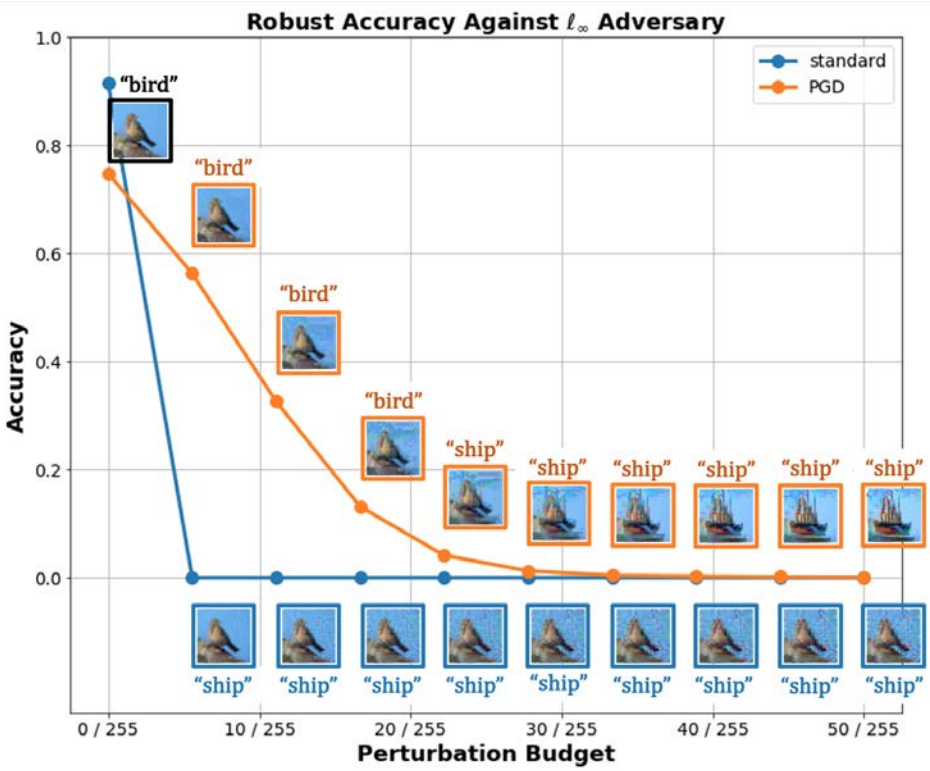


Figure 1.1: Illustration of the **human-recognizability** phenomenon. Adversarially (orange) and non-adversarially (blue) trained pre-activation ResNet-18 models (He et al., 2016) trained on CIFAR-10 (Krizhevsky et al., 2009). The robust accuracy against an untargeted ℓ_∞ adversary are shown alongside targeted adversarial perturbations of a bird image from the test set using varying perturbation budgets.

1. A unified discussion of adversarial training methods around the idea of smoothness regularization. This includes a generalization of Jacobian-based regularizers to ℓ_p with $p \geq 2$ adversaries and a theoretical discussion of the smoothness encouraged by this regularizer via the local Lipschitz constant of the classifier.

2. An empirical demonstration that smoothness regularization is sufficient for human-recognizable adversarial examples. Here we provide qualitative evidence that smoothness regularizers promoting lesser degrees of adversarial robustness provide less human-recognizable adversarial perturbations, suggesting a potential trade-off between robustness and human-recognizability.

Beyond demonstrating that the adversarial examples from smoothness-regularized models are human-recognizable, we do not consider the problem of data privacy in this paper that was suggested by Mejia et al. (2019). Rather, we take as a given that if human-recognizable patterns can be generated from a model without access to the training data, then there will exist privacy attacks capable of violating some security policy for a practical system.

We conclude the paper with a brief discussion of the implications of the suggested trade-off between models that are robust to ℓ_p adversaries and the human-recognizability of their adversarial perturbations. Namely, that machine learning applications that require both robustness and data privacy require careful thought and further research.

2. Background & Related Work

2.1 Background

We restrict our attention to models that perform image classification tasks, the setting in which adversarial examples were first observed (Szegedy et al., 2013). We now discuss both attacks (generation of adversarial examples) and defenses (resistance to adversarial examples) that we will use in the rest of this work.

2.1.1 Attacks

Adversarial examples in the sense of untargeted and targeted evasion attacks are based on the following optimization problems. Take a classifier $f : \mathcal{X} \to (0,1)^K$ over K-many classes with data sample $(\boldsymbol{x}, y) \sim \mathcal{D}$ from the data generating distribution $\mathcal{D}$. We assume that the scores sum to one: $\sum_y f(\boldsymbol{x})[y] = 1$ for any input $\boldsymbol{x}$ (e.g. softmax outputs). For an untargeted attack, our goal is to find a solution to the constrained problem

$$\max_{\boldsymbol{\delta} \in B_\epsilon} \mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x} + \boldsymbol{\delta})), \tag{2.1}$$

where B_ϵ is a constraint set (often an ϵ-radius ℓ_p-ball) and $\mathcal{L}$ is the loss function that consumes the one-hot label vector $\boldsymbol{e}_y \in [0,1]^K$. In words, find a perturbed image that changes the model's prediction to any other class. A targeted attack takes some other label $\widetilde{y} \neq y$ and instead solves the problem

$$\min_{\boldsymbol{\delta} \in B_\epsilon} \mathcal{L}(\boldsymbol{e}_{\widetilde{y}}, f(\boldsymbol{x} + \boldsymbol{\delta})), \tag{2.2}$$

which finds a perturbed image that changes the model's prediction to a specific class.

The Madry Projected Gradient Descent (PGD) algorithm (Madry et al., 2017) is one of the most widely used methods for generating targeted and untargeted adversarial examples, and is based on the following iterative procedure. For some initial point $\boldsymbol{x}_0$ and some label y, it applies PGD to the above optimization problems:

$$\boldsymbol{x}_{t+1} = \mathcal{P}\left(\boldsymbol{x}_t + \alpha \operatorname*{argmax}_{\|\boldsymbol{u}\| \leq 1} \langle \boldsymbol{u}, \nabla_{\boldsymbol{x}} \mathcal{L}\left(\boldsymbol{e}_y, f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)\right)\rangle\right) \tag{2.3}$$

for an arbitrary norm $\|\cdot\|$ (often an ℓ_p-norm, but works like (Wong et al., 2019) consider other norms) and stepsize α. For an untargeted attack, we take $\alpha > 0$ and y as the model's prediction for x, and for a targeted attack, we take $\alpha < 0$ with some particular y chosen *a priori*. The operator $\mathcal{P}$ projects iterates onto the constraint set $B_\epsilon = B_\epsilon(\boldsymbol{x}_0) \cap P$, where $B_\epsilon(\boldsymbol{x}_0)$ is canonically a radius-ϵ norm-ball centered at $\boldsymbol{x}_0$ and the set $P \equiv \{\boldsymbol{x} : \boldsymbol{x}[i] \in [a,b]\}$ for $a < b$, $a, b \in \mathbb{R}$ is the set of images with valid pixel values.

2.1.2 Defenses

We consider an undefended model as one whose learning task is only concerned with maximizing accuracy – specifically, the optimization problem that minimizes the expected loss over the training data without any further regularization:

$$\operatorname*{minimize}_{f \in \mathcal{F}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right]. \tag{2.4}$$

Here, the classifier f belongs to a family of functions $\mathcal{F}$ (e.g. neural networks of a fixed architecture). It is well known that on common image classification tasks, neural networks trained via algorithms based on Eq. (2.4) are highly susceptible to attacks like Eq. (2.1) and Eq. (2.2) (see the blue curve in Figure 1.1 for illustration of this).

The first defense against adversarial examples that we consider is Madry adversarial training (Madry et al., 2017), which is based on the canonical adversarial learning program (Goodfellow et al., 2014)

$$\underset{f \in \mathcal{F}}{\text{minimize}} \; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta} \in B_\epsilon} \mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}+\boldsymbol{\delta})) \right], \tag{2.5}$$

where B_ϵ is canonically a radius-ϵ norm-ball, constraining the adversarial perturbation $\boldsymbol{\delta}$ to be no larger than ϵ. This program directly incorporates the untargeted attack Eq. (2.1) in the training process as an adversary. Note that this minimax procedure is defined across the whole of the training data; the model never is exposed to unperturbed examples. Madry adversarial training remains among the many proposed defenses one of the few that offers meaningful resilience against strong adversaries (Tramer et al., 2020a; Athalye et al., 2018).

