A Manifold View of Adversarial Risk

Wenjia Zhang
Rutgers University

Yikai Zhang
Morgan Stanley

Xiaolin Hu
Stony Brook University

Mayank Goswami
Queens College of CUNY

Chao Chen
Stony Brook University

Dimitris Metaxas
Rutgers University

Abstract

The adversarial risk of a machine learning model has been widely studied. Most previous works assume that the data lies in the whole ambient space. We propose to take a new angle and take the manifold assumption into consideration. Assuming data lies in a manifold, we investigate two new types of adversarial risk, the normal adversarial risk due to perturbation along normal direction, and the in-manifold adversarial risk due to perturbation within the manifold. We prove that the classic adversarial risk can be bounded from both sides using the normal and in-manifold adversarial risks. We also show with a surprisingly pessimistic case that the standard adversarial risk can be nonzero even when both normal and in-manifold risks are zero. We finalize the paper with empirical studies supporting our theoretical results. Our results suggest the possibility of improving the robustness of a classifier by only focusing on the normal adversarial risk.

1 INTRODUCTION

Machine learning (ML) algorithms have achieved astounding success in multiple domains such as computer vision (Krizhevsky et al., 2012; He et al., 2016), natural language processing (Wu et al., 2016; Vaswani et al., 2017), and robotics (Levine and Abbeel, 2014; Nagabandi et al., 2018). These models perform well on massive datasets but are also vulnerable to small perturbations on the input examples. Adding a slight

and visually unrecognizable perturbation to an input image can completely change the model's prediction. Many works have been published focusing on such adversarial attacks (Szegedy et al., 2013; Carlini and Wagner, 2017; Madry et al., 2017). To improve the robustness of these models, various defense methods have been proposed (Madry et al., 2017; Zhang et al., 2019; Shafahi et al., 2019). These methods mostly focus on minimizing the *adversarial risk*, i.e., the risk of a classifier when an adversary is allowed to perturb any data with an oracle.

Despite the progress in improving the robustness of models, it has been observed that compared with a standard classifier, a robust classifier often has a lower accuracy on the original data. The accuracy of a model can be compromised when one optimizes its adversarial risk. This phenomenon is called *the trade-off between robustness and accuracy*. Su et al. (2018) observed this trade-off effect on a large number of commonly used model architectures. They concluded that there is a linear negative correlation between the logarithm of accuracy and adversarial risk. Tsipras et al. (2018) proved that adversarial risk is inevitable for any classifier with a non-zero error rate. Zhang et al. (2019) decomposed the adversarial risk into the summation of standard error and boundary error. The decomposition provides the opportunity to explicitly control the trade-off. They also proposed a regularizer to balance the trade-off by maximizing the boundary margin.

In this paper, we investigate the adversarial risk and the robustness-accuracy trade-off through a new angle. We follow the classic manifold assumption, i.e., data are living in a low dimensional manifold embedded in the input space (Rifai et al., 2011; Cayton, 2005; Narayanan and Mitter, 2010; Niyogi et al., 2008).

Based on this assumption, we analyze the adversarial risk with regard to adversarial perturbations within the manifold and normal to the manifold. By restricting

to in-manifold and normal perturbations, we define the *in-manifold adversarial risk* and *normal adversarial risk*. Using these new risks, together with the standard risk, we prove an upper bound and a lower bound for the adversarial risk. We also show that the bound is tight by constructing a pessimistic case. We validate our theoretical results using synthetic experiments.

Our study sheds light on a new aspect of the robustness-accuracy trade-off. Through the decomposition into in-manifold and normal adversarial risks, we might find an extra margin to exploit without confronting the trade-off. Future work will include developing normal adversarial training algorithms for real-world datasets.

1.1 Related Works

Robustness-accuracy Trade-off There are several works studying the trade-off between robustness and accuracy (Tsipras et al., 2018; Su et al., 2018; Zhang et al., 2019; Dohmatob, 2019). The basic question is whether the trade-off actually exists. i.e. is there a classifier that is both accurate and robust? Empirical and theoretical proofs showed that actual trade-off does exist even in the infinite data limit (Tsipras et al., 2018; Su et al., 2018; Zhang et al., 2019). Dohmatob (2019) showed that a high accuracy model can inevitably be fooled by the adversarial attack. Zhang et al. (2019) gave examples showing that the Bayes optimal classifier may not be robust.

However, some works have different views on this trade-off or even its existence. In contrast to the idea that the trade-off is unavoidable, these works argued that a lack of sufficient optimization methods (Awasthi et al., 2019; Rice et al., 2020; Shaham et al., 2018) or better network architecture (Guo et al., 2020; Fawzi et al., 2018) causes the drop in accuracy, instead of the increase in robustness. Yang et al. (2020) showed the existence of both robust and accurate classifiers and argued that the trade-off is influenced by the training algorithm to optimize the model. They investigated distributionally separated dataset and claimed that the gap between robustness and accuracy arises from the lack of a training method that imposes local Lipschitzness on the classifier. Remarkably, in (Gowal et al., 2020; Raghunathan et al., 2020; Carmon et al., 2019), it was shown that with certain augmentation of the dataset, one may be able to obtain a model that is both accurate and robust.

Manifold Assumption One important line of research focuses on the manifold assumption on the data distribution. This assumption suggests that observed data is distributed on a low dimensional manifold (Rifai et al., 2011; Cayton, 2005; Narayanan and Mitter, 2010) and there exists a mapping that embeds the low

dimension manifold in some higher dimension space. Traditional manifold learning methods (Tenenbaum et al., 2000; Saul and Roweis, 2003) try to recover the embedding by assuming the mapping preserves certain properties like distances or local angles. Following this assumption, on the topic of robustness, Tanay and Griffin (2016) showed the existence of adversarial attack on the flat manifold with linear classification boundary. It was proved later in Gilmer et al. (2018) that in-manifold adversarial examples exist. They stated that high dimension data is highly sensitive to l_2 perturbations and pointed out the nature of adversarial is the issue with potential decision boundary. Later, Stutz et al. (2019) showed that with the manifold assumption, regular robustness is correlated with in-manifold adversarial examples, and therefore, accuracy and robustness may not be contradictory goals. Further discussion (Xie et al., 2020) even suggested that adding adversarial examples in the training process can improve the accuracy of the model. Lin et al. (2020) used perturbation within a latent space to approximate in-manifold perturbation. To the best of our knowledge, no existing work discussed normal perturbation and normal adversarial risk as we do. We are also unaware of any theoretical results proving upper/lower bounds for adversarial risk in the manifold setting.

We also note a classic manifold reconstruction problem, i.e., reconstructing a d-dimensional manifold given a set of points sampled from the manifold. A large group of classical algorithms (Edelsbrunner and Shah, 1994; Dey and Goswami, 2006; Niyogi et al., 2008) are provably good, i.e., they give a guarantee of reproducing the manifold topology with a sufficiently large number of sample points.

2 MANIFOLD BASED RISK DECOMPOSITION

In this section, we state our main theoretical result 3, which decomposes the adversarial risk into appropriately defined normal and in-manifold or tangential risks. We first define these quantities and set up basic notation, with the main theorem following in Section 2.3. For the sake of simplicity, we describe our main theorem in the setting of binary $\{-1, 1\}$ labels.

2.1 Data Manifold

Let $(\mathbb{R}^D, ||.||)$ denote the D dimensional Euclidean space with ℓ_2-norm. For $x \in \mathbb{R}^D$, $B_\epsilon(x)$ be the open ball of radius r in $\mathbb{R}^D$ with center at x. For a set $A \subset \mathbb{R}^D$, define $B_\epsilon(A) = \{y : \exists x \in A, d(x, y) < \epsilon\}$.

Let $\mathcal{M} \subset \mathbb{R}^D$ be a d-dimensional compact smooth manifold embedded in $\mathbb{R}^D$. Thus for any $x \in M$ there is
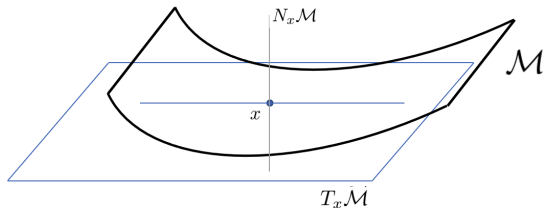
Figure 1: Tangential and Normal Space

a corresponding coordinate chart (U, g) where $U \ni x$ is a open set of $\mathcal{M}$ and g is a homeomorphism from U to a subset of $\mathbb{R}^d$. For $x \in \mathcal{M}$, we let $T_x\mathcal{M}$ and $N_x\mathcal{M}$ denote the tangent and normal spaces at x. Intuitively, the tangent space $T_x\mathcal{M}$ is the space of tangent directions, or equivalence classes of curves in $\mathcal{M}$ passing through x, with two curves considered equivalent if they are tangent at x. The normal space $N_x\mathcal{M}$ is the set of vectors in $\mathbb{R}^D$ that are orthogonal to any vector in $T_x\mathcal{M}$. Since $\mathcal{M}$ is a smooth d-manifold, $T_x\mathcal{M}$ and $N_x\mathcal{M}$ are d and $D - d$ dimensional vector spaces, respectively. See Figure 2.1. For detailed definitions, we refer the reader to Bredon (2013).

