Reinforcing Adversarial Robustness using Model Confidence Induced by Adversarial Training

Xi Wu [*1] **Uyeong Jang** [*2] **Jiefeng Chen** [2] **Lingjiao Chen** [2] **Somesh Jha** [2]

Abstract

In this paper we study leveraging *confidence information* induced by adversarial training to reinforce adversarial robustness of a given adversarially trained model. A natural measure of confidence is $\|F(\mathbf{x})\|_\infty$ (i.e. how confident F is about its prediction?). We start by analyzing an adversarial training formulation proposed by Madry et al.. We demonstrate that, under a variety of instantiations, an only somewhat good solution to their objective induces confidence to be a discriminator, which can distinguish between right and wrong model predictions in a neighborhood of a point sampled from the underlying distribution. Based on this, we propose Highly Confident Near Neighbor (`HCNN`), a framework that combines confidence information and nearest neighbor search, to reinforce adversarial robustness of a base model. We give algorithms in this framework and perform a detailed empirical study. We report encouraging experimental results that support our analysis, and also discuss problems we observed with existing adversarial training.

1. Introduction

In the adversarial-perturbation problem for neural networks, an adversary starts with a neural network model F and a point $\mathbf{x}$ that F classifies correctly (we assume that F ends with a softmax layer, which is common in the literature), and crafts a small perturbation to produce another point $\mathbf{x}'$ that F classifies *incorrectly*. (Szegedy et al., 2013) first noticed the vulnerability of existing neural networks to adversarial perturbations, which is somewhat surprising given their great generalization capability. Since then, a line of research (e.g., (Goodfellow et al., 2014; Papernot et al., 2016b; Miyato et al., 2017; Madry et al., 2017)) has been devoted to hardening neural networks against adversarial perturbation. While modest progress has been made, until now there is still a large gap in successfully defending against more advanced attacks, such as the attack by (Carlini & Wagner, 2017a).

In this paper we study leveraging *confidence information* induced by adversarial training, to reinforce adversarial robustness of a given adversarially trained model. A natural measure of confidence is $\|F(\mathbf{x})\|_\infty$ (i.e. how confident F is about its prediction?). Our motivation comes from a thought experiment based on the manifold assumption (Zhu & Goldberg, 2009) made in unsupervised and semi-supervised learning, which states that *natural data points lie on (or near to) separate low dimensional manifolds for different classes.* If one believes that a good deep neural network model can approximate well the natural manifolds, then ideally it should have the property that it can *confidently* distinguish points from natural manifolds, while not claiming confidence for points that are far away from the natural manifolds. Since natural manifolds are assumed to be separate, it follows that for a good model, *confident model predictions for different classes must be separate from each other.*

Taking this perspective, we formalize a probabilistic property on the separation of confident model predictions for different classes, and use it to examine existing adversarial training formulations. Somewhat surprisingly, we find that a natural min-max formulation proposed by (Madry et al., 2017), under a variety of instantiations, encourages training a model that satisfies our property well. At a high level, the reason for this is that a good solution to the inner maximization of the min-max formulation encourages that there is *no confident wrong* prediction in a neighborhood of a point sampled from the underlying distribution.

Our key observation is that, even if a model is only *a somewhat good* solution under Madry et al.'s adversarial training formulation, it *induces* confidence to be a discriminator which can distinguish between right and wrong model predictions. That is, a model prediction of higher confidence is more likely to be correct. As a result, even though a somewhat good adversarially trained model may still have many wrong predictions *with low confidence*, one can use

[*]Equal contribution [1]Google [2]University of Wisconsin-Madison. Correspondence to: Xi Wu <xiwu@cs.wisc.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

confidence to further reinforce its adversarial robustness. For example, one can now try to protect the model, from making wrong predictions, by *rejecting* a point if model confidence at the point is below certain threshold.

We take a step further to study correcting model predictions at adversarial points, leveraging that model predictions there tend to have lower confidence. We propose Highly Confident Near Neighbor (HCNN), a framework which combines confidence information and nearest neighbor search, to embed a possibly low-confidence point back to high-confidence regions. Specifically, let $\|\cdot\|$ be a norm, and suppose that we want to defend against $\|\cdot\|$-attacks with radius η. Let F be a model with good separation property so that confidence can distinguish right and wrong model predictions in a δ-neighborhood of a point sampled from the underlying distribution. Let $|\cdot|$ be a possibly different norm, and $\xi > 0, \lambda \geq 0$ be two real numbers. The ξ-Highly Confident Near Neighbor, or simply $\texttt{HCNN}_\xi$, computes $\arg\max_{\mathbf{z}\in N(\mathbf{x},\xi)}(\|F(\mathbf{z})\|_\infty - \lambda|\mathbf{z}-\mathbf{x}|)$, where $N(\mathbf{x},\xi)$ is the ξ-neighborhood around $\mathbf{x}$ w.r.t. $\|\cdot\|$. That is, in a ξ-ball centered at $\mathbf{x}$, we encourage finding a point which simultaneously has a high-confidence prediction, and is near to $\mathbf{x}$. Therefore, one can hope to correct adversarial points for $\eta \leq \delta - \xi$. It turns out in our experiments that there are nontrivial settings of δ, η, ξ under which we demonstrate a significant improvement of model robustness over F.