It should be noted that the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) can be seen as a single-iteration version of Eq. (2.3) in which the projection operator $\mathcal{P}$ is removed. FGSM can be applied as an effective defense with some minor algorithmic improvements (Wong et al., 2020), which is desirable for large datasets for which Madry adversarial training does not scale well.

The second defense we consider is the TRADES algorithm proposed by (Zhang et al., 2019), which also belongs to the camp of effective defenses. It is based on a similar learning problem, albeit decomposed into an accuracy promoting term – which is calculated on unperturbed data in contrast to Eq. 2.5) – and a robustness promoting term which is calculated on perturbed data:

$$\underset{f \in \mathcal{F}}{\text{minimize}} \; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathcal{L}(\boldsymbol{e}_y,\, f(\boldsymbol{x})) + \max_{\boldsymbol{\delta} \in B_\epsilon} \beta\, \mathcal{L}(f(\boldsymbol{x}), f(\boldsymbol{x}+\boldsymbol{\delta})) \right], \tag{2.6}$$

where the regularization weight $\beta > 0$ trades off between the two terms. It was noted by (Zhang et al., 2019) that in the binary classification setting, f tends to Bayes optimality as $\beta \to 0$, whereas f tends towards an all-ones classifier as $\beta \to \infty$.

Note that for TRADES, the attack is similar to Eq. (2.1, only with $\nabla_{\boldsymbol{x}_t} \mathcal{L}(f(\widetilde{\boldsymbol{x}}_0), f(\boldsymbol{x}_t))$ instead of the loss with respect to the label y, where $\widetilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 + \boldsymbol{\xi}$ with $\boldsymbol{\xi}$ random noise to prevent the gradient from vanishing initially.

We also consider defenses closely related to canonical adversarial training Eq. (2.5) and TRADES Eq. (2.6) insofar as being approximations of both programs. In particular, we will derive connections between these programs and various other proposed defenses (Hoffman et al., 2019; Moosavi-Dezfooli et al., 2019; Jakubovitz and Giryes, 2018; Miyato et al., 2018; Zhao et al., 2019) via Taylor series approximations in Section 3.

We also consider a historical defense, model distillation (Hinton et al., 2015), which smooths a model by raising its softmax temperature parameter and then training a new model using the smoothed predictions as pseudo-labels. More specifically, distillation takes a pre-trained model with logit output $g : \mathcal{X} \to \mathbb{R}^K$, define $p(y \,|\, \boldsymbol{x}) \equiv \text{Softmax}(g(\boldsymbol{x})/\tau)$ for some $\tau > 0$, and solve the learning problem

$$\underset{f \in \mathcal{F}}{\text{minimize}} \; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathcal{L}\left(p(y \,|\, \boldsymbol{x}),\, f(\boldsymbol{x})\right) \right].$$

Larger values of temperature τ push the scores of g closer to a uniform distribution, which results in a smoother model f, which is called the distilled model. Defensive distillation (Papernot et al., 2016) works by training the initial – also called the teacher – model at a raised temperature τ and then distills the model using the same temperature. Defensive distillation as a defense technique is well-known to be ineffective against adversarial examples (Carlini and Wagner, 2016), thus we do not consider it to be a defense in the same sense as the others discussed in this section. However, defensive distillation provides a useful point of comparison with other methods because it is still a smoothness regularizer, which we will discuss in more detail in Section 3.2.

2.2 Related Work

Gradient-based regularizers for the promotion of adversarial robustness have been considered by (Ross and Doshi-Velez, 2017; Etmann et al., 2019; Lin et al., 2019; Finlay and Oberman, 2019). Similarly, Jacobian-based regularizers have been investigated for similar purposes in (Hoffman et al., 2019; Zhao et al., 2019; Moosavi-Dezfooli et al., 2019).

Prior work has observed that the saliency maps (gradients of the score functions with respect to the input) of defended models evaluated at train and test images resemble the input points themselves – a phenomenon often referred to as interpretability (Tsipras et al., 2019; Finlay and Oberman, 2019; Ross and Doshi-Velez, 2017; Tramer et al., 2020b; Etmann et al., 2019). We use the term *human-recognizability* instead of interpretability (which has been used in prior literature to describe this same phenomenon (Ross and Doshi-Velez, 2017)) because interpretability connotates the idea of models that explain their decision processes transparently to the user; it is not clear to us that this visualization of model gradients has any meaningful correspondence with the traditional notion of interpretability. Moreover, "the term *interpretability* holds no agreed upon meaning, and yet machine learning conferences frequently publish papers which wield the term in a quasi-mathematical way" (Lipton, 2018). The term *human-recognizability* better emphasizes the reliance on the qualitative nature of human perception and interpretation, which is a prevalent measure of quality in generative modeling literature (e.g. Generative Adversarial Networks (GANs)).

It is observed in Anil et al. (2019) that classifiers whose Lipschitz constants are constrained have a similar gradient human-recognizability phenomenon. In Kaur et al. (2019), it is demonstrated that this human-recognizability phenomenon also occurs for randomized smoothing (Cohen et al., 2019), and they further suggest that human-recognizability may be a general property of robust models. However, Kaur et al. (2019) does not explore this latter claim for defenses beyond randomized smoothing.

Works like Tsipras et al. (2019); Ross and Doshi-Velez (2017); Grathwohl et al. (2019) have further noted that large adversarial perturbations (under an iterative procedure like Eq. (2.3 with large ϵ) of input points (even noise) to a robust model remain recognizable. The hypothesis of Ilyas et al. (2019) is that standard training Eq. (2.4) is ill-posed and that adversarial training helps to reduce the set of features that the classifier can learn to compute to those that are more human-recognizable. In Santurkar et al. (2019), this human-recognizability phenomenon of robust image classifiers is leveraged to perform computer vision tasks like inpainting and superresolution with a classifier instead of the usual generative modeling framework. Similarly, Engstrom et al. (2019) demonstrates that robust image classifiers can be used for smooth image feature manipulation, also arguing that the feature representations of such models are approximately invertible i.e. the original training images can be approximately recovered.

Model inversion (Fredrikson et al., 2015) – proposed as a privacy attack – can be thought of as an unconstrained, targeted adversarial perturbation (see Appendix F). It was observed in Fredrikson et al. (2015) that these targeted adversarial perturbations are recognizable for linear models and neural networks trained on a small dataset. Although model inversion is generally unrecognizable for undefended convolutional models trained on datasets with sufficiently diverse classes (Papernot et al., 2018; Shokri et al., 2017), it has been observed that model inversion applied to defended models yields highly recognizable perturbations (Terzi et al., 2020; Mejia et al., 2019). Although Mejia et al. (2019) identifies the recognizably phenomenon as a privacy concern, the paper does not consider the conditions under which it appears beyond simply noting that it is present for popular defenses.

In this work, we aim to build on Mejia et al. (2019), Tsipras et al. (2019), and Kaur et al. (2019) by identifying a sufficient condition for the human-recognizability phenomenon to occur.

3. Unification of Adversarial Training around Smoothness Regularization

In this section, we argue that adversarial training of a classifier in the sense of the canonical program Eq. (2.5) and the TRADES program Eq. (2.6) both decompose via Taylor expansions into two terms, one that orients the decision boundary and one that smooths the decision surface.

First, we discuss undefended training Eq. (2.4) as purely orientation of the decision boundary in Section 3.1. We then consider canonical adversarial training Eq. (2.5), and show that a first-order Taylor approximation includes both orientation and smoothing. Finally, we consider the TRADES program Eq. (2.6), showing how this program includes decision boundary orientation as well as a stronger form of smoothing than canonical adversarial training.

Refer to Appendix A for unfamiliar notation that is not defined explicitly within the text.

3.1 Undefended Training

In Eq. (2.4), we only seek to minimize the expected loss $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$, which corresponds to maximizing the accuracy on the training set. This will happen when the model perfectly separates the classes into distinct regions separated by the decision boundary itself – there is no preference for one geometry of the decision boundary over another beyond this separation. Traditional optimization wisdom then dictates that regularization terms should be used to express a more specific preference. Note that when solving the standard learning problem Eq. (2.4) in practice, various kinds of regularization terms are typically introduced (e.g. weight decay, data augmentation, batch normalization, etc.). Moreover, the solution to the learning problem is computationally intractable, thus other factors such as model architecture, weight initialization, and the choice of optimization algorithm all influence the decision boundary of the trained model.