We assume that the data and (binary) label pairs are drawn from $\mathcal{M} \times \{-1, 1\}$ according to some unknown distribution $p(x, y)$. Note that $\mathcal{M}$ is unknown. A score function $f(x)$ is a continuous function from $\mathbb{R}^D$ to $[0, 1]$. We denote by $\mathbb{1}(A)$ the indicator function of the event A that is 1 if A occurs and 0 if A does not occur, and will use it to represent the 0-1 loss.

2.2 Robustness and Risk

Given data from $\mathcal{M} \times \{-1, 1\}$ drawn according to p and a classifier f on $\mathbb{R}^D$, we define three types of risks. The first, adversarial risk, has been extensively studied in machine learning literature:

Definition 1 (Adversarial Risk). *Given $\epsilon > 0$, define the adversarial risk of classifier f with budget ϵ to be*

$$R_{adv}(f, \epsilon) := \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(\exists x' \in B_\epsilon(x) : f(x')y \leq 0)$$

Notice that $B_\epsilon(x)$ is the open ball around x in $\mathbb{R}^D$ (the ambient space).

We next define risk that is concerned only with in-manifold perturbations. Previously, Gilmer et al. (2018) and Stutz et al. (2019) showed that there exist in-manifold adversarial examples, and empirically demonstrated that in-manifold perturbations are a cause of the standard classification error. Therefore, in the following, we define the in-manifold perturbations and in-manifold adversarial risk.

Definition 2 (In-manifold Risk). *Given $\epsilon > 0$, the in-manifold adversarial perturbation for classifier f with budget ϵ is the set*

$$B_\epsilon^{in}(x) := \{x' \in \mathcal{M} : \|x - x'\| \leq \epsilon\}$$

The in-manifold adversarial risk is

$$R_{adv}^{in}(f, \epsilon) := \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(\exists x' \in B_\epsilon^{in}(x) : f(x')y \leq 0)$$

We remark that while the above perturbation is on the manifold, in many manifold-based defense algorithms use generative models to estimate the homeomorphism (the manifold chart) $z = g(x)$ for real-world data. Therefore, instead of in-manifold perturbation, one can also use an equivalent η-budget perturbation in the *latent* space. However, for our purposes, the in-manifold definition will be more convenient to use. Lastly, we define the *normal* risk:

Definition 3 (Normal Adversarial Risk). *Given $\epsilon > 0$, the normal adversarial perturbation for classifier f with budget ϵ is be the set*

$$B_\epsilon^{nor}(x) := \{x' : x' - x \in N_x\mathcal{M}, \|x - x'\| \leq \epsilon\}$$

Define the normal adversarial risk as

$$R_{adv}^{nor}(f, \epsilon) := \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(\exists x' \neq x \in B_\epsilon^{nor}(x) : f(x')y \leq 0)$$

Notice that the normal adversarial risk is non-zero if there is an adversarial perturbation $x' \neq x$ in the normal direction at x. Finally, we have the usual *standard risk*: $R_{std}(f) := \mathbb{E}_{(x,y)\sim p} \mathbb{1}(f(x)y \leq 0)$.

2.3 Main Result: Decomposition of Risk

In this section, we state our main result that decomposes the adversarial risk into its tangential and normal components. Our theorem will require a mild assumption on the decision boundary $DB(f)$ of the classifier f, i.e., the set of points x where $f(x) = 0$.

Assumption [A]: For all $x \in DB(f)$ and all neighborhoods $U \ni x$ containing x, there exist points x_0 and x_1 in U such that $f(x_0) < 0$ and $f(x_1) > 0$.

This assumption states that a point that is difficult to classify by f has points of both labels in any given neighborhood around it. In particular, this means that the decision boundary does not contain an open set. We remark that both Assumption A and the continuity requirement for the score function f are implicit in previous decomposition results like Equation 1 in Zhang et al. (2019). Without Assumption A, the "neighborhood" of the decision boundary in Zhang et al. (2019) will not contain the decision boundary, and it is easy to give a counterexample to Equation 1 in Zhang et al. (2019) if f if not continuous.

Our decomposition result will decompose the adversarial risk into the normal and tangential directions: however, as we will show, an "extra term" appears, which we define next:

Definition 4 (NNR Nearby-Normal-Risk). *Fix $\epsilon > 0$. Denote by $A(x, y)$ the event that $\forall x' \in B_\epsilon^{nor}(x), f(x')y > 0$, i.e., the normal adversarial risk of x is zero.*

Denote by $B(x, y)$ the event that

$$\exists x' \in B_{2\epsilon}^{in}(x) : (\exists z \in B_\epsilon^{nor}(x') : f(z)f(x') \leq 0),$$

i.e., x has a point x' near it such that x' has non-zero normal adversarial risk.

Denote by $C(x, y)$ the event $\forall x' \in B_{2\epsilon}^{in}(x), f(x')y > 0$, i.e., x has no adversarial perturbation in the manifold within distance 2ϵ.

The Nearby-Normal-Risk (denoted as NNR) of f with budget ϵ is defined to be

$$\mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(A(x,y) \wedge B(x,y) \wedge C(x,y)),$$

where $\wedge$ denotes "and".

We are now in a position to state our main result.

Theorem 1. *[Risk Decomposition] Let $\mathcal{M}$ be a smooth compact manifold in $\mathbb{R}^D$, and let data be drawn from $\mathcal{M} \times \{-1, 1\}$ according to some distribution p. There exists a $\Delta > 0$ depending only on $\mathcal{M}$ such that the following statements hold for any $\epsilon < \Delta$. For any score function f satisfying assumption A,*

(i)

$$\begin{aligned} R_{adv}(f, \epsilon) &\leq R_{std}(f) + R_{adv}^{nor}(f, \epsilon) + R_{adv}^{in}(f, 2\epsilon) \\ &+ NNR(f, \epsilon). \end{aligned} \quad (1)$$

(ii) If $R_{adv}^{nor}(f, \epsilon) = 0$, then

$$R_{adv}(f, \epsilon) \leq R_{std}(f) + R_{adv}^{in}(f, 2\epsilon)$$

Remark:

1. The first result decomposes the adversarial risk into the standard risk, the normal adversarial risk, the in-manifold risk, and an "extra term" — the Nearby-Normal-Risk. The NNR comes into play when a point x doesn't have normal adversarial risk, and the score function on all points nearby agrees with $y(x)$, yet there is a point near x that has non-zero normal adversarial risk.

2. The second result states that if the normal adversarial risk is zero, then the ϵ-adversarial risk is bounded by the sum of the standard risk and the 2ϵ in-manifold risk.

One may wonder if a decomposition of the form $R_{adv}(f, \epsilon) \leq R_{std}(f) + R_{adv}^{nor}(f, \epsilon) + R_{adv}^{in}(f, 2\epsilon)$ is possible. We prove that this is not possible.

Theorem 2. *[Tightness of Decomposition Result]*

For any $\epsilon < 1/2$, there exists a sequence $\{f_n\}_{n=1}^\infty$ of continuous score functions such that

1. *$R_{std}(f) = 0$ for all $n \geq 1$,*

2. *$R_{adv}^{in}(f_n, 2\epsilon) = 0$ for all $n \geq 1$, and*

3. *$R_{adv}^{nor}(f_n, \epsilon) \to 0$ as n goes to infinity,*

but $R_{adv}(f, \epsilon) = 1$ for all $n > \frac{1}{\sqrt{3}\epsilon}$.

Thus all three terms except the NNR term go to zero, but the adversarial risk (the left side of Equation 2) goes to one.

2.4 Decomposition when y is Deterministic

Let $\eta(x) = Pr(y = 1|x)$. We consider here the simplistic setting when $\eta(x)$ is either zero or one, i.e., y is a deterministic function of x. In this case, we can explain our decomposition result in a simpler way.

Let $Z^{nor}(f, \epsilon) := \{x \in \mathcal{M} : f(x)y > 0 \text{ and } \exists x' \neq x \in B_\epsilon^{nor}(x), f(x')y(x) \leq 0\}$. That is, $Z^{nor}(f, \epsilon)$ is the set of points with no standard risk, but with a non-zero normal adversarial risk under a positive but less than ϵ normal perturbation. Let $\overline{Z^{nor}(f, \epsilon)} = \mathcal{M} \setminus Z^{nor}(f, \epsilon)$ be the complement of $Z^{nor}(f, \epsilon)$. For a set $A \subset \mathcal{M}$, let $\mu(A)$ denote the measure of A.