We perform a detailed empirical study over CIFAR10 for ℓ_∞ attacks. We reuse the robust ResNet model trained by Madry et al. as *base model*, and use $\texttt{HCNN}_\xi$. We modify state-of-the-art ℓ_∞ attacks, such as the CW attack (Carlini & Wagner, 2017a), and the PGD attack (Madry et al., 2017), to *exploit confidence information in order to break our method by generating high-confidence attacks*. We first empirically validate that Madry et al.'s model is better, in view of our probabilistic separation property, than models trained without a robustness objective. We then evaluate using confidence to reject adversarial examples, and finally end-to-end defense results. Our results are both encouraging and discouraging: for small radius, we find that confidence is indeed a good discriminator to distinguish right and wrong predictions, and it does improve adversarial robustness. For large radius that approaches .03, however, we find that Madry et al.'s base model loses control there. We further argue that modifications are needed for Madry et al.'s adversarial training in order to defend against more effectively ℓ_∞-attacks of large radius.

In summary this paper makes the following contributions:

- We propose a probabilistic property of separation between confident predictions for different classes. We prove that a natural min-max adversarial training encourages a model that satisfies this property well. As a result, confidence can be used as a discriminator to distinguish between right and wrong predictions, even when a model is only a somewhat good solution to the training objective.
- We propose using confidence induced by adversarial training to further reinforce adversarial robustness of a given adversarially trained model. Specifically, we propose rejecting adversarial points using confidence, and further correcting adversarial points by combining confidence with nearest neighbor search.
- We perform a detailed empirical study to validate our proposal. We report encouraging results that support our analysis, and discuss problems we observed with existing adversarial training.

The rest of the paper is organized as follows: We start with some preliminaries in Section 2. Then Section 3 proposes the probabilistic separation property and we use it to analyze Madry et al.'s adversarial training formulation. We then present embedding objectives and algorithms for handling low-confidence points, and end-to-end instantiations, in Section 4. Section 5 performs a detailed empirical study of our method. We discuss important prior work in Section 6 and conclude in Section 7.

2. Preliminaries

As in existing work, such as (Carlini & Wagner, 2017a; Papernot et al., 2016b), we define F to be a neural network after the softmax layer. With this notation, the final classification is then $C_F(\mathbf{x}) = \arg\max_i F(\mathbf{x})_i$ where $F(\mathbf{x})_i$ gives the confidence of the network in classifying $\mathbf{x}$ for class i. We use $Z(\mathbf{x})$ to denote part of F except the softmax layer. That is, $Z(\mathbf{x})$ computes the *logits* to be fed into the softmax function. Let $\mathcal{C}$ denote the class of all labels. A network is typically parameterized by parameters θ, and in this case we denote by F_θ to indicate that F is parameterized by θ. Finally, let $p \in [0,1]$ be a parameter, and $l \in \mathcal{C}$. A point $\mathbf{x}$ is p-confident for label l if $F(\mathbf{x})_l \geq p$. We consider the following adversarial model which is implicit in several previous work (e.g. (Madry et al., 2017)):

Definition 1 (**Adversarial Model**). *Let $\mathcal{D}$ be a data generating distribution over $\mathbb{R}^d$, F be a model, and $\mathcal{S} \subseteq \mathbb{R}^d$ be a set of perturbations. We consider the following game between an adversary and a defender:*
Adversary*: the adversary draws $\mathbf{x} \sim \mathcal{D}$, produces $\mathbf{x}' = \mathbf{x} + \Delta$ for some $\Delta \in \mathcal{S}$, and sends $\mathbf{x}'$ to the defender.*
Defender*: the defender outputs a label $l \in \mathcal{C}$ for $\mathbf{x}'$.*
The defense succeeds if $C_F(\mathbf{x}) = C_F(\mathbf{x}')$.

An important point regarding this adversarial model is that it only considers points on $\mathcal{D}$ or nearby (as specified by the allowable perturbations $\mathcal{S}$), instead of *the entire domain*. While seemingly one should consider every point in the space, this definition better reflects the intuition behind adversarial perturbation problem: that is, for "natural images," the classification should be consistent with respect to a set of "small perturbations."

3. Probabilistic Separation Property and Madry et al.'s Formulation

This section develops the following:

- In Section 3.1, we propose a goodness property of models which states that *confident regions of a good model should be well separated.* We further formalize this as a probabilistic separation property.

- In Section 3.2, we examine existing robust training formulations and demonstrate that a natural min-max formulation of (Madry et al., 2017) encourages very good separation of p-confident points of different classes for large p, but has a weak control of low-confidence wrong predictions due to estimation error.