3.2 Adversarial Training: Smoothness & Orientation

Adversarial training can be used as an inductive bias towards decision boundaries that correspond to human-recognizability. As an intuitive illustration of this, Ilyas et al. (2019) shows that for a linear, binary classifier and two Gaussian distributed classes that are centered symmetrically (i.e. their means satisfy $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2$), adversarial training prefers a model whose decision boundary is approximately orthogonal to the means (notice that for a linear classifier, the adversarial direction $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$ will be orthogonal to the decision boundary).

As such, we want to better understand the kind of regularization that adversarial training provides. Towards this understanding, we analyze local approximations of two adversarial learning problems: TRADES (Zhang et al., 2019) and canonical adversarial training (Madry et al., 2017; Goodfellow et al., 2014). We identify smoothness regularization as the key addition in both cases. We select these two variants of adversarial learning because they form the foundation of a wide range of defenses against adversarial examples.

3.2.1 Canonical Adversarial Training

We begin with canonical adversarial training (Madry et al., 2017; Goodfellow et al., 2014), written in Eq. (2.5). A first-order Taylor expansion of the loss term gives

$$\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}+\boldsymbol{\delta})) = \mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \boldsymbol{\delta}^\top \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \mathcal{O}(\|\boldsymbol{\delta}\|_2^2).$$

For an ℓ_p threat model with $\mathcal{L}$ the cross-entropy loss, the solution $\boldsymbol{\delta}^\star$ to the inner maximization can be found by solving

$$\max_{\|\boldsymbol{\delta}\|_p \leq \epsilon} \boldsymbol{\delta}^\top \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) = \epsilon \left\|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right\|_q = \frac{\epsilon}{f(\boldsymbol{x})[y]} \left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[y]\right\|_q,$$

which is by definition the dual norm with $\frac{1}{p} + \frac{1}{q} = 1$. The solution is $\boldsymbol{\delta}^\star = \epsilon \, \|\varphi_{q-1}(\nabla_{\boldsymbol{x}}\mathcal{L})\|_q^{-1} \varphi_{q-1}(\nabla_{\boldsymbol{x}}\mathcal{L})$ where $\varphi_q(\boldsymbol{x}) \equiv \text{sign}(\boldsymbol{x}) \circ |\boldsymbol{x}|^q$ (see Appendix E for derivation), which for an ℓ_∞ adversary reduces to $\boldsymbol{\delta}^\star = \epsilon \, \text{sign}(\nabla_{\boldsymbol{x}}\mathcal{L})$ whence the FGSM method as derived by (Goodfellow et al., 2014). In other words, up to a first-order approximation, gradient penalties on the loss are simply the FGSM algorithm.

Plugging this approximation into Eq. (2.5) yields the input-gradient penalized learning program

$$\underset{f \in \mathcal{F}}{\text{minimize}} \; \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \beta\epsilon \, \|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\|_q \right], \tag{3.1}$$

where $\beta > 0$ is an additional regularization weight. Eq. (3.1) with $q = 2$ is used by Etmann et al. (2019); Finlay and Oberman (2019); Ross and Doshi-Velez (2017). The classic norm-inequalities $\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_\infty \leq \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2 \leq \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_1$ indicate that the ℓ_∞ adversary (whose dual norm is $\|\cdot\|_1$) has the strongest smoothing effect. Moreover, the $1/f(\boldsymbol{x})[y]$ term will discourage scores that are too large and too small, since we have the requirement that $\sum_y f(\boldsymbol{x})[y] = 1$ (this also happens with the TRADES penalty, see Appendix C).

In the form of Eq. (3.1), we can see that within sufficiently small regions around the training data (i.e. sufficiently small ϵ), the canonical adversarial training program Eq. (2.5) decomposes into two terms: $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$ which orients the decision boundary and $\|\nabla_{\boldsymbol{x}}\mathcal{L}\|_q$ which smooths the decision surface by penalizing the gradient and thereby the local Lipschitz constant around the data – this idea is encapsulated in Proposition 3.1, which we expound on in the following section.

This decomposition is the foundation of our hypothesis that two key ingredients for human-recognizability are smoothing of the model and orienting the decision boundary. This hypothesis makes intuitive sense, since the orientation of the decision boundary is what aligns gradients towards high density regions of the data distribution and smoothness of the model helps prevent first-order iterative procedures like Eq. (2.3) from getting stuck in local minima away from the data.

3.2.2 TRADES

We now consider the TRADES learning problem from Eq. (2.6), and show that it yields a similar decomposition into a smoothing term and an orientation of the decision boundary term. This culminates in Proposition 3.1, which characterizes the impact of (a second-order Taylor approximation of) TRADES on the local Lipschitz constant of the model. We also provide Lemma 3.1, which directly relates TRADES to other Jacobian-based smoothness regularizers.

Recall the TRADES loss from Eq. (2.6), $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \max_{\boldsymbol{\delta}\in B_\epsilon} \beta \, \mathcal{L}(f(\boldsymbol{x}), f(\boldsymbol{x}+\boldsymbol{\delta}))$. It is immediately clear that the loss term $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$ serves as the orientation term. We now discuss the maximization term, which intuitively is indeed a smoothing regularizer as claimed by (Zhang et al., 2019). We articulate this smoothing more formally.

Assuming an ℓ_p threat model with $p \geq 2$ and that $\mathcal{L}$ is the cross-entropy loss, we can connect the TRADES regularizer to more canonical Jacobian regularization via the Fisher Information Matrix (FIM). In particular, for the cross-entropy loss, we have an identify $\mathcal{L}(\boldsymbol{p}, \boldsymbol{q}) = H(\boldsymbol{p}) + \text{KL}(\boldsymbol{p} \,||\, \boldsymbol{q})$ in terms of Shannon entropy H and the KL-divergence. Minimizing the entropy regularization term $H(f(\boldsymbol{x}))$ pushes f towards confident class scores, which is redundant with the standard loss term $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$, so we drop this term from our approximation*. This means that the inner maximization reduces to $\max_{\boldsymbol{\delta}\in B_\epsilon} \beta \, \text{KL}(f(\boldsymbol{x}) \,||\, f(\boldsymbol{x}+\boldsymbol{\delta}))$. A second-order Taylor expansion yields

$$\text{KL}(f(\boldsymbol{x}) \,||\, f(\boldsymbol{x}+\boldsymbol{\delta})) = \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{F}_{\boldsymbol{x}} \boldsymbol{\delta} + \mathcal{O}(\|\boldsymbol{\delta}\|_2^3), \tag{3.2}$$

*In the implementation of TRADES provided by the authors, the inner maximization is with respect to the KL-divergence. See `https://github.com/yaodongyu/TRADES/blob/master/trades.py#L17-L84` for details.

where $\boldsymbol{F}_{\boldsymbol{x}} \equiv \mathbb{E}_y[(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})))\,(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})))^\top]$ is the Fisher Information Matrix (FIM). This FIM can be rewritten in terms of the Jacobian of f and another FIM: $\boldsymbol{F}_{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f \boldsymbol{J}(\boldsymbol{x})$, where

$$\boldsymbol{F}_f \equiv \mathbb{E}_y[(\nabla_{f(\boldsymbol{x})}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})))\,(\nabla_{f(\boldsymbol{x})}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})))^\top].$$

Supposing the ℓ_p threat model $\boldsymbol{\delta} \in B_\epsilon = \{\boldsymbol{x} \in \mathcal{X} : \|\boldsymbol{x}\|_p \leq \epsilon\}$ and that the threshold $\epsilon > 0$ is sufficiently small relative to the smoothness of f, then the higher order terms in Eq. (3.2) vanish and we have a reasonable approximation. We can then substitute this approximation into the inner maximization of Eq. (2.6), which gives a Jacobian type penalty

$$\underset{f\in\mathcal{F}}{\text{minimize}}\ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \frac{\beta\epsilon^2}{2}\left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\right\|_{p\hookrightarrow 2}^2\right]. \tag{3.3}$$

Note that here, $\|\cdot\|_{p\hookrightarrow q}$ indicates the vector-induced (p, q)-norm (see Appendix A). See Appendix B for the full derivation. This form is similar to the decomposition of the canonical adversarial training program Eq. (2.5) into a gradient regularization term Eq. (3.1). In the case of Eq. (3.3), however, we regularize the entire Jacobian matrix (through the FIM) rather than just one gradient direction corresponding to a particular label.

The following lemma provides bounds on the FIM penalty in Eq. (3.3) which are necessary for the proof of Proposition 3.1 below.