Corollary 1. *Let $\mathcal{M}$ be a smooth compact manifold in $\mathbb{R}^D$, and let $\eta(x) \in \{0, 1\}$ for all $x \in M$. There exists a $\Delta > 0$ depending only on $\mathcal{M}$ such that the following statements hold for any $\epsilon < \Delta$. For any score function f satisfying assumption A,*

(i)

$$\begin{aligned} R_{adv}(f, \epsilon) &\leq R_{std}(f) + R_{adv}^{in}(f, 2\epsilon) + R_{adv}^{nor}(f, \epsilon) \\ &+ \mu(\overline{Z^{nor}(f, \epsilon)} \cap B_{2\epsilon}(Z^{nor}(f, \epsilon)) \quad (2) \end{aligned}$$

(ii) If $R_{adv}^{nor}(f, \epsilon) = 0$, then $R_{adv}(f, \epsilon) \leq R_{std}(f) + R_{adv}^{in}(f, 2\epsilon)$.

Therefore in this setting, the adversarial risk can be decomposed into the in-manifold risk and the measure of a neighborhood of the points that have non-zero normal adversarial risk.

2.5 Proofs of Theorems 3 and 2

The complete proof of Theorem 3 is technical and is provided in the supplementary materials. Here we provide a sketch of the proof first. Then we give the complete proof of Theorem 2.

2.5.1 Proof Sketch of Theorem 3

We first address the existence of the constant Δ that only depends on $\mathcal{M}$ in the theorem statement. Define a *tubular neighborhood* of $\mathcal{M}$ as a set $\mathcal{N} \subset \mathbb{R}^D$ containing $\mathcal{M}$ such that any point $z \in \mathcal{N}$ has a unique projection $\pi(z)$ onto $\mathcal{M}$ such that $z - \pi(z) \in N_{\pi(z)}\mathcal{M}$. Thus the normal line segments of length ϵ at any two points $x, x' \in \mathcal{M}$ are disjoint.

By Theorem 11.4 in Bredon (2013), we know that there exists Δ such that $N := \{y \in \mathbb{R}^D : dist(y, \mathcal{M}) < \Delta\}$ is a tubular neighborhood of $\mathcal{M}$. The Δ guaranteed by Theorem 11.4 is the Δ referred to in our theorem, and the budget ϵ is constrained to be at most Δ.

For simplicity, we first sketch the proof of the case when y is deterministic (the setting of Corollary 1). Consider a pair $(x, y) \sim p$, such that x has an adversarial perturbation x' within distance ϵ. We show that one of the four cases must occur:

- $x' = x$ (standard risk).

- $x' \neq x$, $x' \in N_x\mathcal{M}$, and $f(x)y > 0$ (normal adversarial risk).

- Let $x'' = \pi(x')$ (the unique projection of x' onto $\mathcal{M}$), then $d(x'', x) \leq 2\epsilon$ and either

  - $f(x'')y \leq 0$, and x has an 2ϵ in-manifold adversarial perturbation (in-manifold risk), or
  - $f(x'')f(x') \leq 0$, which implies that x is within 2ϵ of a point $x'' \in \mathcal{M}$ that has non-zero normal adversarial risk. (NNR: nearby-normal-risk)

The second of these sets is $Z^{nor}(f, \epsilon)$ in the setting of Corollary 3. One can see that the four cases correspond to the four terms in Equation 2.

For the proof of Theorem 3, one has to observe that since y is not deterministic, the set $Z^{nor}(f, \epsilon)$ is random. One then has to average over all possible $Z^{nor}(f, \epsilon)$, and show that the average equals NNR.

For the second part of Theorem 1 and Corollary 1, observe that if the normal adversarial risk is zero, then in the last case, x'' has non-zero normal adversarial risk, with normal adversarial perturbation x'. Unless x'' is on the decision boundary, by continuity of f one can show that there exists an open set around x'' such that all points here have non-zero normal adversarial risk. This contradicts the fact that the normal adversarial risk is zero, implying that case 4 happens only on a set of measure zero (recall that by assumption A the decision boundary does not contain any open set). This completes the proof sketch.

2.5.2 Proof of Theorem 2

Let $\mathcal{M} = [0, 1]$ and fix $\epsilon < 1/2$ and $n \geq 1$. We will think of data as lying in the manifold $\mathcal{M}$, and $\mathbb{R}^2$ as the ambient space. The true distribution is simply $\eta(x) = 1$ for all $x \in \mathcal{M}$, hence $y \equiv 1$ (all labels on $\mathcal{M}$ are 1).

Let $\ell_1 = \frac{n-1}{n(n+1)}$ and $\ell_2 = \frac{1}{n^2}$. Note that $(n+1)\ell_1 + n\ell_2 = 1$. Consider the following partition of $\mathcal{M} = A_0 \cup B_1 \cup A_1 \cup B_2 \cup \cdots \cup B_n \cup A_n$, Where A_i $(0 \leq i \leq n)$ is of length ℓ_1 and B_i $(1 \leq i \leq n)$ is an interval of length ℓ_2. The interval $A_0, B_1, A_1, \cdots, B_n, A_n$ appear in this order from left to right.

For ease of presentation, we will consider $\{0, 1\}$ binary labels, and build score functions f_n taking values in $[0, 1]$ that satisfy the conditions of the Theorem.

For an $x \in A_i$ for some $0 \leq i \leq n$, define $g_n(x) = 1$. For $x \in B_i$ for some $1 \leq i \leq n$, define $g_n(x) = \epsilon/2$. Observe that $\epsilon/2 < 1/4$.

We now define the decision boundary of f_n as the set of points in $\mathbb{R}^2$ on the "graph" of g_n and $-g_n$. That is,

$$DB(f_n) = \{(x, cg_n(x)) : x \in [0, 1], c \in \{-1, 1\}\}.$$

See Figure 2 for a picture of the upper decision boundary. Now let f_n be any continuous function with decision boundary $DB(f_n)$ as above. That is, $f_n : \mathbb{R}^2 \to [0, 1]$ is such that $f_n(x, t) > 1/2$ if $|t| < g_n(x)$, $f_n(x, t) < 1/2$ if $|t| > g_n(x)$ and $f_n(x, y) = 1/2$ if $|t| = g_n(x)$.

In-manifold Risk Is Zero Observe that since $\eta(x) = 1$ on $[0, 1]$, the in-manifold risk of f_n is zero, since $f_n(x, 0) > 1/2$, and so $\text{sign}(2f_n - 1)$ equals 1, which is the same as the label y at x. This means that there are no in-manifold adversarial perturbations, no matter the budget. Thus $R_{adv}^{in}(f_n, \epsilon) = 0$ for all $n \geq 1$.

Normal Adversarial Risk Goes To Zero Next we consider the normal adversarial risk. If $x \in A_i$ for some i, then a point in the normal ball with budget ϵ is of the form (x, t) with $|t| < \epsilon < 1/2$, but $f_n(x, t) > 1/2$ for such points, and thus $\text{sign}(2f_n - 1) = y(x)$. Thus $x \in A_i$ does not contribute to the normal adversarial risk.

If $x \in B_i$ for some i then $f_n(x, \epsilon) < 1/2$ while $f_n(x, 0) > 1/2$, and hence such x contributes to the normal adversarial risk. Thus $R_{adv}^{nor}(f_n, \epsilon) = \sum_{i=1}^n \mu(B_i) = \sum_{i=1}^n \ell_2 = 1/n$, which goes to zero as n goes to infinity.

Adversarial Risk Goes To One Now we show that $R_{adv}(f_n, \epsilon)$ goes to one. In fact, we will show that as
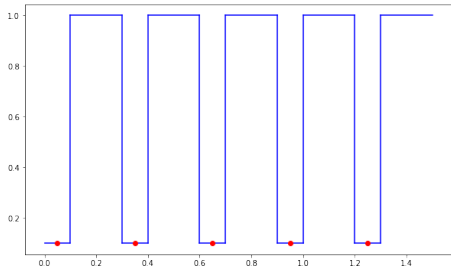
Figure 2: Lower bound illustration

long as n is sufficiently large, the adversarial risk is 1. Consider n such that $\ell_1 := \frac{n-1}{n(n+1)} < \sqrt{3}\epsilon$. Note that such an n exists simply because ℓ_1 goes to zero as n goes to infinity, and $n > \frac{1}{\sqrt{3}\epsilon}$ works.

Clearly, points in B_i contribute to adversarial risk as they have adversarial perturbations in the normal direction. However, if we consider $x \in A_i$ (which does not have adversarial perturbations in the normal direction or in-manifold), we show that there still exists an adversarial perturbation in the ambient space: that is, there exists a point x' such that a) the distance between $(x', \epsilon/2)$ and $(x, 0)$ is at most ϵ, and b) $sign(2f_n(x, \epsilon/2)) \neq sign(2f_n(x, 0))$. Let x' be the closest point in $B := \cup B_i$ to x. Then $|x' - x| \leq \ell_1/2 < \sqrt{3}\epsilon/2$. Thus the distance between $(x', \epsilon/2)$ and $(x, 0)$ is at most $\sqrt{(\sqrt{3}\epsilon/2)^2 + (\epsilon/2)^2} = \epsilon$. Since $x' \in B$, $f_n(x', \epsilon/2) < 1/2$ whereas $f_n(x, 0) < 1/2$, $(x', \epsilon/2)$ is a valid adversarial perturbation around x.