3.1. Probabilistic Separation Property

The manifold assumption in unsupervised and semi-supervised learning states that *natural data points lie on (or near to) separate low dimensional manifolds for different classes.* Under this assumption what would an ideal model look like? Clearly, we would expect that an ideal model can *confidently* classify points from the manifolds, while *not* claiming confidence for points that are far away from those manifold. Therefore, since by assumption low dimensional manifolds are separate from each other, we would expect that confident correct predictions of different classes are separate from each other. Therefore, we propose the following goodness property

Confident predictions of different classes should be well separated.

We formalize this as the following property:

Definition 2 ((p, q, δ)-separation)**.** *Let $\mathcal{D}$ be a data generating distribution, $p, q \in [0, 1]$, $\delta \geq 0$, and $d(\cdot, \cdot)$ be a distance metric. Let $\mathcal{B}$ be the event $\{\exists y' \neq y, \mathbf{x}' \in N(\mathbf{x}, \delta), F_\theta(\mathbf{x}')_{y'} \geq p\}$. F is said to have (p, q, δ)-separation if $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{B}\right] \leq q$, where $N(\mathbf{x}, \delta) = \{\mathbf{x}' \mid d(\mathbf{x}, \mathbf{x}') \leq \delta\}$.*

This definition says that for a point $(\mathbf{x}, y)$ sampled from the data generating distribution, in its neighborhood there should *not* be *confident* wrong predictions (though wrong model predictions with low confidence may still exist). Thus, if a model satisfies this definition well, then confident predictions of different classes must be well separated. Finally, this definition allows certain points to be "bad."

3.2. An Analysis of the Min-Max Formulation of (Madry et al., 2017)

This section gives an analysis of the min-max formulation proposed by (Madry et al., 2017). Essentially, our analysis shows that a variety of its instantiations encourage training a model that satisfies Definition 2. More specifically, we prove two things: (1) the formulation encourages training a model that satisfies Definition 2, and (2) complementing this, we show that the formulation may have a weak control of points with low-confidence but wrong predictions, due to estimation error with finite samples. To start with, Madry et al. proposes the following formulation:

$$\begin{aligned}&\text{minimize} \quad \rho(\theta),\\ &\text{where } \rho(\theta) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\max_{\Delta\in\mathcal{S}} L(\theta, \mathbf{x}+\Delta, y)\right],\end{aligned} \tag{1}$$

where $\mathcal{D}$ is the data generating distribution, $\mathcal{S}$ is set of allowed perturbations (e.g., $\mathcal{S} = \{\Delta \mid \|\Delta\|_\infty \leq \delta\}$), and $L(\theta, \mathbf{x}, y)$ is the loss of θ on $(\mathbf{x}, y)$. For the rest of the discussion, we denote $\kappa(\theta, \mathbf{x}, y) = \max_{\Delta\in\mathcal{S}} L(\theta, \mathbf{x}+\Delta, y)$,

Analyzing a Family of Loss Functions. Let us consider the following family of loss functions,

$$\mathcal{L} = \big\{L(\theta, \mathbf{x}, y) \text{ where } L(\theta, \mathbf{x}, y) \text{ is monotonically decreasing in } F_\theta(\mathbf{x})_y.\big\}$$

In other words, as we have higher confidence about the correct prediction, we have smaller loss. For $L \in \mathcal{L}$, there is a monotonically decreasing *loss-lower-bound function* $\tau_L : [0, 1] \mapsto \mathbb{R}^{\geq 0}$, so that if $F_\theta(\mathbf{x})_y \leq q$, then $L(\theta, \mathbf{x}, y) \geq \tau_L(q)$.

Example 1 (**Cross Entropy Loss**)**.** *Let $L(\theta, \mathbf{x}, y) = \mathrm{H}(\mathbf{1}_y, F_\theta(\mathbf{x})) = -\log F_\theta(\mathbf{x})_y$, be the cross entropy between $\mathbf{1}_y$ and $F_\theta(\mathbf{x})$, where F_θ is the model instantiated with parameters θ, and $\mathbf{1}_y$ is the indicator vector for label y. Then we can define τ_L as $\tau_L(\alpha) = -\log\alpha$.*

With these notations we have the following proposition,

Proposition 1. *Let $L \in \mathcal{L}$, τ_L be a loss-lower-bound function, and $\mathcal{B}$ be the event $\{(\exists y' \neq y, \mathbf{x}' \in \mathbf{x}+\mathcal{S})\ F_\theta(\mathbf{x}')_{y'} \geq p\}$. If $\rho(\theta) \leq \varepsilon$, then $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathcal{B}\right] \leq \frac{\varepsilon}{\tau_L(1-p)}$. That is, the probability of an $\mathbf{x}' \in \mathbf{x}+\mathcal{S}$ that is p-confident on a wrong label decreases as $p \to 1$.*