Lemma 3.1. *Let $\mathcal{L}$ be the cross-entropy loss and assume that $f : \mathcal{X} \to (0,1)^K$ is once-differentiable. If $p \geq 2$, then for each $\boldsymbol{x} \in \mathcal{X}$, the FIM $\boldsymbol{F}_{\boldsymbol{x}}$ satisfies*

$$\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2 \leq \lambda_{\max}(\boldsymbol{F}_{\boldsymbol{x}}) \leq \left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\right\|_{p\hookrightarrow 2}^2 \leq \left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_F^2. \tag{3.4}$$

Moreover,

$$\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_F \geq \|\boldsymbol{J}(\boldsymbol{x})\|_F. \tag{3.5}$$

See Appendix C for proof. It can be easily seen from Lemma 3.1 Eq. (3.4) that Eq. (3.3) is upper bounded by the following problem:

$$\underset{f\in\mathcal{F}}{\text{minimize}}\ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \frac{\beta\epsilon^2}{2}\left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\right\|_F^2\right], \tag{3.6}$$

which we use in our experiments in Section 4 below.

Now that we have Lemma 3.1, it is worth briefly pointing out explicitly how the FIM penalty of Eq. (3.3) is related to various robustness promoting regularizers suggested in prior literature.

- For an ℓ_2 adversary, the Eq. (3.3) penalty is the spectral norm $\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\|_{2\hookrightarrow 2} = \lambda_{\max}(\boldsymbol{F}_{\boldsymbol{x}})$, which was considered in Zhao et al. (2019); Miyato et al. (2018).
- From the upper bound problem Eq. (3.6), it can be seen Lemma 3.1 Eq. (3.5) in the appendix) that the $\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\|_F$ term also upper bounds the Jacobian penalty $\|\boldsymbol{J}(\boldsymbol{x})\|_F$, which was considered by Hoffman et al. (2019); Moosavi-Dezfooli et al. (2019); Jakubovitz and Giryes (2018).

To make the smoothing of Eq. (3.3) even more explicit, the following Proposition 3.1 shows that the FIM penalty corresponds to the smoothness of the classifier f via its local Lipschitz constant.

Proposition 3.1. *Assume an ℓ_p threat model with $p \geq 2$, let $\mathcal{L}$ is the cross-entropy loss, and assume that the classifier $f : \mathcal{X} \to (0,1)^K$ is once-differentiable. If $\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\|_{p\hookrightarrow 2} \leq \frac{L}{\sqrt{K}}$ over some $B \subseteq \mathcal{X}$, then*

(i) f *is* L*-Lipschitz on* B *and, moreover, each component* $f(\cdot)[k]$ *is* $L/\sqrt{K}$*-Lipschitz on* B*.*

(ii) If $f(\cdot) = \mathit{Softmax}(g(\cdot))$ *for some* $g : \mathcal{X} \to \mathbb{R}^K$*, then (i) holds for* g *as well.*

See Appendix C for proof. Note that since Eq. (3.6) is an upper bound on Eq. (3.3), Proposition 3.1 holds for a bounded $\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\|_F \leq \frac{L}{\sqrt{K}}$ as well.

In other words, in Eq. (3.3), the loss term $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$ orients the decision boundary of f while the FIM penalty biases towards f that are smooth around the data points. We can tune the smoothness by increasing the weight $\epsilon^2\beta$, which corresponds to larger localities around the data that should be smooth. This further informs our hypothesis from the previous section that the human-recognizability phenomenon requires two ingredients in order to manifest: orientation of the decision boundary and smoothness of the model itself.

We now experimentally investigate the impact of smoothness regularization on the human-recognizability of adversarial perturbations.

4. Experimental Results

Per our analysis in the previous section, we now present an experiment providing evidence that a the human-recognizability of adversarial perturbations occurs when orientation of the decision boundary are smoothness of the decision surface are combined. Our investigation in the prior section shows that this gives a model that is both smooth and has an adversarially aligned decision boundary.

To design our experiment, we group the defenses into three categories: minimax defenses (FGSM, Madry, and TRADES), approximate defenses (first order Eq. (3.1) and second order Eq. (3.6)), and undefended models (defensive distillation and standard). The minimax and approximate defenses makes use of both decision boundary orientation and decision surface smoothness in the sense of Proposition 3.1. We also train a model via TRADES with a high penalty weight so that smoothness entirely dominates orientation of the decision boundary. The first undefended model, defensive distillation, smooths the decision surface but does not adversarially orient the decision boundary since the underlying teacher model is trained via the standard learning program Eq. (2.4). The second undefended model, standard training, is neither smooth nor decision boundary oriented.

We then compare both the robustness to and human-recognizability of adversarial examples for each of the three categories. We find that when models are both smoothed and their decision boundaries are adversarially oriented, the models exhibit robustness and yield human-recognizable adversarial perturbations. We find that when the models are simply smoothed, as in defensive distillation, standard training, or highly penalized TRADES, the models do not yield human-recognizable adversarial perturbations.

4.1 Methods

For all experiments, we use a pre-activation ResNet18 and train on CIFAR-10. We augment the training data using random horizontal flipping and random cropping, normalizing each image according to the sample mean and standard deviation of the training data.

We initialize all models' weights using the same random seed and train for 15 epochs using a batch size of 128 and cyclic learning rate schedule with maximum rate 0.2 and minimum learning rate 0: the learning rate increases for 7 epochs and decreases for the remaining epochs. For the optimizer, we use SGD with momentum 0.9 and weight decay 2×10^{-4}.

For defensive distillation, we train a teacher model using softmax temperature $\tau = 100$. We then distill this model again with the same temperature value.

For training the minimax and approximate defenses, we use an ℓ_∞ adversary with $\epsilon = 8/255$ where applicable – note that the gradient and FIM penalties Eq. (3.1) and Eq. (3.3) use regularization weights

involving ϵ. For the gradient penalty approximate defenses, we choose a weight of $\beta = 64$, and we choose the norm order $q = 1$ (recall that this is the dual norm to the adversary's ℓ_p constraint) so as to correspond with an ℓ_∞-adversary. We implement this using double backpropagation in PyTorch (Paszke et al., 2019). For the FIM penalized defense, we use the Frobenius norm upper bound Eq. (3.6), which we compute using a minor adaptation of (Hoffman et al., 2019, Algorithm 1) (see Appendix D for more details). We set $\beta = 1$. For the FGSM defense, we use the improvements suggested by (Wong et al., 2020): random initialization of the perturbation $\boldsymbol{\delta}$ within the ℓ_p ϵ-ball and early stopping. For the Madry defense, we run each perturbation for 7 iterations with no random restarts. For the TRADES defense, we train models with both $\beta = 6$ and $\beta = 10,000$, again running each perturbation for 7 iterations with no random restarts.

To evaluate the robustness of the defenses, we generate a robust accuracy profile against an ℓ_∞ adversary (Figure 4.1) on the CIFAR-10 test-set. We use Madry PGD attacks Eq. (2.3) with an ℓ_∞ adversary using a stepsize $\alpha = 2/255$ for 50 iterations initialized randomly within the ϵ-ball; if an adversarial example is not found (i.e. the classifier's prediction is not changed), we restart the perturbation up to a maximum of 10 total restarts. We sweep the perturbation budget from 0/255, which gives the baseline accuracy of each model, and a maximum budget of 50/255, which is much larger than the perturbation budget that was used during training.

To generate the adversarial examples to evaluate human recognizability, we use the targeted variant of Madry PGD Eq. (2.3) (i.e. we select a class and approximate Eq. (2.2)), setting the perturbation budget to $\epsilon = \infty$ i.e. we do not project the iterates at all since we are not concerned with remaining imperceptible. We use a stepsize of $2/255$ and run each perturbation for 2,048 iterations.

All training and experiments are implemented in PyTorch (Paszke et al., 2019) and run on a single Tesla V100 GPU. The source code for all of our experiments is made available*.

4.2 Results

Figure 4.1 displays the robust accuracy profiles for the seven defenses we consider and replicates findings from prior work. Note that profiles can be grouped by the category of defense. The undefended models have the highest accuracy on an unperturbed test set, but the accuracy quickly falls to zero for small amounts of perturbation. The minimax defenses have a different, more robust pattern, where their unperturbed accuracy is lower, but the model's accuracy decreases slowly as the strength of the perturbation increases. For this model on these data, TRADES is the most robust, followed closely by PGD and FGSM. The approximate methods fall somewhere between the minimax and undefended models, except that the unperturbed accuracy of the approximate methods is the lowest of the three categories. Note that we do not include the TRADES $\beta = 10,000$ model in this robustness evaluation because it converged to a classifier that predicts every input as the class *ship* (the multi-class analogue of an all-ones classifier), which is completely robust but uninteresting for this evaluation.