Thus for all $x \in [0, 1]$, there exists an adversarial perturbation within budget ϵ, and therefore $R_{adv}(f_n, \epsilon) = 1$ as long as $n > \frac{1}{\sqrt{3}\epsilon}$. This completes the proof.

3 EXPERIMENT

In this section, we verify the decomposition upper bound in Corollary 3 on synthetic data sets. For both i) and ii) in Corollary 3, we empirically evaluate each term in the inequality on several classifiers and compare the values according to the claims in Corollary 3.

In the following experiments, instead of using l_2 norm to evaluate the perturbation, we search the neighborhood under l_∞ norm, which would produce a stronger attack than l_2 norm one. The experimental results indicate that our theoretical analysis may hold for an even stronger attack.

3.1 Toy Data Set and Perturbed Data

We generate four different data sets where we study both the *single decision boundary case* and the *double*

decision boundary case. The first pair of datasets are in 2D space and the second pair is in 3D. We aim to provide empirical evidence for the claim $ii)$ in the Corollary 3 using the single decision boundary data, having observed that one can sufficiently reduce R_{adv}^{nor}, allowing one directly compare $R_{adv}(f, \epsilon)$ and $R_{adv}^{in}(f, 2\epsilon)$. We aim to provide empirical evidence for the claim $i)$ in the Corollary 3 using double boundary, since R_{adv}^{nor} can not be sufficiently reduced using a simple classifier, since decision boundary is complicated.

For the 2D case, we sample training data uniformly from a unit circle $C_1 : x_1^2 + x_2^2 = 1$. For the single decision boundary data set, we set

$$y = 2\mathbb{1}(x_1 > 0) - 1 \text{ (Single Decision Boundary)}$$
$$y = 2\mathbb{1}(x_1 x_2 > 0) - 1 \text{ (Double Decision Boundary)}$$

The visualization of the dataset is in Figure 3 a) and b). In particular, we set unit circle C_1 has $\Delta = 1$, we set the perturbation budget to be $\varepsilon \in [0.01, 0.3]$. And the normal direction is alone the radius of the circle.

In the 3D case, we set the manifold to be $\mathcal{M} : x_3 = 0$ and generate training data in region $[-\pi, \pi] \times [-\pi, \pi]$ on $x_1 x_2$-plane. We set

$$y = 2\mathbb{1}[x_1 > sin(x_2)] - 1 \text{(Single)}$$
$$y = 2\mathbb{1}[(x_1 - sin(x_2))x_2 > 0] - 1 \text{(Double)}$$

Figure 3 c) and d) show these two cases. For the single decision boundary example, due to the manifold being flat, we have $\Delta = \infty$, we explore the ϵ value in range $[0.1, 0.8]$. For the double decision boundary, the distance to the decision boundary is half of the distance in the single boundary case. Therefore, we set the range of perturbation to be $[0.1, 0.4]$.

3.2 Algorithm

To empirically estimate the decomposition of adversarial risk, we need to generate adversarial data alone different directions, i.e. the normal direction risk R_{adv}^{nor}, the in-manifold risk R_{adv}^{in} and the general adversarial risk R_{adv}. For general adversarial risk, we evaluate risks on perturbed example x^{adv} computed by Projected Gradient Descent algorithm in (Madry et al., 2017).

By the definition of toy data sets, we know that the dimension of ambient space is 1. The normal space at point x can be represented by $N_x \mathcal{M} = \{x + t \cdot v | 0 < t < \epsilon\}$, here v is a unit normal vector. Therefore we could explicitly compute the normal vector v and select normal direction adversarial data x^{nor}.

We evaluate different components in the inequality in Corollary 3 on three classifiers. The standard classifier f trained by original training data set,

Table 1: 2D Adversarial risk comparison

Single Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R^{in}_{adv}(2\epsilon)$	R^{nor}_{adv}	R_{std}
0.01	0.0110	0.022	0.0110	0.022	0.0090	0.0220	0.0140	0.0050	0.0050
0.02	0.0130	0.0449	0.0130	0.0449	0.0130	0.0449	0.0290	0.0060	0.0060
0.03	0.0230	0.063	0.0250	0.0671	0.0230	0.0633	0.0400	0.0120	0.0120
0.05	0.0280	0.0794	0.0300	0.0784	0.0280	0.0794	0.0620	0.0040	0.0040
0.1	0.0709	0.1652	0.0699	0.1645	0.0709	0.1650	0.133	0.0	0.0040
0.15	0.0979	0.2831	0.1009	0.2886	0.1009	0.2866	0.1850	0.0050	0.0050
0.2	0.128	0.3951	0.126	0.3971	0.128	0.4086	0.261	0.0050	0.0040
0.25	0.1660	0.4966	0.1630	0.4931	0.1660	0.4986	0.3259	0.0040	0.0040
0.3	0.1979	0.4509	0.1979	0.5613	0.1979	0.4505	0.35	0.0	0.0
Double Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R^{in}_{adv}(2\epsilon)$	R^{nor}_{adv}	R_{std}
0.01	0.0080	0.0286	0.0060	0.0296	0.0070	0.0276	0.0180	0.0030	0.0030
0.02	0.0240	0.0694	0.0230	0.2525	0.0240	0.0694	0.0490	0.0050	0.0050
0.03	0.0510	0.1333	0.0460	0.1363	0.0510	0.1383	0.0949	0.0110	0.0110
0.05	0.0620	0.1810	0.0620	0.1640	0.0629	0.1640	0.1139	0.0080	0.0080
0.1	0.1170	0.3398	0.1169	0.3071	0.12	0.2746	0.2400	0.0060	0.0060
0.15	0.1850	0.6059	0.1860	0.4895	0.1939	0.5948	0.3860	0.0040	0.0040
0.2	0.242	0.8763	0.247	0.8002	0.265	0.878	0.5409	0.0060	0.0050
0.25	0.3139	1.	0.3169	0.9971	0.3239	1.	0.6500	0.0080	0.0080
0.3	0.386	0.9615	0.379	1.	0.394	1.	0.6520	0.0070	0.0060

Table 2: 3D Adversarial risk comparison

Single Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R^{in}_{adv}(2\epsilon)$	R^{nor}_{adv}	R_{std}
0.1	0.0450	0.0992	0.0410	0.092	0.0470	0.1002	0.0959	0.0050	0.0050
0.2	0.1139	0.2297	0.0999	0.229	0.1099	0.2143	0.1929	0.0100	0.0199
0.3	0.1550	0.3106	0.136	0.3216	0.1540	0.2852	0.239	0.0080	0.0265
0.4	0.2089	0.3765	0.1680	0.3889	0.2059	0.3579	0.26	0.0080	0.0193
0.5	0.247	0.4910	0.1860	0.4404	0.250	0.4104	0.252	0.0040	0.0174
0.6	0.2700	0.5910	0.2179	0.5198	0.257	0.417	0.257	0.0090	0.0153
0.7	0.2600	0.6057	0.2009	0.7571	0.2731	0.4224	0.273	0.0030	0.0139
0.8	0.2329	0.6775	0.1670	0.5630	0.2339	0.4083	0.2329	0.0020	0.0129
Double Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R^{in}_{adv}(2\epsilon)$	R^{nor}_{adv}	R_{std}
0.1	0.0649	0.1654	0.0789	0.153	0.0759	0.1654	0.1540	0.0130	0.0140
0.15	0.1460	0.3065	0.1280	0.2949	0.1510	0.3026	0.272	0.0220	0.0270
0.2	0.1700	0.3858	0.1370	0.3341	0.1670	0.3541	0.3040	0.0170	0.0170
0.25	0.2049	0.4608	0.1500	0.4203	0.2099	0.4486	0.361	0.0210	0.0210
0.3	0.2159	0.4740	0.1810	0.4208	0.2119	0.4450	0.3289	0.0190	0.0190
0.35	0.275	0.5176	0.2039	0.5289	0.2750	0.4830	0.356	0.0110	0.0130
0.4	0.3000	0.6051	0.2069	0.5325	0.3040	0.6593	0.3690	0.0520	0.0080

the adversarial classifier f^{adv} trained by Adversarial Training algorithm in Madry et al. (2017) and the classifier trained using adversarial samples generated in the normal direction x^{nor}, we denote it as f^{nor}. To compute the in-manifold perturbation, we design two methods. The first one is using grid search to go through all the perturbations in the manifold within the ϵ budget and return the point with maximum loss as in-manifold perturbation x^{in}. The second is using PGD method to find a general adversarial point x^{adv} in ambient space and project x^{adv} back to the data manifold $\mathcal{M}$. Due to grid-search being