Proof. Let $\alpha = 1 - p$. If $\mathcal{B}$ happens, then $F_\theta(\mathbf{x}')_y \leq \alpha$ for some $\mathbf{x}' \in \mathbf{x} + \mathcal{S}$, and $\kappa(\theta, \mathbf{x}, y) \geq \tau_L(\alpha)$, therefore[1],

$$\begin{aligned}\Pr_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{B}] \leq & \Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\kappa(\theta, \mathbf{x}, y) \geq \tau_L(\alpha)\right] \\ \leq & \frac{\mathbb{E}[\kappa(\theta, \mathbf{x}, y)]}{\tau_L(\alpha)} \qquad \text{(Markov's inequality)} \\ \leq & \frac{\varepsilon}{\tau_L(\alpha)}.\end{aligned}$$

The proof is complete. □

This immediately generates the following corollary,

Corollary 1. *Let $\mathcal{S}$ be a region defined as $\{\Delta \mid d(\Delta, \mathbf{0}) \leq \delta\}$. If $\rho(\theta) \leq \varepsilon$, then the model F_θ is $(p, \frac{\varepsilon}{\tau_L(1-p)}, \delta)$-separated.*

This indicates that even if a model is only a somewhat good solution to (1), meaning $\rho(\theta) \leq \varepsilon$ for only a somewhat small ε, then p-confident points will be well separated (in the sense of Definition 2) as soon as p increases.

The above proposition considers a situation where we have a confident but wrong point (with confidence p). What about points that have wrong but low-confidence predictions? For example, the confidence on the wrong label is only $\frac{1}{2} + \nu$ for some small ν? Note that by setting $p = 1/2$ we derive immediately a bound $\varepsilon/\tau_L(1/2)$ (which can be much weaker than $\varepsilon/\tau_L(1-p)$ for large p). We note that this bound is tight without further assumptions:

Proposition 2. *Let F_θ be a neural network parameterized by θ, and C_{F_θ} be the corresponding classification network. If $\rho(\theta) \leq \varepsilon$, then $\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\exists y' \neq y, \mathbf{x}' \in \mathbf{x} + \mathcal{S}),\ C_{F_\theta}(\mathbf{x}') = y'\right] \leq \frac{\varepsilon}{\tau_L(1/2)}$. The bound is tight.*

Proof. Let $\mathcal{B}$ be the event $\{(\exists y' \neq y, \mathbf{x}' \in \mathbf{x} + \mathcal{S})\ C_{F_\theta}(\mathbf{x}') = y'\}$. If $\mathcal{B}$ happens then $F_\theta(\mathbf{x}')_y \leq \frac{1}{2}$ (otherwise $\mathbf{x}'$ will be classified as y), and so $\kappa(\theta, \mathbf{x}, y) \geq \tau_L(1/2)$. On the other hand, if $\mathcal{B}$ does not happen, then we can lower bound $\kappa(\theta, \mathbf{x}, y)$ by 0. Therefore $\varepsilon \geq \mathbb{E}[\kappa(\theta, \mathbf{x}, y)] \geq \Pr[\neg\mathcal{B}] \cdot 0 + \Pr[\mathcal{B}] \cdot \tau_L(1/2) = \Pr[\mathcal{B}] \cdot \tau_L(1/2)$. Tightness follows as we can force equality for each of the inequalities. The proof is complete. □

Instantiations. The above two propositions can be specialized to many different forms of L. Basically, any loss function that encourages high confidence on the correct label, when plugged into (1), encourages that there is no wrong prediction with high confidence. We list a few of them:

- ***Cross Entropy Loss***. If we use $L(\theta, \mathbf{x}, y) = -\log F(\mathbf{x})_y$, we immediately get separation guarantee $-\frac{\rho(\theta)}{\log(1-p)}$ in Proposition 1.
- ***Squared Loss***. It is also natural to consider $L(\theta, \mathbf{x}, y) = (1 - F_\theta(\mathbf{x})_y)^2$. This then gives separation guarantee $\rho(\theta)/p^2$. Note that this guarantee is weaker than the one above with cross entropy: As $p \to 1$, the probability of bad events happening only converges to $\rho(\theta)$, instead of 0.
- ***Entropy Regularization***. In (Miyato et al., 2017) the authors proposed to use the entropy of $F(\mathbf{x})$, $\mathrm{H}(F(\mathbf{x}))$, as a regularizer. We note that such regularization is compatible with our argument here. Essentially as one increases $F(\mathbf{x})_y$, the entropy decreases.
- ***Loss functions in Carlini-Wagner Attacks.*** It is also not hard to check that any objective function used for targeted attacks in (Carlini & Wagner, 2017a) (see Objective Function on pp. 6 of their paper) can be adapted as loss function L in the min-max paradigm to draw similar conclusions.

Contrasting Proposition 1 and 2, we note the following:

- We note that in reality we only have a finite-sample approximation for (1). Therefore, due to estimation error of solving the objective, even though one can hope for a good separation between p-confident points from different classes for large p (by Proposition 1), there may still be many *low-confidence wrong* predictions (by Proposition 2).
- Our analysis indicates that even if the model is only a somewhat good solution to Madry et al.'s formulation, confidence gives additional information, and is provably a *discriminator* which can distinguish between right and wrong predictions in a neighborhood of a point. Therefore in this situation, one can leverage confidence to *reinforce robustness of the model*. For example, we can now try to apply confidence to protect a model with good separation property, from making wrong predictions, by *rejecting adversarial points.* Specifically, with an appropriately chosen threshold p_0, we can modify the model so that we output $\perp$ if $\|F(\mathbf{x})\|_\infty < p_0$, otherwise we output $C_F(\mathbf{x})$. In this way, we hope that we can correctly predict on natural points while rejecting adversarial points. In the next section, we investigate further combining confidence and search to correct adversarial examples.