Figure 4.2 shows adversarial perturbations generated against the defended models, organized by the category of the defense. Part (a) gives the initial, random starts of all of the adversarial examples in a given class. Parts (b) and (c) give the adversarial examples generated against undefended models, a model with standard training and defensive distillation, respectively. Parts (d) and (e) display the results for the approximate methods, gradient penalization, and FIM penalization respectively. Parts (f)-(h) display the results for the minimax methods, for FGSM, Madry, and TRADES respectively. Part (i) shows the TRADES model trained with a high penalty $\beta = 10,000$ to be an all-*ship* classifier.

Similarly to how the robustness accuracy profiles quantitatively distinguished between the three categories of defenses, the adversarial perturbations generated qualitatively distinguish between the three categories. First, the undefended models generated perturbations that are more perceptually similar to the initial

*https://github.com/cmu-sei/smoothness-and-recognizability

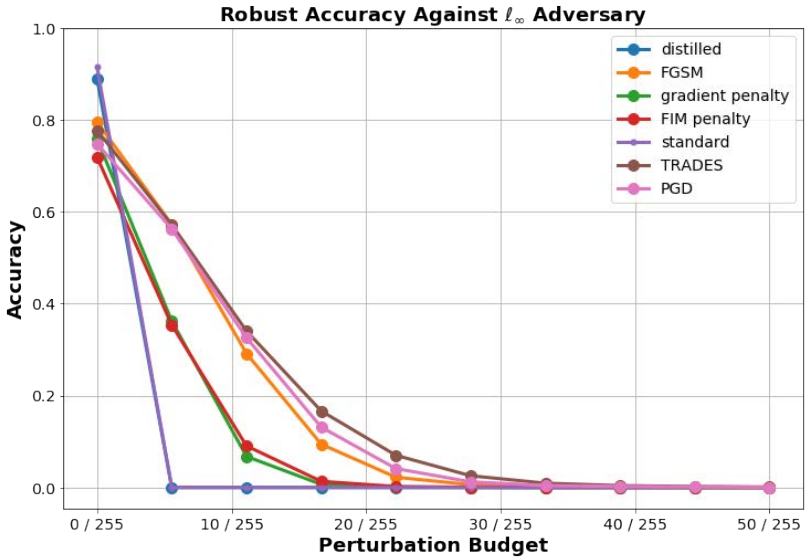


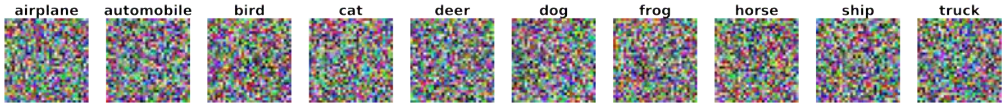
Figure 4.1: Robust accuracy profiles against an ℓ_∞ adversary for the models trained in Section 4.

point than they are to a human characterization of the target class. They are unrecognizable, although some of the standard model's perturbations are identifiable retrospectively given knowledge of the target class – the bird and boat perturbations, for example. Similarly, the distilled model has various perceptually interesting patterns that appear, but those patterns are nonetheless largely insufficient to identify the class without knowledge of the target label. It is worth pointing out, however, that the distilled model is distinctly more human-recognizable than the standard model. For defensive distillation, the original teacher model's decision boundary orientation percolates to the distilled model while the high temperature smooths the decision surface, thus providing some notion of the smoothness and orientation coupling that we have considered throughout this work.

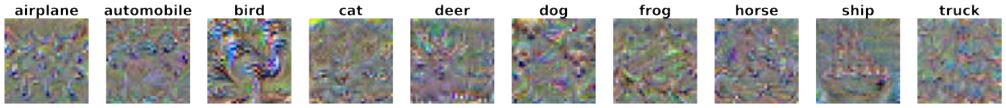
In contrast, the adversarial perturbations generated by the approximate defenses appear qualitatively different in kind from the starting images. These have patterns and colors that appear closer to exemplars of their classes relative to random noise, such that specific classes have recognizable features. For example, the gradient penalized method has recognizable exemplars of the various classes: automobile (tires and car shape), bird (pose, tail, and head), deer (shape, with antler-like structures), and ship (mast and water). Similarly, the FIM penalized method appears to have scattered versions of the gradient penalized examples.

Similarly, the minimax defenses appear qualitatively different in kind from both the starting images and the adversarial perturbations generated from the approximate methods. The minimax defenses begin to show adversarial examples that are recognizable perturbations of the target classes, while the approximate methods merely had recognizable features. For example the FGSM dog, the Madry PGD deer, and the Madry PGD truck are particularly striking. Interestingly, the TRADES model seems to have less human-recognizable perturbations despite being the strongest smoothness regularizer of the three minimax defenses. Since the approximate methods are also distinctly less human-recognizable than the FGSM and Madry PGD perturbations, it may be that decoupling the smoothness and orientation terms in the manner of TRADES Eq. (2.6) requires more careful tuning of β in order to match the same level of human-recognizability.

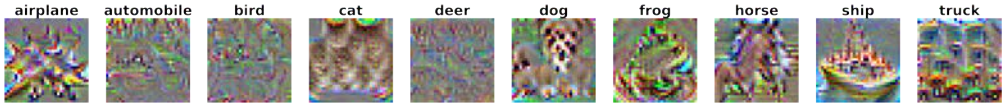
The highly penalized TRADES $\beta = 10,000$ model in Figure 4.2(i) shows that if smoothness is practically speaking the only objective, then human-recognizability is lost. This makes sense, since a classifier that only predicts one of the K classes will yield adversarial gradient directions that are meaningless and, moreover, close to zero.



(a) The initial points to adversarial perturbations in (b)-(i).

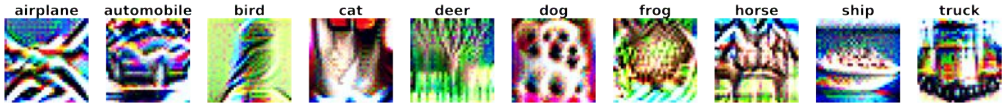


(b) Standard model.



(c) Distilled model $\tau = 100$ – distilled from the standard model in (a).

(d) Gradient penalized model, $\beta = 64$.

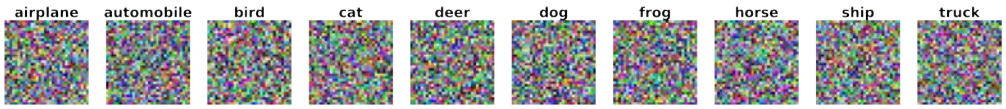


(e) FIM penalized model, $\beta = 1$.

(f) FGSM ℓ_∞ model.

(g) Madry PGD ℓ_∞ model.

(h) TRADES ℓ_∞ model, $\beta = 6$.



(i) TRADES ℓ_∞ model, $\beta = 10,000$.

Figure 4.2: Unconstrained adversarial perturbations for various pre-activation ResNet18 models. The robust accuracy profiles of each model are shown in Figure 4.1. Each image is the result of a distinct perturbation, which is initialized with random noise. In each column, we initialize the adversarial perturbations with the random noise shown in (a).

5. Discussion

Consider a situation in which a machine learning model is part of a high-stakes decision process that relies on sensitive training data. In cases like these, the high-stakes nature of the decision process suggests that the model needs to be robust to adversarial examples, but also the sensitivity of the data suggests that the model needs to also be protected from revealing certain types of information.

In this paper, we have demonstrated that certain state of the art defenses against adversarial attacks lead to human recognizable features in adversarial examples generated from those defended models. Other works like Kaur et al. (2019), Grathwohl et al. (2019), and Yang et al. (2019) have noticed the same phenomenon for other kinds of defenses, suggesting an incapability of the human-recognizability phenomenon. We do not attempt to develop a privacy attack in this work, but we do claim that this kind of human-recognizability present in the adversarial examples suggests that privacy attacks are possible. It is important for future research to pursue this line of inquiry to better understand the ways in which such privacy attacks may be possible and practically applicable.