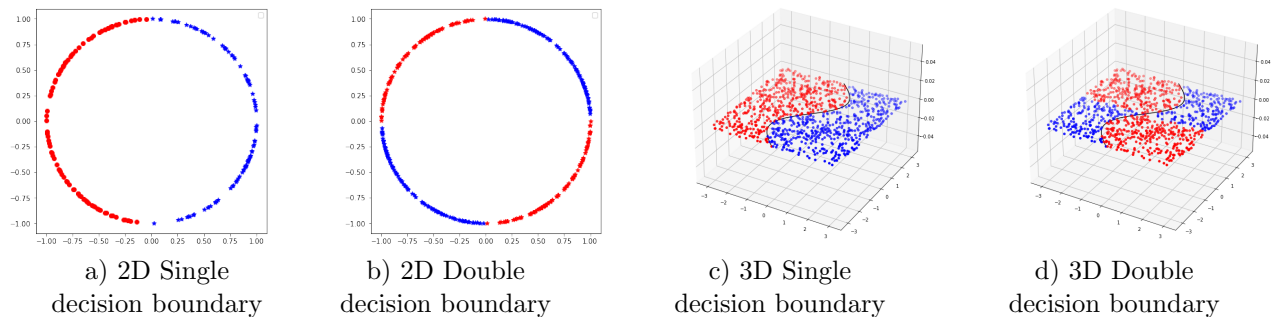| a) 2D Single decision boundary | b) 2D Double decision boundary | c) 3D Single decision boundary | d) 3D Double decision boundary |

Figure 3: In this figure, we show our four toy data set. On the left side is 2D data set on a unit circle. The single decision boundary data is linearly separated by the y-axis. And in the double decision boundary case, the circle is separated into 4 parts with x and y-axis. On the right side is the 3D data set. The data is distributed in a square area on $x_1 x_2$-plane. In the single decision boundary example, the data is divided by the curve $x_1 = sin(x_2)$. And in the double decision boundary situation, we add the y-axis as the extra boundary.

time-consuming, we use the second method in our experiments below. We further compare these two methods in supplementary materials.

Here we propose a method based on Adversarial Training to compute f^{nor} in Algorithm 1. One thing worth mentioning here is, instead of using grid search to find the actual x^{nor}, we use an intermediate method to generate normal data. We randomly choose a point along the normal direction within the ϵ budget to be our normal direction perturbed data. This might worsen the normal adversarial risk of f^{nor}, but our empirical results show that $R_{adv}^{nor}(f^{nor})$ still close to 0.

Algorithm 1 Normal direction Adversarial Training

1: Input: Training data set $\{x_i, y_i\}_{i=1}^n$, training iterations K, perturbation budget ϵ
2: **for** $iterations$ in $1, \ldots, K$ **do**
3:     **for** x_i in $\{x_i, y_i\}_{i=1}^n$ **do**
4:         Find normal space $N_{x_i}(\mathcal{M})$ for x_i.
5:         Find $v_i \in B_\varepsilon^{nor}(x_i)$ such that $l(f(x_i + v_i), y_i) \neq 0$.
6:         $x_i^{nor} \leftarrow v_i$
7:         Update f^{nor} with x_i^{nor}.
8:     **end for**
9: **end for**

Note in Corollary 3 claim i), we have a component $\mu(B_\epsilon(Z^{nor}(f, \epsilon)) \cap \mathcal{M})$ on the right hand side (RHS) of the inequality. We also give a practical way of estimating such quantity in the empirical study. By the definition of $Z^{nor}(f, \epsilon)$, we first select point x_i in training data such that there exists point x_i' in $N_{x_i}\mathcal{M}$ s.t $f(x_i') \neq y_i$ to form set $\widehat{Z}$. Since we uniformly sample points from data manifold, the volume of $B_{2\epsilon}(\widehat{Z}) \cap \mathcal{M}$ is proportional to $\mu(B_{2\epsilon}(Z^{nor}(f, \epsilon)) \cap \mathcal{M})$ which is a set derived by point wise augment $\widehat{Z}$ by a 2ε-ball. In

2D example, this quantity is simply the length of curve segment on unit circle as the volume of $B_{2\epsilon}(z)$ for any $z \in \widehat{Z}$. In 3D example, we use area of $\widehat{Z}$ point wise augmented by a 2ϵ square. We list the RHS value for 2D and 3D datasets in Table 1 and Table 2 for all three classifiers.

3.3 Empirical Results and Discussion

2D Unit Circle We generate 1000 training data uniformly. The classifier is a 2-layer feed-forward network. Each classifier is trained with Stochastic Gradient Descent (SGD) with a learning rate of 0.1 for 1000 epochs. Also, since $\Delta = 1$ for the unit circle, the upper bound of ϵ value is up to 1. Hence we run experiments for ϵ from 0.01 to 0.3. By increasing the ϵ budget, we also observe that the decision boundary of f^{nor} becomes perpendicular to the data manifold. In Table 1, the value of $R_{adv}^{nor}(f^{nor})$ also confirm our observation. We leave more discussion and visualization of this phenomenon in the supplementary material.

To verify our results in Corollary 3 claim i). We compute the adversarial risk for three classifiers. And for the upper bound, we evaluate the component $\mu(B_\epsilon(Z^{nor}(f, \epsilon)) \cap \mathcal{M})$ following the description in Section 3.2. The right hand side value in the inequality is given in Table 1. We could observe that the upper bounds hold for 2D data.

Since we train f^{nor} to minimize its empirical risk in normal direction. By Table 1, we know $R_{adv}^{nor}(f^{nor})$ is close to zero. Therefore it is reasonable to study claim ii) in Corollary 3 using f^{nor}. The summation of in-manifold risk and standard risk of f^{nor} certainly upper bounds $R_{adv}(f^{nor})$.

3D X_1X_2-plane We generate 1000 training data from the data set. The classifier is a 4-layer feedforward

network. We use SGD with a learning rate of 0.1 and weight decay of 0.001 to train the network. The total training epoch is 2000.

In Table 2, we list same three classifiers trained on 3D data set. We have R^{adv} been upper bounded by the right hand side of the inequality in Corollary 3 claim i). The claim ii) also holds in 3D cases. Due to the limit of the space, we provide visualization of the decision boundary and additional empirical results in the supplemental material.

4 CONCLUSION

In this work, we study the adversarial risk of the machine learning model from the manifold perspective. We report theoretical results that decompose the adversarial risk into the normal adversarial risk, the in-manifold adversarial risk, and the standard risk with the additional Nearby Normal Risk term. We present a pessimistic case suggesting the additional Nearby Normal Risk term can not be removed in general, without additional assumptions. Observing that the Nearby Normal Risk term can be wiped out by enforcing zero normal adversarial risk, our theoretical analysis suggests a potential training strategy that only focuses on the normal adversarial risk.

Acknowledgements

References

Pranjal Awasthi, Abhratanu Dutta, and Aravindan Vijayaraghavan. On robustness to adversarial examples and polynomial optimization. *arXiv preprint arXiv:1911.04681*, 2019.

Glen E Bredon. *Topology and geometry*, volume 139. Springer Science & Business Media, 2013.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17): 1, 2005.

Tamal K Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry*, 35(1-2):124–141, 2006.

Elvis Dohmatob. Generalized no free lunch theorem for adversarial robustness. In *International Conference on Machine Learning*, pages 1646–1654. PMLR, 2019.

Herbert Edelsbrunner and Nimish R Shah. Triangulating topological spaces. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 285–292, 1994.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. *arXiv preprint arXiv:1802.08686*, 2018.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *NIPS*, volume 27, pages 1071–1079. Citeseer, 2014.

Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. *Advances in Neural Information Processing Systems*, 33:3487–3498, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*, pages 1786–1794, 2010.

Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

Salah Rifai, Yann N Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. *Advances in neural information processing systems*, 24:2294–2302, 2011.

Lawrence K Saul and Sam T Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Departmental Papers (CIS)*, page 12, 2003.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290 (5500):2319–2323, 2000.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.

Supplementary Material:
A Manifold View of Adversarial Risk

A   PROOF OF THEOREM 1

Theorem 3. *[Risk Decomposition] Let $\mathcal{M}$ be a smooth compact manifold in $\mathbb{R}^D$, and let data be drawn from $\mathcal{M} \times \{-1, 1\}$ according to some distribution p. There exists a $\Delta > 0$ depending only on $\mathcal{M}$ such that the following statements hold for any $\epsilon < \Delta$. For any score function f satisfying assumption A,*

(i)
$$R_{adv}(f, \epsilon) \leq R_{std}(f) + R_{adv}^{nor}(f, \epsilon) + R_{adv}^{in}(f, 2\epsilon) + NNR(f, \epsilon).$$

(ii) If $R_{adv}^{nor}(f, \epsilon) = 0$, then
$$R_{adv}(f, \epsilon) \leq R_{std}(f) + R_{adv}^{in}(f, 2\epsilon)$$

Proof of i): We first address the existence of the constant Δ that only depends on $\mathcal{M}$ in the theorem statement.

Definition 5 (Tubular Neighborhood). *A tubular neighborhood of a manifold $\mathcal{M}$ is a set $\mathcal{N} \subset \mathbb{R}^D$ containing $\mathcal{M}$ such that any point $z \in \mathcal{N}$ has a unique projection $\pi(z)$ onto $\mathcal{M}$ such that $z - \pi(z) \in N_{\pi(z)}\mathcal{M}$.*

By Theorem 11.4 in Bredon (2013), we know that there exists $\Delta > 0$ such that $N := \{y \in \mathbb{R}^D : dist(y, \mathcal{M}) < \Delta\}$ is a tubular neighborhood of $\mathcal{M}$. This also implies that for any $0 < \epsilon < \Delta$, the normal line segments of length ϵ at any two points $x, x' \in \mathcal{M}$ are disjoint, a fact that will be used later.