4. From Confidence to Defenses

Our analysis from the previous section shows that there can be good models with well-separated confident regions,yet there may be a weak control of points that have wrong predictions but low model confidence. However, since now

[1] Let X be a nonnegative random variable and $a > 0$, Markov's inequality says that $\Pr[X \geq a] \leq \mathbb{E}[X]/a$.

right and wrong predictions are distinguishable by confidence, one can try to correct an adversarial point by embed it back to high-confidence regions. In the following we establish embedding and end-to-end instantiations.

Embedding Objectives. We propose a framework, Highly Confident Near Neighbor ($\mathtt{HCNN}$), for embedding:

Definition 3 (**Highly Confident Near Neighbor**). *Let F be a model, $\|\cdot\|, |\cdot|$ be two possibly different norms, and $\xi > 0, \lambda \geq 0$ be two real numbers. The ξ-*Highly Confident Near Neighbor *objective (or simply $\mathtt{HCNN}_\xi$) computes:*

$$\mathtt{HCNN}_\xi(F, \mathbf{x}) \equiv \underset{\mathbf{z} \in N(\mathbf{x}, \xi)}{\arg\max}(\|F(\mathbf{z})\|_\infty - \lambda |\, \mathbf{z} - \mathbf{x}\,|). \tag{2}$$

where $N(\mathbf{x}, \xi)$ is the ξ-ball around $\mathbf{x}$ with respect to $\|\cdot\|$.

In short, in a ξ-ball centered at $\mathbf{x}$, this objective tries to encourage finding a point which simultaneously has a high-confidence prediction, and is near to $\mathbf{x}$. We have two immediate variants. First, by setting $\lambda = 0$, we have ξ-Most Confident Neighbor ($\mathtt{MCN}_\xi$):

Definition 4 (**Most Confident Neighbor**). *Let F be a model, $\|\cdot\|$ be a norm, and $\xi > 0$ be a real number. The ξ-*Most Confident Neighbor *objective (or simply $\mathtt{MCN}_\xi$) computes:*

$$\mathtt{MCN}_\xi(F, \mathbf{x}) \equiv \underset{\mathbf{z} \in N(\mathbf{x}, \xi)}{\arg\max} \|F(\mathbf{z})\|_\infty. \tag{3}$$

where $N(\mathbf{x}, \xi)$ is the ξ-ball around $\mathbf{x}$ with respect to $\|\cdot\|$.

Second, by putting a threshold on $\|F(\mathbf{z})\|_\infty$ and minimizing $|\,\mathbf{z} - \mathbf{x}\,|$ alone, we have p-Nearest Confident Neighbor:

Definition 5 (**Nearest Confident Neighbor**). *Let F be a model, $\|\cdot\|, |\cdot|$ be two possibly different norms, and $p \in (0, 1)$ be a real number. The p-*Nearest Confident Neighbor *objective (or simply $\mathtt{NCN}_p$) computes:*

$$\begin{aligned} \mathtt{NCN}_p(F, \mathbf{x}) \equiv & \underset{\mathbf{z} \in N(\mathbf{x}, \xi)}{\arg\min} |\,\mathbf{z} - \mathbf{x}\,|, \\ & \textit{subject to } \|F(\mathbf{z})\|_\infty \geq p. \end{aligned} \tag{4}$$

where $N(\mathbf{x}, \xi)$ is the ξ-ball around $\mathbf{x}$ with respect to $\|\cdot\|$.

Algorithms. For the rest of the paper we focus on solving $\mathtt{HCNN}_\xi$ (and hence $\mathtt{MCN}_\xi$ as well). To start with, we note that the optimization for solving $\mathtt{HCNN}_\xi$ is nothing but for each label $l \in \mathcal{C}$ we try to find

$$\mathbf{z}^{(l)} = \underset{\mathbf{z} \in N(\mathbf{x}, \xi)}{\arg\max}(F(\mathbf{z})_l - \lambda |\,\mathbf{z} - \mathbf{x}\,|) \tag{5}$$

and then compute $\mathbf{z}^{(l^*)}$ for $l^* = \arg\max_l F(\mathbf{z}^{(l)})_l - \lambda |\,\mathbf{z}^{(l)} - \mathbf{x}\,|$. (5) can be solved using any preferred gradient-based optimization (e.g., projected gradient descent (Nocedal & Wright, 2006)). This gives Algorithm 1.

Algorithm 1 Solving $\mathtt{HCNN}_\xi$ by solving for each label.