For example, anecdotally, we have found that adversarial perturbations targeted towards the CIFAR-10 horse class often recover the presence of rider on the horses. Indeed, an inspection of the CIFAR-10 training set suggests that roughly 20% of the exemplars of that class have riders present. If this information was not known *a priori* to an attacker, they would be able to determine it if they had access to a robustly trained model's weights.

6. Conclusion

In this work, we investigated the human-recognizability phenomenon of adversarially trained deep image classifiers that has been noted in a myriad of other works. In particular, we investigated the human-recognizability of adversarial perturbations to models that were trained using both gradient and Jacobian-based regularization, which are fundamentally approximations of canonical adversarial training and TRADES adversarial training respectively. We identified that state-of-the-art adversarial training approaches involve both a smoothness term and a term that orients the decision boundary, and that this coupling appears to be a sufficient condition for the human-recognizability of large-ϵ adversarial perturbations, which fundamentally rely on local information vis-à-vis gradient directions. We demonstrated this sufficiency qualitatively using visualizations of the adversarial perturbations. Finally, we discussed some implications of the human-recognizability phenomenon in the privacy setting, suggesting a direction for future research to consider.

7. Acknowledgements

Copyright 2020 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MER-

CHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.

DM20-0917

References

Anil, C., Lucas, J., and Grosse, R. (2019). Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301.

Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

Carlini, N. and Wagner, D. (2016). Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., and Madry, A. (2019). Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*.

Etmann, C., Lunz, S., Maass, P., and Schönlieb, C.-B. (2019). On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*.

Finlay, C. and Oberman, A. M. (2019). Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468*.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333.

Gao, B. and Pavel, L. (2017). On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*.

Goodfellow, I. and Papernot, N. (2017). Is attacking machine learning easier than defending it? http://www.cleverhans.io/security/privacy/ml/2017/02/15/why-attacking-machine-learning-is-easier-than-defending-it.html. Accessed: 2020-10-05.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2019). Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hoffman, J., Roberts, D. A., and Yaida, S. (2019). Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136.

Jakubovitz, D. and Giryes, R. (2018). Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529.

Kaur, S., Cohen, J., and Lipton, Z. C. (2019). Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lin, A. T., Dukler, Y., Li, W., and Montúfar, G. (2019). Wasserstein diffusion tikhonov regularization. *arXiv preprint arXiv:1909.06860*.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, 16(3):31–57.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mejia, F. A., Gamble, P., Hampel-Arias, Z., Lomnitz, M., Lopatina, N., Tindall, L., and Barrios, M. A. (2019). Robust or Private? Adversarial Training Makes Models More Vulnerable to Privacy Attacks. *arXiv:1906.06449 [cs, stat]*. arXiv: 1906.06449.

Merikoski, J. K. and Kumar, R. (2004). Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes*, 4:150–159.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Moosavi-Dezfooli, S.-M., Fawzi, A., Uesato, J., and Frossard, P. (2019). Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9078–9086.

Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. P. (2018). Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414. IEEE.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Ross, A. S. and Doshi-Velez, F. (2017). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *arXiv preprint arXiv:1711.09404*.

Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*, pages 1262–1273.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Terzi, M., Achille, A., Maggipinto, M., and Susto, G. A. (2020). Adversarial training reduces information and improves transferability. *arXiv preprint arXiv:2007.11259*.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. (2020a). On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. (2020b). On Adaptive Attacks to Adversarial Example Defenses. *arXiv:2002.08347 [cs, stat]*. arXiv: 2002.08347.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2019). Robustness May Be at Odds with Accuracy. *arXiv:1805.12152 [cs, stat]*. arXiv: 1805.12152.

Wong, E., Rice, L., and Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*.

Wong, E., Schmidt, F. R., and Kolter, J. Z. (2019). Wasserstein adversarial examples via projected sinkhorn iterations. *arXiv preprint arXiv:1902.07906*.

Yang, Y., Zhang, G., Katabi, D., and Xu, Z. (2019). Me-net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971*.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*.

Zhao, C., Fletcher, P. T., Yu, M., Peng, Y., Zhang, G., and Shen, C. (2019). The adversarial attack and detection under the fisher information metric. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5869–5876.

A. Notation

Notation	Description
$\mathbb{R}$	The set of real numbers.
$\mathcal{X}$	The input domain of a classification model.
$a,\ \boldsymbol{a},\ \boldsymbol{A}$	Scalar, vector, matrix respectively.
$\boldsymbol{a}[i]$	The i-th entry of a vector $\boldsymbol{a}$.
$\boldsymbol{e}_i$	The i-th canonical basis vector i.e. the i-th column of the identity matrix $\boldsymbol{I}$ i.e. a one-hot vector with $\boldsymbol{e}_i[i] = 1$.
$\mathbb{E}$	Expectation operator.
$\boldsymbol{a} \circ \boldsymbol{b}$	Hadamard product (entrywise multiplication) between $\boldsymbol{a}$ and $\boldsymbol{b}$.
$\|\boldsymbol{A}\|_{p\hookrightarrow q} = \max_{\boldsymbol{x}}\ \|\boldsymbol{A}\boldsymbol{x}\|_q / \|\boldsymbol{x}\|_p$	Vector-induced matrix norm.
$\|\boldsymbol{A}\|_{p,q} = \left\|\begin{bmatrix}\|\boldsymbol{a}_1\|_q & \cdots & \|\boldsymbol{a}_m\|_q\end{bmatrix}\right\|_p$	Entrywise matrix norm i.e. p-norm of column-wise q-norm.
$\|\cdot\|_F$	Frobenius norm (equivalent to $\|\cdot\|_{2,2}$).
$\text{Softmax} : \mathbb{R}^K \to (0,1)^k$	The softmax function defined entrywise as $\text{Softmax}(\boldsymbol{a})[k] \equiv \exp(a_k) / \sum_i \exp(a_i)$.

B. Derivation of Eq. (3.3)

Let $f : \mathcal{X} \to (0,1)^K$ be a classifier over K-many classes, where the domain is denoted by $\mathcal{X} \subseteq \mathbb{R}^d$. Using a second-order Taylor expansion, it is straightforward to verify that

$$\text{KL}(f(\boldsymbol{x}) \,||\, f(\boldsymbol{x}+\boldsymbol{\delta})) = \mathbb{E}_y\left[\log \frac{f(\boldsymbol{x})[y]}{f(\boldsymbol{x}+\boldsymbol{\delta})[y]}\right] = \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{F}_{\boldsymbol{x}}\boldsymbol{\delta} + \mathcal{O}(\|\boldsymbol{\delta}\|_2^3), \tag{B.1}$$

where $\boldsymbol{F}_{\boldsymbol{x}}$ is the FIM

$$\begin{aligned}
\boldsymbol{F}_{\boldsymbol{x}} &\equiv \mathbb{E}_y\left[\left(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)\left(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)^\top\right] \\
&= \sum_y f(\boldsymbol{x})[y]\left(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)\left(\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)^\top \\
&= \sum_y f(\boldsymbol{x})[y]\ \left(\nabla_{\boldsymbol{x}}\log f(\boldsymbol{x})[y]\right)\left(\nabla_{\boldsymbol{x}}\log f(\boldsymbol{x})[y]\right)^\top \qquad \text{(for } \mathcal{L} \text{ cross-entropy)},
\end{aligned} \tag{B.2}$$

since the cross-entropy loss is $\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) = -\log f(\boldsymbol{x})[y]$. By defining another FIM

$$\begin{aligned}\boldsymbol{F}_f &\equiv \mathbb{E}_y\left[\left(\nabla_{f(\boldsymbol{x})}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)\left(\nabla_{f(\boldsymbol{x})}\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))\right)^\top\right] \\ &= \begin{bmatrix} f(\boldsymbol{x})[1] & & \\ & \ddots & \\ & & f(\boldsymbol{x})[K] \end{bmatrix}^{-1} \qquad \text{(for } \mathcal{L} \text{ cross-entropy)},\end{aligned} \tag{B.3}$$

we have by the chain rule that $\boldsymbol{F}_{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f \boldsymbol{J}(\boldsymbol{x})$, where $\boldsymbol{J}(\boldsymbol{x})$ is the Jacobian of the model f with respect to its input. Note that $\boldsymbol{F}_f \succeq 0$ regardless of the choices of $\mathcal{L}$ and f, thus $\operatorname{rank}(\boldsymbol{F}_{\boldsymbol{x}}) \leq K$ and hence $\boldsymbol{F}_{\boldsymbol{x}}$ is singular when the data dimension $d > K$. In fact, for the cross-entropy loss, we have that $\boldsymbol{F}_f \succ 0$.