The Δ guaranteed by Theorem 11.4 is the Δ referred to in our theorem, and the budget $\epsilon > 0$ is constrained to be at most Δ.

Next we consider the left hand side, the adversarial risk:

$$R_{adv}(f, \epsilon) := \mathbb{E}_{(x,y) \sim p} \mathbb{1}(\exists x' \in B_\epsilon(x) : f(x')y \leq 0)$$

Denote by $E(x, y)$ the event that $\exists x' \in B_\epsilon(x) : f(x')y \leq 0$.

We will write the indicator function above as the sum of indicator functions of four events. Specifically, define by $E_1(x, y), E_2(x, y), E_3(x, y), E_4(x, y)$ the following four events:

- $E_1(x, y)$: $f(x)y \leq 0$.

- $E_2(x, y)$: $f(x)y > 0$ and $\exists x' \neq x \in B_\epsilon(x)$ such that $x' - x \in N_x\mathcal{M}$ and $f(x')y \leq 0$.

For the next two cases, let $x' \neq x \in B_\epsilon(x)$ be such that $x' - x \notin N_x\mathcal{M}$ and $f(x')y \leq 0$ (if such an x' exists). Let $x'' = \pi(x')$ be the unique projection of x' onto $\mathcal{M}$. Note that $x'' \neq x$. Define:

- $E_3(x, y)$: $f(x'')y \leq 0$.

- $E_4(x, y)$: $f(x'')y > 0 \iff f(x'')f(x') \leq 0$.

Lemma 1.
$$\mathbb{1}(E(x, y)) = \mathbb{1}(E_1(x, y)) + \mathbb{1}(E_2(x, y)) + \mathbb{1}(E_3(x, y)) + \mathbb{1}(E_4(x, y))$$

Proof. Assume $E(x, y)$ occurs, i.e, $\exists x' \in B_\epsilon(x) : f(x')y \leq 0$. Either $x' = x$ satisfies the condition (which is event E_1) or some $x' \neq x$ satisfies the condition.

Now we further divide into the case when $f(x)y > 0$ and $x' - x \in N_x\mathcal{M}$ (which is event E_2), or $f(x)y > 0$ and $x' - x \notin N_x\mathcal{M}$. In the latter case, note that $x'' = \pi(x')$ cannot equal x as otherwise $x' - x$ would be in the normal space at x, since the projection map is unique inside the tubular neighborhood. Thus x'' is well-defined, and it is easy to see that the last two cases are disjoint and cover this remaining case. Thus we have shown that if $E(x, y)$ occurs, then one of the four disjoint events E_i must occur, proving the lemma.

$\square$

Finally we have the following lemma, which completes the proof of the theorem after combining with Lemma 1.

Lemma 2. *The following relation holds between the risk and the expectation of the indicator functions in Lemma 1*

1. $\displaystyle \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(E_1(x, y)) = R_{std}(f)$

2. $\displaystyle \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(E_2(x, y)) \leq R_{adv}^{nor}(f, \epsilon)$

3. $\displaystyle \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(E_3(x, y)) \leq R_{adv}^{in}(f, 2\epsilon)$

4. $\displaystyle \mathop{\mathbb{E}}_{(x,y)\sim p} \mathbb{1}(E_4(x, y)) \leq NNR(f, \epsilon)$

Proof. 1) and 2) follow by definitions of standard adversarial risk and normal adversarial risk, respectively. Consider the setting of $E_3(x, y)$: i.e., $f(x)y > 0$, the adversarial perturbation x' is not in the normal direction (so $f(x')y \leq 0$), and $f(x'')y \leq 0$. Observe that by the triangle inequality, $d(x, x'') \leq d(x, x') + d(x', x'') \leq \epsilon + \epsilon = 2\epsilon$, simply because a) x' is within the ϵ-ball of x, and b) x'' is closer to x' than x.

This means that there is a point $x'' \in B_{2\epsilon}^{in}(x)$ such that $f(x'')y \leq 0$. The expectation over a random $(x, y) \sim p$ of this event is clearly at most $R_{adv}^{in}(f, 2\epsilon)$ (the inequality need not be tight because x may have adversarial perturbation within 2ϵ and also satisfy some other events like E_1).

Lastly, by the definition of the NNR, we see that $A(x, y)$ occurs when $E_1(x, y)$ or $E_2(x, y)$ do not. Also $C(x, y)$ implies that the event $E_3(x, y)$ does not occur. We are now in the situation where x'' is within 2ϵ of x, $f(x')y \leq 0$, and $f(x'')y > 0$. But this implies that $f(x'')f(x') \leq 0$, and since $x' \in B_\epsilon^{nor}(x'')$, it implies that $B(x, y)$ occurs. Thus all of $A(x, y)$, $B(x, y)$ and $C(x, y)$ occur, which is the definition of NNR. $\square$

Proof of ii)

If $R_{adv}^{nor}(f, \epsilon) = 0$, we claim that $NNR(f, \epsilon) = 0$. Setting these two terms to zero in i) proves ii).

Note that although $R_{adv}^{nor}(f, \epsilon) = 0$, it does not imply that there are no normal adversarial perturbations for any x— it just means that the measure of such x with normal adversarial perturbation is zero.

Also note that $R_{adv}^{nor}(f, \epsilon) = 0$ does not exclude $A(x, y)$ or $C(x, y)$ from occurring (in fact A occurs for almost all x). Thus the proof will focus on the measure of points where $B(x, y)$ can occur. We will prove the following lemma, which will complete the proof of the theorem.

Lemma 3. *Let (x, y) be such that $B(x, y)$ occurs, i.e., there exist $x' \in B_\epsilon(x)$ and $x'' = \pi(x)$ such that $f(x')y \leq 0$, $f(x'')y > 0$ and $d(x, x'') \leq 2\epsilon$. Then $C(x, y)$ cannot occur, i.e., there exists a point $w \in B_{2\epsilon}^{in}(x)$ such that $f(w)y \leq 0$. Consequently, $NNR(f, \epsilon) = 0$.*

Proof. We first claim that if $B(x, y)$ occurs, it must be the case that $f(x'') = 0$. Assuming this, if $f(x'') = 0$, then by Assumption A we know there exists an $s \in B_\epsilon(x'') \cap B_{2\epsilon}(x)$ such that $f(s)y \leq 0$, which imply that $C(x, y)$ cannot occur. This will complete the proof of the lemma.

To prove that $f(x'') = 0$, consider what happens if $f(x'') \neq 0$. Assume first that $f(x') \neq 0$, and note that $f(x')f(x'') \leq 0$. By continuity of f, there exist open neighborhoods $U \ni x''$ and $V \ni x'$ such that f has the same sign on all of U and the same sign on all of V, i.e., $sign(f|U) = sign(f(x''))$ and $sign(f|V) = sign(f(x'))$.

Consider the normal bundle on U defined as the set $U' = \{y \in \mathcal{M}_\Delta : \pi(y) \in U\}$. In other words, U' is the union of the normal line segments passing through points in U (here $\mathcal{M}_\Delta$ denotes the tubular neighborhood of $\mathcal{M}$). Note that U' is an open set.

Define $W' = U' \cap V$, and $W = \pi(W')$. $W \subset \mathcal{M}$ is an open set, but for every $w \in W$, there exists a point $w' \in W' \cap B_\epsilon^{nor}(w)$ such that $f(w')f(w) \leq 0$. Therefore there exists anormal adversarial perturbation for every point in W. Since the measure of W is not zero, this contradicts the fact that $R_{adv}^{nor}(f, \epsilon) = 0$.

The proof is completed by observing that in the remaining case when $f(x'') \neq 0$ but $f(x') = 0$, there must exist (by assumption A) a point w near x' such that $f(w) \neq 0$ and $f(w)y < 0$. This lands us in the previous case, which we showed contradicts the hypothesis that $R_{adv}^{nor}(f, \epsilon) = 0$. $\qquad\square$

Remark: In Corollary 3, $\mu(\overline{Z^{nor}(f,\epsilon)} \cap B_{2\epsilon}(Z^{nor}(f,\epsilon)))$ is the NNR under deterministic case. Therefore, Corollary 3 follows directly from the proof of Theorem 1.

B   ADDITIONAL EXPERIMENTS

In the main paper, we leave some experimental results to discuss in this supplementary materials. In the following section, we will first compare different ways of generating in-manifold attack data. In the later section, we compare the decision boundary of different classifiers. By visualization of the decision boundary, we aim to show that the defense training algorithm can defend the model against adversarial examples in the normal direction implying that the adversarial risk in the normal direction can be controlled.