Input: $\mathbf{x}$ a feature vector, $\xi > 0$ a real parameter, $\lambda \geq 0$ a real parameter, a base model F, any gradient-based optimization algorithm $\mathcal{O}$ to solve the constrained optimization problem defined in (5).

function $\mathrm{OracleHCNN}(\mathbf{x}, \xi, F)$
  for $l \in \mathcal{C}$ **do**
    $\mathbf{z}^{(l)} \leftarrow \mathcal{O}(\mathbf{x}, F, l)$
  return $\mathbf{z}^{(l^*)}$ where

$$l^* = \underset{l \in \mathcal{C}}{\arg\max} \left(F(\mathbf{z}^{(l)})_l - \lambda |\,\mathbf{z}^{(l)} - \mathbf{x}\,| \right).$$

Attacks as Defenses. Note that solving (5) is similar to *a targeted adversarial attack* which modifies $\mathbf{x}$ to increase the confidence on target label t, while maintaining small distance to $\mathbf{x}$. Thus *an adversarial attack*, such as those proposed in (Carlini & Wagner, 2017a), can be used as $\mathcal{O}$.

Entropy Approximations. Algorithm 1 may be inefficient. To mitigate this issue, one can replace $\|F(\cdot)\|_\infty$ by entropy functions, which are then differentiable and so we can directly apply gradient descent. We have the following:

Shannon entropy approximation. Approximate $\|F(\mathbf{z})\|_\infty$ by $\mathrm{H}(F(\mathbf{z})) = -\sum_{l \in \mathcal{C}} F(\mathbf{z})_l \log F(\mathbf{z})_l$.

α-Rényi entropy approximation. For $\alpha \geq 0, \alpha \neq 1$, approximate $\|F(\mathbf{z})\|_\infty$ by $\mathrm{H}_\alpha(F(\mathbf{z})) = \frac{\alpha}{1-\alpha} \log(\|F(\mathbf{z})\|_\alpha)$, where $\|F(\mathbf{z})\|_\alpha$ is the α-norm of $F(\mathbf{z})$.

Essentially, these approximations share some of the most important optima as $\|F(\mathbf{z})\|_\infty$: for example if the probability vector $F(\mathbf{z})$ concentrates on one coordinate.

Putting Things Together. Let F be a base model, H_ξ be an embedding algorithm (e.g. $\mathrm{OracleHCNN}$). Then the end-to-end model is $\Gamma(\mathbf{x}) = F(H_\xi(\mathbf{x}, F))$, for some appropriately chosen ξ. We next show how to set parameters (ξ, δ, etc.). Consider an adversary who wants to perform an $\|\cdot\|$-attack with parameter η. Suppose that we decide to use $\mathtt{HCNN}$ with parameter ξ. Note that the guarantee that confidence being a discriminator (Proposition 1) holds in a neighborhood of radius δ. Thus to have the guarantee hold in the search of radius ξ, we must have $\delta \geq \xi + \eta$.

Suppose that F satisfies (p, q, δ)-separation. We say that $(\mathbf{x}, y) \sim \mathcal{D}$ is (p, δ)-*good* if $\{(\forall y' \neq y, \mathbf{x}' \in N(\mathbf{x}, \delta)), F(\mathbf{x}')_{y'} < p\}$. We note that $\Gamma_\xi^{\mathtt{MCN}}(\cdot) \equiv F(\mathtt{MCN}_\xi(\cdot))$ will output a correct prediction at a point $\mathbf{z} \in N(\mathbf{x}, \delta)$, if $\mathbf{z}$ satisfies the following definition,

Definition 6 (**($p, \mathtt{MCN}_\xi$)-goodness**). *A point $\mathbf{z}$ is said to be $(p, \mathtt{MCN}_\xi)$-good if there is a point in $N(\mathbf{z}, \xi)$ that is p-confident for the correct label.*

We can further show that $\Gamma_\xi^{\mathtt{MCN}}$ improves separation:

Proposition 3 ($\mathtt{MCN}_\xi$ **improves separation property**). *Suppose that F satisfies (p, q, δ)-separation. Let η, ξ satisfy that $\eta + \xi \leq \delta$. If for every (p, δ)-good point $(\mathbf{x}, y)$, we have that every point $\mathbf{z} \in N(\mathbf{x}, \eta)$ is $(p, \mathtt{MCN}_\xi)$-good, then $\Gamma_\xi^{\mathtt{MCN}}$ satisfies $(1-p, q, \eta)$ separation.*

Proof. By contraposition it suffices to prove the following

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[(\forall y' \neq y, \mathbf{x}' \in N(\mathbf{x}, \eta)), \Gamma_\xi^{\mathtt{MCN}}(\mathbf{x}')_{y'} < 1-p\right] \geq 1-q.$$

By assumption that F satisfies (p, q, δ)-separation, with probability at least $1-q$ that $(\mathbf{x}, y) \sim \mathcal{D}$, $(\mathbf{x}, y)$ is (p, δ)-good. For every such (p, δ)-good point $(\mathbf{x}, y)$, by assumption, every $\mathbf{z} \in N(\mathbf{x}, \eta)$ is $(p, \mathtt{MCN}_\xi)$-good. Therefore for every such $\mathbf{z}$, $\Gamma_\xi^{\mathtt{MCN}}(\mathbf{z})_y \geq p$, and so $(\forall y' \neq y)$, $\Gamma_\xi^{\mathtt{MCN}}(\mathbf{z})_{y'} < 1-p$. The proof is complete. □

That is, $\Gamma_\xi^{\mathtt{MCN}}$ satisfies a much stronger separation property, though in a smaller η-neighborhood, in the sense that there is no point in the η-neighborhood at which $\Gamma_\xi^{\mathtt{MCN}}$ will assign confidence $\geq 1-p$ to a wrong label.