Plugging the approximation Eq. (B.1) into the inner maximization of the TRADES program Eq. (2.6) yields

$$\begin{aligned}\max_{\boldsymbol{\delta}\in B_\epsilon} \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{F}_{\boldsymbol{x}}\boldsymbol{\delta} &= \max_{\|\boldsymbol{\delta}\|_p^2=1} \frac{\epsilon^2}{2}\boldsymbol{\delta}^\top \boldsymbol{F}_{\boldsymbol{x}}\boldsymbol{\delta} \\ &= \max_{\|\boldsymbol{\delta}\|_p^2=1} \frac{\epsilon^2}{2}\boldsymbol{\delta}^\top \boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{\delta} \\ &= \max_{\boldsymbol{\delta}} \frac{\epsilon^2}{2}\frac{\left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\boldsymbol{\delta}\right\|_2^2}{\|\boldsymbol{\delta}\|_p^2} \\ &= \frac{\epsilon^2}{2}\left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\right\|_{p\hookrightarrow 2}^2,\end{aligned}$$

since $\boldsymbol{F}_f \succ 0$ allows the square root $\boldsymbol{F}_{\boldsymbol{x}}^{1/2}$, whence TRADES is approximated as

$$\underset{f\in\mathcal{F}}{\text{minimize}}\ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x})) + \frac{\beta\epsilon^2}{2}\left\|\boldsymbol{F}_f^{1/2}\boldsymbol{J}(\boldsymbol{x})\right\|_{p\hookrightarrow 2}^2\right]. \tag{B.4}$$

C. Proof of Proposition 3.1

Recall the FIMs $\boldsymbol{F}_{\boldsymbol{x}}$ Eq. (B.2) and $\boldsymbol{F}_f$ Eq. (B.3), as well as the Jacobian of f at $\boldsymbol{x}$, denoted $\boldsymbol{J}(\boldsymbol{x})$. Now, to prove Proposition 3.1, we will need two intermediary lemmas. The first is Lemma C.1, which provides some useful spectral bounds on the FIM $\boldsymbol{F}_{\boldsymbol{x}}$.

Lemma C.1 (Section 3.2.2 Lemma 3.1)**.** *Let $\mathcal{L}$ be the cross-entropy loss and assume that $f : \mathcal{X} \to (0,1)^K$ is once-differentiable. For each $\boldsymbol{x} \in \mathcal{X}$, the FIM $\boldsymbol{F}_{\boldsymbol{x}}$ satisfies*

$$\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2 \leq \lambda_{\max}(\boldsymbol{F}_{\boldsymbol{x}}) \leq \left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_F^2. \tag{C.1}$$

Moreover,

$$\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_F \geq \|\boldsymbol{J}(\boldsymbol{x})\|_F. \tag{C.2}$$

Proof. Denote the k-th row of $\boldsymbol{J}(\boldsymbol{x})$ as

$$\boldsymbol{g}_k \equiv -\nabla_{\boldsymbol{x}} \log f(\boldsymbol{x})[k] = -\frac{1}{f(\boldsymbol{x})[k]}\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k],$$

noting that $f(\boldsymbol{x})[k] \in (0,1)$ for any $\boldsymbol{x} \in \mathbb{X}$, and observe subsequently that $\boldsymbol{F}_{\boldsymbol{x}} = \sum_k f(\boldsymbol{x})[k] \boldsymbol{g}_k \boldsymbol{g}_k^\top \succeq 0$. Then, by (Merikoski and Kumar, 2004, Theorem 1), we have the lower bound

$$\begin{aligned}
\lambda_{\max}(\boldsymbol{F}_{\boldsymbol{x}}) &\geq \max_k \frac{1}{f(\boldsymbol{x})[k]} \lambda_{\max}\left(\boldsymbol{g}_k \boldsymbol{g}_k^\top\right) + \lambda_{\min}\left(\sum_{k \neq k^\star} f(\boldsymbol{x})[k]\, \boldsymbol{g}_k \boldsymbol{g}_k^\top\right) \\
&\geq \max_k \frac{1}{f(\boldsymbol{x})[k]} \left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k]\right\|_2^2 \\
&= \left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2
\end{aligned}$$

by definition of the entrywise $\|\cdot\|_{2,\infty}$ norm, which is the ℓ_∞-norm of the column-wise ℓ_2-norms. Note that $k^\star$ is the index on which the maximum is achieved and the second inequality follows since $\operatorname{rank}(\sum_{k \neq k^\star} \boldsymbol{g}_k \boldsymbol{g}_k^\top) \leq K - 1 < d$. To complete Eq. (C.1), observe that the upper bound is a simple application of the Cauchy-Schwarz inequality.
To see Eq. (C.2), observe

$$\begin{aligned}
\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_F^2 = \operatorname{tr}\left(\boldsymbol{F}_f \boldsymbol{J}(\boldsymbol{x}) \boldsymbol{J}(\boldsymbol{x})^\top\right) &= \sum_k \frac{1}{f(\boldsymbol{x})[k]} \left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k]\right\|_2^2 \\
&\geq \sum_k \left\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k]\right\|_2^2 = \left\|\boldsymbol{J}(\boldsymbol{x})\right\|_F^2,
\end{aligned}$$

where the inequality is because $f(\boldsymbol{x})[k] \in (0,1)$, thus completing the proof. □

The second lemma inequality on the vector-induced operator norms of the kind found in Eq. (3.3) for different choices of order p. This will allow us to state our result for ℓ_p threat models with $p \geq 2$, which is reasonable since $p \geq 2$ are the most common ℓ_p threat models for adversarial training.

Lemma C.2. *For $1 \leq p \leq q$, $r \geq 1$, and real matrix $\boldsymbol{A}$,*

$$\|\boldsymbol{A}\|_{p \hookrightarrow r} \leq \|\boldsymbol{A}\|_{q \hookrightarrow r}.$$

Proof. Let $\boldsymbol{x}^\star$ be the global maximizer of

$$\|\boldsymbol{A}\|_{p \hookrightarrow r} = \max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_r}{\|\boldsymbol{x}\|_p}.$$

Then

$$\frac{\|\boldsymbol{A}\boldsymbol{x}^\star\|_r}{\|\boldsymbol{x}^\star\|_p} \leq \frac{\|\boldsymbol{A}\boldsymbol{x}^\star\|_r}{\|\boldsymbol{x}^\star\|_q} \leq \max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_r}{\|\boldsymbol{x}\|_q} = \|\boldsymbol{A}\|_{q \hookrightarrow r}.$$

□

We are now ready to prove Proposition 3.1.

Proposition C.1 (Proposition 3.1)**.** *Assume an ℓ_p threat model with $p \geq 2$, let $\mathcal{L}$ is the cross-entropy loss, and assume that the classifier $f : \mathcal{X} \to (0,1)^K$ is once-differentiable. If $\|\boldsymbol{F}_f^{1/2} \boldsymbol{J}(\boldsymbol{x})\|_{p \hookrightarrow 2} \leq \frac{L}{\sqrt{K}}$ over some $B \subseteq \mathcal{X}$, then*

(i) f is L-Lipschitz on B and, moreover, each component $f(\cdot)[k]$ is $L/\sqrt{K}$-Lipschitz on B.

(ii) If $f(\cdot) = \mathit{Softmax}(g(\cdot))$ for some $g : \mathcal{X} \to \mathbb{R}^K$, then (i) holds for g as well.