Also, we need to mark out that when we have a small ϵ value. The RHS for classifier f^{nor} might be a little bit smaller than claim $ii)$ in Corollary 3. This is due to the fact that our way of computing measure $\mu(\overline{Z^{nor}(f,\epsilon)} \cap B_{2\epsilon}(Z^{nor}(f,\epsilon)))$ includes the standard risk R^{std} and normal risk R^{nor}. So the summation of $\mu(\overline{Z^{nor}(f,\epsilon)} \cap B_{2\epsilon}(Z^{nor}(f,\epsilon)))$ and R_{adv}^{in} forms the RHS. When we have a small ϵ value, the points with non-zero normal adversarial risk are concentrated near the decision boundary. Since we compute the $\mu(\overline{Z^{nor}(f,\epsilon)} \cap B_{2\epsilon}(Z^{nor}(f,\epsilon)))$ based on the ratio between the length of line segment (or area of cube in 3D) and the circumference of unit circle (or area of $x_1 x_2$-plane unit square), the ratio could be close to zero. Therefore, we might have the value of RHS smaller than the summation of $R_{adv}^{in} + R^{std} + R_{adv}^{nor}$. Aside from this, RHS still upper bounds R^{adv} in all cases.

B.1   In-Manifold Attack Algorithm

To estimate the *in-manifold adversarial risk*, we have tested two potential algorithms for generating in-manifold adversarial examples. We present our observations on these two methods. Our empirical study in the paper leverage one of the two methods presented below, which generates a more powerful in-manifold adversarial example.

One way to generate the adversarial samples is by brutal force. We use the grid search method to search the $B_\epsilon^{in}(x)$ region and find the maximum loss point in that region. We treat the maximum loss point as the in-manifold adversarial data. We call this approach the grid search method. Another approach we name as the projected method. We set the step size of the grid search proportional to the perturbation budget ϵ. In general, we search 100 points in 1D cases and 400 points in the 2D manifold. In the projected method, we first use a general adversarial attack algorithm to generate adversarial data in ambient space. Then we project the generated adversarial example back to the manifold and return the results as our in-manifold adversarial data. In the following experiment, we use PGD as our generator of adversarial data in ambient space. Both methods will find in-manifold data that is adversarial to the given model. The rest of the experiment settings follow Section 3 in the main paper.

In Figure 4 we plot the after-attack accuracy of these two in-manifold attack methods. The experiments follow the same setting as the one we described in the main paper. We could observe that the grid search is slightly stronger in the 3D single boundary case and equivalent to the projection method in the rest of the cases. In the graph, the after-attack accuracy of the grid search method matches with the projection methods in the 2D case. And in the 3D case, when the ϵ is larger than 0.5, then the grid search method achieves smaller after attack accuracy. This is due to the projection method searching the adversarial example in a smaller in-manifold ball.

a) 2D Single
decision boundary

b) 2D Double
decision boundary

c) 3D Single
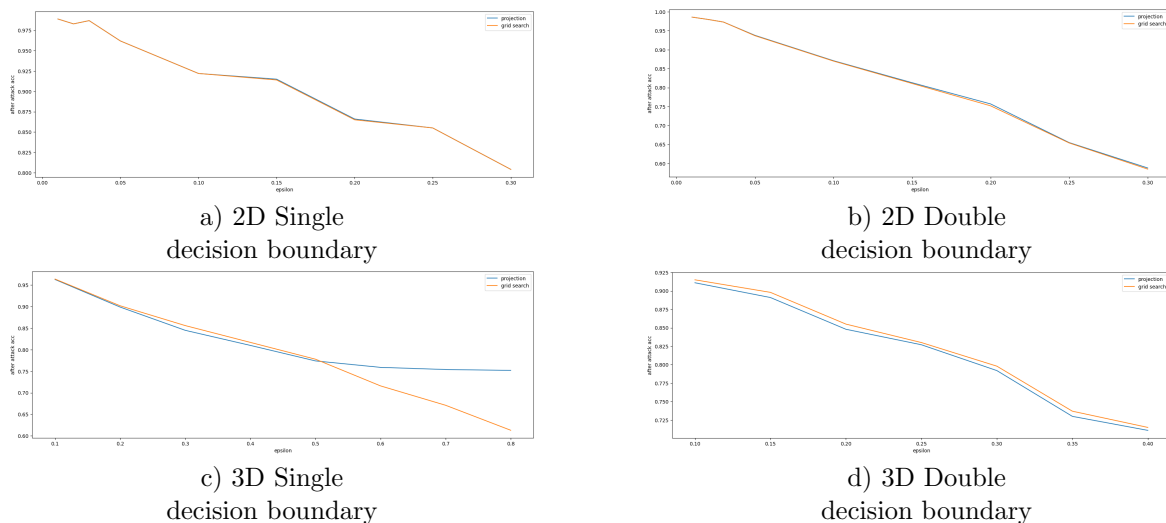decision boundary

d) 3D Double
decision boundary

Figure 4: We compare the grid search method and projection method to generate in-manifold attack data. The first row is after attack accuracy on the 2D data set. The blue line is the accuracy of the projection approach. Orange is for the grid search method. The ϵ range is smaller than the range we choose in the discussion of the main paper. This is because ϵ-budget is larger than 0.05. The after-attack accuracy remains zero. The lower row is after attack accuracy on two different 3D data sets.

In other words, it hasn't fully explored the ϵ ball around the original data point. Therefore we could observe this small gap between these two methods. In the paper, we rely on the grid search method for generating in-manifold adversarial examples.

Furthermore, we compute the in-manifold risk in Table 1 and 2 using the grid search method. We plot our results in Table 3 and Table 4. Since the attack performance of the grid search approach is stronger than the projection approach, the upper bound holds. In Table 3 and Table 4 we could observe this result.

Comparing Table 1 and 1, we could see that R_{adv}^{in} in Table 3 and Table 4 has similar results. It implies that the projection method does not underestimate the upper in most cases. For the 3D double boundary dataset, the projection method has weaker results but the upper bound still holds. It implies that the upper bound in Corollary 3 is loose in our study case. We could further prove a tighter upper bound in 3 claim ii).

B.2 Decision Boundary Discussion

In this section, we explain one of our intuitions of deriving this decomposing. In geometry, we know that if the decision boundary of the classifier is perpendicular to the manifold, then along normal direction, it is hard to find an adversarial example that can successfully attack the model. Therefore, the general adversarial risk is owing to tangential or in-manifold direction perturbation. Under this setting classifiers with decision boundary perpendicular to the manifold in ambient space would have R_{adv}^{nor} equal zero. And this gives us claim ii) in theorem 3. In the following section, we will plot the classifier's decision boundary in ambient space to state that our intuition holds on the synthetic data set.

B.2.1 2D Decision Boundary

In the 2D synthetic data set, we plot multiple decision boundaries of f^{nor} in the double decision boundary case. As we increase the ϵ budget in the defense algorithm (Algorithm 1 in the paper), the decision boundary becomes more perpendicular to the unit circle. And it matches the results for $R_{adv}^{nor}(f^{nor})$ in Table 1. Around $\epsilon = 0.1$, $R_{adv}^{nor}(f^{nor})$ achieves the minimum value. And we could observe that the shape of the decision boundary is perpendicular and matches with the true label.

Table 3: Computing $R_{adv}^{in}(f^{nor})$ using Grid Search methods for 2D data set

Single Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R_{adv}^{in}(2\epsilon)$	R_{adv}^{nor}	R_{std}
0.01	0.0110	0.0200	0.0110	0.022	0.0090	0.0180	0.0100	0.0050	0.0050
0.02	0.0130	0.0426	0.0130	0.0425	0.0130	0.0439	0.0280	0.0060	0.0060
0.03	0.0230	0.0499	0.0250	0.0595	0.0230	0.0613	0.0380	0.0120	0.0120
0.05	0.0280	0.0871	0.0300	0.0881	0.0280	0.0843	0.0669	0.0040	0.0040
0.1	0.0709	0.1974	0.0699	0.2026	0.0709	0.1620	0.1300	0.0	0.0040
0.15	0.0979	0.2721	0.1009	0.3243	0.1009	0.3225	0.2209	0.0050	0.0050
0.2	0.128	0.4063	0.126	0.4160	0.128	0.4206	0.2730	0.0050	0.0040
0.25	0.1660	0.498	0.1630	0.5218	0.1660	0.5026	0.3299	0.0040	0.0040
0.3	0.1979	0.6117	0.1979	0.6239	0.1979	0.5005	0.4000	0.0	0.0
Double Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R_{adv}^{in}(2\epsilon)$	R_{adv}^{nor}	R_{std}
0.01	0.0080	0.0404	0.0060	0.038	0.0070	0.0386	0.0290	0.0030	0.0030
0.02	0.0240	0.0467	0.0230	0.0457	0.0240	0.0594	0.0390	0.0050	0.0050
0.03	0.0510	0.1279	0.0460	0.1309	0.0510	0.1273	0.0839	0.0110	0.0110
0.05	0.0620	0.1545	0.0620	0.1738	0.0629	0.1711	0.121	0.0080	0.0080
0.1	0.1170	0.4037	0.1169	0.5155	0.12	0.3076	0.273	0.0060	0.0060
0.15	0.1850	0.5649	0.1860	0.5619	0.1939	0.5768	0.368	0.0040	0.0040
0.2	0.242	0.8709	0.247	0.82	0.265	0.88	0.5429	0.0060	0.0050
0.25	0.3139	1.	0.3169	1.	0.3239	1.	0.696	0.0080	0.0080
0.3	0.386	1.	0.379	1.	0.394	1.	0.833	0.0070	0.0060