Finally, as it turns out in our experiments, there are non-trivial settings of δ, η, ξ under which there is a significant improvement of model robustness over the base model.

5. Empirical Study

In this section we perform a detailed empirical study of our method. We call a model "natural" if it is trained in the usual way without considering adversarial robustness, and a model "adversarially trained" if it is trained using the paradigm specified in (1). A key objective of the empirical study is to compare the behavior of confidence of adversarially trained models and that of natural models, as our analysis predicts that confidence will act as a discriminator between right and wrong predictions for adversarially trained models. We ask the following questions:

1. *Does an adversarially trained model satisfy well our probabilistic separation property (Definition 2)?*
2. *Is confidence information effective to help rejecting adversarial examples (e.g., by returning $\perp$), while retaining generalization on points sampled from the underlying distribution?*
3. *Is $\mathtt{HCNN}_\xi$ effective in defending against adversarial perturbations, while retaining generalization capability of the adversarially trained base model?*

Overall Setup. We study the above questions using ℓ_∞ attacks over CIFAR10 (Krizhevsky, 2009). We denote by η the radius parameter of ℓ_∞ attacks. We reuse the ResNet model trained by (Madry et al., 2017) as a representative of an adversarially trained model. This model is trained with respect to an ℓ_∞-adversary with parameter $\delta = 8/256 \approx 0.030$. We denote by ξ the radius parameter of $\mathtt{HCNN}$.

Attacks and Gradient Masking. We note that $\mathtt{HCNN}$ may induce an effect of *gradient masking* (Athalye et al., 2018), and thus may be susceptible to attacks that try to intentionally bypass gradient masking. Taking this into account, we modified existing attacks, including the CW attack in (Carlini & Wagner, 2017a), and the PGD attack (Madry et al., 2017), to *exploit confidence information*. Basically, these modified attacks find adversarial points *with high confidence*, in order to break our defense method which relies on confidence as a discriminator.

Summary of Findings. Our main findings are as follows: **(1)** The behavior of confidence of adversarially trained models, as predicted by our analysis, is significantly better than natural models in distinguishing right and wrong predictions. In particular, within small radius, confidence information of Madry et al.'s model gives a good discriminator to improve adversarial robustness. **(2)** Unfortunately, confidence of the adversarially trained mode by (Madry et al., 2017) loses control as one approaches the .03 boundary. We give further reasoning and argue that modifications to Madry et al.'s formulation seem necessary in order to defend against .03-ℓ_∞ attacks effectively.

	Adv. trained model	Natural model
# adv. points	26	277

Table 1. Results for testing separation property. We randomly sample 300 points. For each column, we report number of samples that we can find attacks of confidence at least 0.9.

Validating the Probabilistic Separation Property. To answer our first empirical question, we statistically estimate $\Pr_{(x,y)\sim\mathcal{D}}\left[(\exists y' \neq y, x' \in N(x, \delta))F(x')_{y'} \geq p\right]$, from Definition 2 and Proposition 1. We compare the *adversarially trained model* in (Madry et al., 2017) and its *natural variant*. Let $\mathcal{B}$ denote the bad event $(\exists y' \neq y, x' \in N(x, \delta))F(x')_{y'} \geq p$. We randomly sample a batch of data points from the test set, and compute frequency that $\mathcal{B}$ happens for each batch. To do so, we generate p*-confident attacks* which are adversarial examples that have model confidence at least p. We use a modified ℓ_∞-CW attack (Carlini & Wagner, 2017a) to search as long as possible to find p-confident adversarial examples that lie within the norm bound $\eta = \delta = 0.030$. Table 1 summarizes the results.

With these statistics we can thus estimate (p, q, δ)-separation (Definition 2) of the models in comparison. We have (details are deferred to Appendix A).