Proof. Recall that f, being once-differentiable, is L-Lipschitz on B if and only if its Jacobian is bounded: $\|\boldsymbol{J}(\boldsymbol{x})\|_2 \leq L$ for all $\boldsymbol{x} \in B$. We will show this condition by straightforward application of ℓ_p-norm inequalities and Lemma C.1.
First, observe that

$$\begin{aligned}\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2 &= \max_k \frac{1}{f(\boldsymbol{x})[k]} \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k]\|_2^2 \\ &\geq \max_k \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})[k]\|_2^2 \\ &= \left\|\boldsymbol{J}(\boldsymbol{x})^\top\right\|_{2,\infty}^2,\end{aligned}$$

since $f(\boldsymbol{x})[k] \in (0,1)$ for any $\boldsymbol{x} \in \mathcal{X}$. It then follows that

$$\begin{aligned}\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2 &\geq \left\|\boldsymbol{J}(\boldsymbol{x})^\top\right\|_{2,\infty}^2 \\ &\geq \frac{1}{K} \|\boldsymbol{J}(\boldsymbol{x})\|_F^2 \\ &\geq \frac{1}{K} \|\boldsymbol{J}(\boldsymbol{x})\|_2^2 .\end{aligned} \tag{C.3}$$

Now, Lemma C.1 gives

$$\begin{aligned}\left\|\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2}\right\|_{2,\infty}^2 &\leq \lambda_{\max}(\boldsymbol{F}_{\boldsymbol{x}}) = \lambda_{\max}\left(\boldsymbol{J}(\boldsymbol{x})^\top \boldsymbol{F}_f^{1/2} \boldsymbol{F}_f^{1/2} \boldsymbol{J}(\boldsymbol{x})\right) \\ &= \left\|\boldsymbol{F}_f^{1/2} \boldsymbol{J}(\boldsymbol{x})\right\|_{2\hookrightarrow 2}^2 \\ &\leq \left\|\boldsymbol{F}_f^{1/2} \boldsymbol{J}(\boldsymbol{x})\right\|_{p\hookrightarrow 2}^2 \\ &\leq \frac{L^2}{K},\end{aligned} \tag{C.4}$$

where the second inequality is due to Lemma C.2. Combining (C.3) and (C.4) gives $\|\boldsymbol{J}(\boldsymbol{x})\|_2 \leq L$, and it immediately follows that f is L-Lipschitz on B and thus each $f(\cdot)[k]$ is also $L/\sqrt{K}$-Lipschitz on B. This proves (i).
Property (ii) follows immediately from (i) since $\text{Softmax}(\cdot)$ is a 1-Lipschitz function (Gao and Pavel, 2017, Proposition 4). □

D. Computing the Frobenius Norm FIM Penalty

The Frobenius Norm FIM penalty in Eq. (3.6) involves the term $\|\boldsymbol{F}_f^{1/2} \boldsymbol{J}(\boldsymbol{x})\|_F^2$ that must be computed for each iteration of training. The Jacobian, however, can be difficult to work with in practice for high dimensional datasets with many classes. As such, an approximation algorithm is needed. Such an algorithm is proposed in (Hoffman et al., 2019, Algorithm 1) for computing $\|\boldsymbol{J}(\boldsymbol{x})\|_F$, which is simple to adapt to our case involving the FIM by scaling the Jacobian-vector product by $\boldsymbol{F}_f^{1/2}$.

E. Deriving The ℓ_p FGSM Algorithm

In this section, we derive a generic ℓ_p-adversary version of the FGSM algorithm (Goodfellow et al., 2014), starting at the maximization problem that follows from a first-order Taylor expansion of the canonical adversarial learning program Eq. (2.5). Specifically, this maximization is

$$\max_{\|\boldsymbol{\delta}\|_p=\epsilon} \boldsymbol{\delta}^\top \nabla_{\boldsymbol{x}} \mathcal{L},$$

where we use the shorthand $\nabla_{\boldsymbol{x}} \mathcal{L} = \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{e}_y, f(\boldsymbol{x}))$. The Lagrangian is $\mathfrak{L}(\boldsymbol{\delta}, \lambda) = \boldsymbol{\delta}^\top \nabla_{\boldsymbol{x}} \mathcal{L} - \lambda \left(\|\boldsymbol{x}\|_p - \epsilon \right)$ and hence by the KKT conditions,

$$\nabla_{\boldsymbol{\delta}} \mathfrak{L} = \nabla_{\boldsymbol{x}} \mathcal{L} - \lambda \|\boldsymbol{\delta}\|_p^{1-p} \varphi_{p-1}(\boldsymbol{\delta}) = \mathbf{0} \quad \Leftrightarrow \quad \nabla_{\boldsymbol{x}} \mathcal{L} = \lambda \|\boldsymbol{\delta}\|_p^{1-p} \varphi_{p-1}(\boldsymbol{\delta}),$$

where $\varphi_p(\boldsymbol{\delta}) \equiv \text{sign}(\boldsymbol{\delta}) \circ |\boldsymbol{\delta}|^p$ with $\circ$ the Hadamard product and the absolute value and power taken entry-wise. Now, let $q \equiv p/(p-1)$, noticing, then, that $(p-1)(q-1) = 1$ and thus $\varphi_{q-1}(\varphi_{p-1}(\boldsymbol{\delta})) = \delta$. It follows that

$$\varphi_{q-1}\left(\nabla_{\boldsymbol{x}} \mathcal{L}\right) = \left(\frac{\lambda}{\|\boldsymbol{\delta}\|_p^{p-1}} \right)^{q-1} \varphi_{q-1}(\varphi_{p-1}(\boldsymbol{\delta})) = \frac{\lambda^{q-1}}{\|\boldsymbol{\delta}\|_p} \boldsymbol{\delta}.$$

Rearranging this equation, we have

$$\frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_p} = \frac{1}{\lambda^{q-1}} \varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L}) \quad \Rightarrow \quad \lambda^{q-1} = \left\| \varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L}) \right\|_p$$

but, since feasibility dictates that $\|\boldsymbol{\delta}\|_p = \epsilon$, we have

$$\boldsymbol{\delta} = \frac{\epsilon}{\lambda^{q-1}} \varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L}) = \epsilon \frac{\varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L})}{\left\| \varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L}) \right\|_p}.$$

It is easy to see that in the $p = 2$ case, we obtain $\boldsymbol{\delta}^\star = \epsilon \|\nabla_{\boldsymbol{x}} \mathcal{L}\|_2^{-1} \nabla_{\boldsymbol{x}} \mathcal{L}$ and in the $p = \infty$ case, we have $\boldsymbol{\delta}^\star = \epsilon \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L})$ – precisely the FGSM update. This means that we perturb the input as

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \frac{\varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L})}{\left\| \varphi_{q-1}(\nabla_{\boldsymbol{x}} \mathcal{L}) \right\|_p}.$$

F. A Connection Between Model Inversion & Adversarial Perturbation

In this section, we show that model inversion is equivalent to an unbounded adversarial perturbation in the sense that the algorithms for performing both types of attacks spawn from the same optimization problem.

The model inversion attack proposed in (Fredrikson et al., 2015) can be written as a nonconvex program

$$\underset{\boldsymbol{x} \in C}{\text{minimize}} \; 1 - f(\boldsymbol{x})[k] + \Omega(\boldsymbol{x}), \tag{F.1}$$

where C is a constraint set, $\Omega : \mathcal{X} \to \mathbb{R}$ is a penalty function, and $f : \mathcal{X} \to (0,1)^K$ is the classifier over K-many classes. In (Fredrikson et al., 2015), $\Omega(\boldsymbol{x}) = 0$ and the constraint set is $C = \mathcal{X}$ or the codomain of a denoising autoencoder. The solution to Eq. (F.1) can be approximated via PGD.

We now show that Eq. (F.1) as used by (Fredrikson et al., 2015) can be thought of as an unbounded adversarial perturbation, which in turn provides an algorithmic connection between model inversion and adversarial perturbations. Note that the solution to Eq. (F.2) below is a targeted adversarial attack Eq. (2.2).

Proposition F.1. *Let $\Omega(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$. Then (F.1) is equivalent to*

$$\underset{\boldsymbol{\delta} \in B}{\textit{minimize}} \; L\left(\boldsymbol{e}_k, f(\boldsymbol{x} + \boldsymbol{\delta})\right) \tag{F.2}$$

for a properly chosen B.

Proof. Since $f(\boldsymbol{x})[k] \in [0, 1]$, it is equivalent to solve

$$\underset{\boldsymbol{x} \in C}{\text{maximize}} \;\; f(\boldsymbol{x})[k].$$

Now, defining $L : [0, 1]^K \to \mathbb{R}$ as the cross-entropy loss $L(\boldsymbol{p}, \boldsymbol{q}) \equiv -\sum_k p_k \log q_k$, solving

$$\underset{\boldsymbol{x} \in C}{\text{minimize}} \;\; L\left(\boldsymbol{e}_k, f(\boldsymbol{x})\right),$$

where $\boldsymbol{e}_k$ is a one-hot vector corresponding to the k-th class, is equivalent to solving (F.1) since $L(\boldsymbol{e}_k, f(\boldsymbol{x})) = -\log f(\boldsymbol{x})[k]$ is a monotonically decreasing transformation of the previous objective, thereby converting local maxima to local minima and local minima to local maxima. Defining $B(\boldsymbol{x}) \equiv \{\boldsymbol{x}' - \boldsymbol{x} : \boldsymbol{x}' \in C\}$ as the feasible region yields the desired program (F.2). □