Table 4: Computing $R_{adv}^{in}(f^{nor})$ using Grid Search methods for 3D data set

Single Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R_{adv}^{in}(2\epsilon)$	R_{adv}^{nor}	R_{std}
0.1	0.0450	0.0974	0.0410	0.098	0.0470	0.0932	0.0889	0.0050	0.0050
0.2	0.1139	0.2062	0.0999	0.2201	0.1099	0.2093	0.1879	0.0100	0.0199
0.3	0.1550	0.3957	0.136	0.3557	0.1540	0.3482	0.3020	0.0080	0.0265
0.4	0.2089	0.5124	0.1680	0.5008	0.2059	0.4729	0.375	0.0080	0.0193
0.5	0.247	0.6057	0.1860	0.5405	0.250	0.6354	0.477	0.0040	0.0174
0.6	0.2700	0.8444	0.2179	0.6828	0.257	0.7169	0.5569	0.0090	0.0153
0.7	0.2600	1.	0.2009	0.8673	0.2731	0.8004	0.651	0.0030	0.0139
0.8	0.2329	1.	0.1670	1.	0.2339	0.8774	0.702	0.0020	0.0129
Double Boundary	f		f^{adv}		f^{nor}				
ϵ	R^{adv}	RHS	R^{adv}	RHS	R^{adv}	RHS	$R_{adv}^{in}(2\epsilon)$	R_{adv}^{nor}	R_{std}
0.1	0.0649	0.1688	0.0789	0.1517	0.0759	0.1624	0.1510	0.0130	0.0140
0.15	0.1460	0.2581	0.1280	0.228	0.1510	0.2405	0.2099	0.0220	0.0270
0.2	0.1700	0.3476	0.1370	0.3174	0.1670	0.3441	0.2940	0.0170	0.0170
0.25	0.2049	0.4700	0.1500	0.4300	0.2099	0.4576	0.37	0.0210	0.0210
0.3	0.2159	0.5745	0.1810	0.5240	0.2119	0.5331	0.4170	0.0190	0.0190
0.35	0.275	0.5756	0.2039	0.5469	0.2750	0.555	0.4280	0.0110	0.0130
0.4	0.3000	0.76	0.2069	0.7255	0.3040	0.8133	0.523	0.0520	0.0080

B.2.2   3D Decision Boundary

In 3D cases, we plot the projection of points in ambient space back to the data manifold $x_1 x_2$-plane. If the decision boundary is fully perpendicular to the $x_1 x_2$-plane, the projection would have a clear separation and matches with the $x_2 = sin(x_1)$ boundary in the manifold. If not, we will have a region close to $x_2 = sin(x_1)$ with mixing red and blue points or the projection does not match with the in-manifold separation.
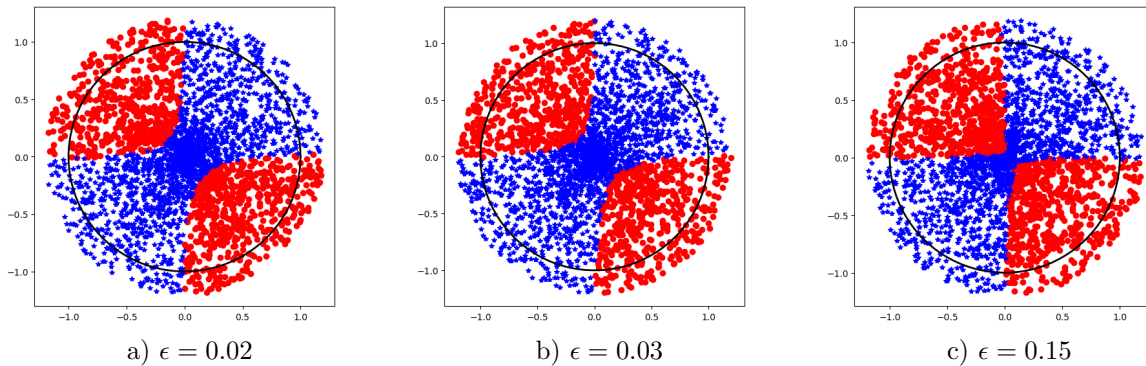
a) $\epsilon = 0.02$ b) $\epsilon = 0.03$ c) $\epsilon = 0.15$

Figure 4: In the graph, we first sample 3000 points in whole space and use f^{nor} to classifier these sampled points. We use red dots and blue dots to mark two different classes. And the decision boundary of the classifier is easy to see in this setting. From left to right, we increase the ϵ budget from 0.02 to 0.15 and use the corresponding normal adversarial data to train the f^{nor}. The decision boundary of f^{nor} is correlated with the size of ϵ.

We show the results in Figure 5. In the single decision boundary case, only f^{nor} has (nearly) perpendicular decision boundary. For f, we can observe that the red points step into the region of the blue points and so does the blue points. And the adversarial training classifier f^{adv} has an even worse result, its decision boundary does not fully match with the $x_2 = sin(x_1)$ curve inside the manifold, which implies that the classifier does not have good standard accuracy, which implies the trade-off between robustness and accuracy for the general robust classifier. And the same results and conclusions hold for the double boundary case.
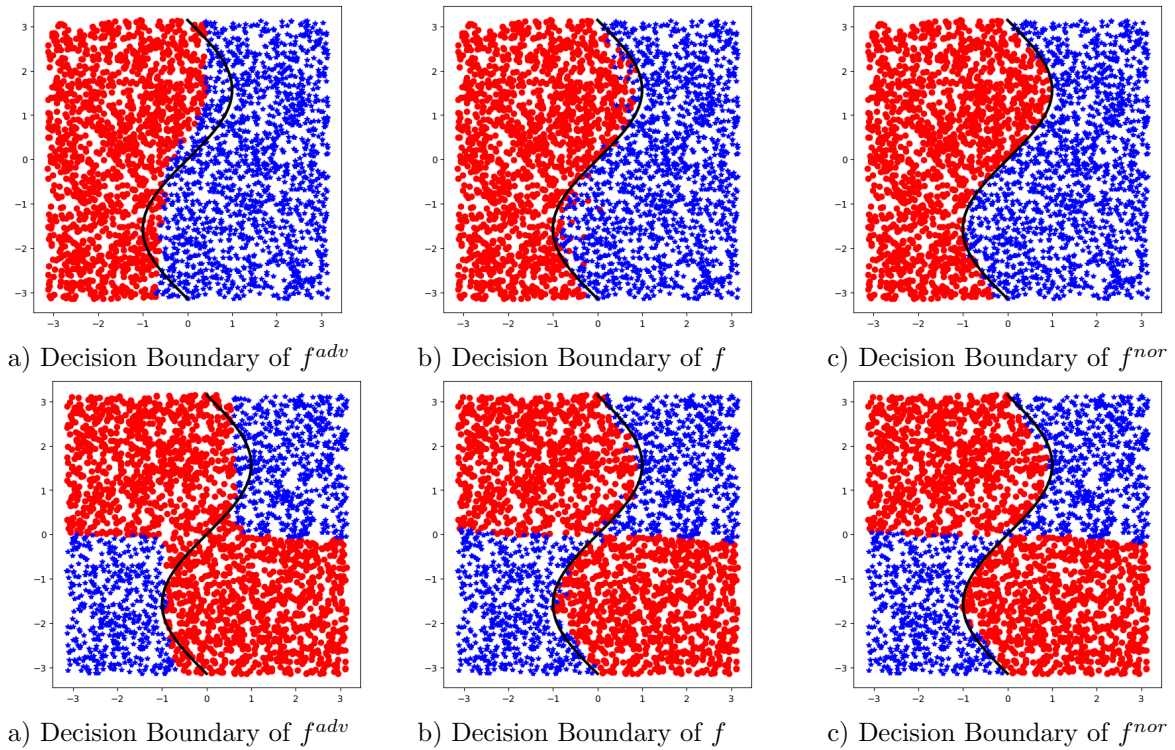
Wenjia Zhang, Yikai Zhang, Xiaolin Hu

a) Decision Boundary of f^{adv}　　b) Decision Boundary of f　　c) Decision Boundary of f^{nor}

a) Decision Boundary of f^{adv}　　b) Decision Boundary of f　　c) Decision Boundary of f^{nor}

Figure 5: In this graph we show the projection of data classified with f, f^{adv} and f^{nor}. We sample 5000 points in the tubular space of x_1x_2-plane. And use f, f^{adv} and f^{nor} to classify these points and mark with red and blue dots. If the point is in the ambient space, we project them back to the x_1x_2-plane. The first row are f, f^{adv} and f^{nor} trained with 3D single boundary synthetic data set. The second row is classifiers trained with the double boundary synthetic data set.