Proposition 4 (Separation from statistics)**.** *Let $\mathcal{B}$ be the*

	AWPR (original)	AWPR (w/ rejection)	Recall of $\mathcal{N}$
$\eta = .010, p = .90$	11%	2.2%	86.7%
$\eta = .010, p = .95$	11%	1.1%	82.5%
$\eta = .010, p = .99$	11%	0.0%	72.6%
$\eta = .015, p = .90$	14%	5.4%	89.6%
$\eta = .015, p = .95$	14%	2.9%	85.6%
$\eta = .015, p = .99$	14%	0.3%	75.9%
$\eta = .020, p = .90$	21.7%	14.7%	90.0%
$\eta = .020, p = .95$	21.7%	9%	85.7%
$\eta = .020, p = .99$	21.7%	3.8%	75.4%
$\eta = .030, p = .90$	34.7%	29.7%	90.0%
$\eta = .030, p = .95$	34.7%	26.8%	86.0%
$\eta = .030, p = .99$	34.7%	16.0%	76.5%

Table 2. Results on rejecting adversarial examples with an adversarially trained model. We report adversarially-wrong-prediction-ratio (AWPR) before applying rejection rule, AWPR after applying rejection rule, and the recall of natural points.

	AWPR (original)	AWPR (w/ rejection)	Recall of $\mathcal{N}$
$\eta = .010, p = .90$	99.3%	99.1%	97.2%
$\eta = .010, p = .95$	99.3%	99.1%	96.2%
$\eta = .010, p = .99$	99.3%	98.9%	92.6%

Table 3. Results on rejecting adversarial examples with a natural model. We only tested $\eta = .010$, the weakest setting of ℓ_∞ attack, and AWPRs are very high, even for $p = .99$. This indicates that the behavior of confidence of a natural model is poor.

event $\big\{(\exists y' \neq y, \mathbf{x}' \in N(\mathbf{x}, \delta))F(\mathbf{x}')_{y'} \geq .9\big\}$. *The following two hold:*

- *With probability at least* $.9$ *the robust model in (Madry et al., 2017) satisfies* $\Pr_{(x,y)\sim\mathcal{D}}\big[\mathcal{B}\big] \leq \frac{14}{75} = 0.18667\ldots$ *That is, the robust model has* $(\frac{9}{10}, \frac{14}{75}, \frac{8}{255})$*-separation.*
- *With probability at least* $.9$ *the natural model satisfies* $\Pr_{(x,y)\sim\mathcal{D}}\big[\mathcal{B}\big] \geq \frac{247}{300} = 0.82333\ldots$ *That is, the natural model does* not *have* $(\frac{9}{10}, q, \frac{8}{255})$*-separation for* $q < \frac{247}{300}$.

Rejecting Adversarial Examples. To answer our second empirical question, we perform the following experiment. We sample 1000 points from the test set, where the base model is correct at, as a set of natural points $\mathcal{N}$. For each point $\mathbf{x} \in \mathcal{N}$, we generate an adversarial example $\mathbf{x}^\star$ whose model confidence is maximal. To do so, we use a strengthened version of the PGD attack used in (Madry et al., 2017) (with ℓ_∞ radius η, 10 random starts, and 100 iterations)[2] to first generate, for each wrong label, an adversarial example whose model confidence is as large as possible. Then among all possible adversarial examples, we pick one that has the largest confidence. We collect all the adversarial $\mathbf{x}^\star$ into $\mathcal{A}$. Note that $|\mathcal{A}| \leq |\mathcal{N}|$ because for each $\mathbf{x} \in \mathcal{N}$ we can generate at most one $\mathbf{x}^\star$ (or we do not find any). Now, for a confidence parameter p, we reject a point $\mathbf{x}$ as adversarial by "$F(\mathbf{x}) < p$," where F is the base ResNet model that is adversarially trained in (Madry et al., 2017). A core metric we report is *Adversarially Wrong Predictions Ratio* (AWPR), the fraction of points where we can generate adversarially wrong predictions. Without the rejection rule, the AWPR is $|\mathcal{A}|/|\mathcal{N}|$, and we hope that this ratio decreases significantly after applying the rejection rule. Meanwhile, we also hope that the *recall* of natural points is large enough which means that we do not reject many natural points.

Table 2 and 3 present the results for adversarially and naturally trained models, respectively. In summary: **(1)** For adversarially trained models, rejection based on confidence gives a statistically significant reduction in AWPR, while having a good recall of retaining natural points. While the results are far from being perfect, they give encouraging evidence that our analysis of Madry et al.'s formulation on confidence holds non-trivially in practice. **(2)** By contrast, rejection based on confidence has no effect at all with a natural model. Even we set $p = 0.99$, there are

[2]The PGD attack used in (Madry et al., 2017) used 1 random start and 20 iterations on CIFAR.

	$\eta = \xi = 0.010$		$\eta = \xi = 0.015$	
	Label change	Confidence reduction	Label change	Confidence reduction
1st confident point	94	8621	114	8511
2nd confident point	208	27	260	20
Other points	50	0	66	0
Missing	0	0	29	0
Total	352	8648	469	8531

Table 4. $\mathtt{MCN}_\xi$ results. 9000 adversarial examples are divided into *label change* (column 1 and 3) and *confidence reduction* (column 2 and 4). Row 1 and 2 give the number of adversarial examples that $\mathtt{MCN}_\xi$ succeeded to narrow down to two most confident predictions over 10 perturbed points. Row 3 gives the number of cases where the correct label is not in the two most confident points, but is one of the model predictions over 10 perturbed points. Row 4 gives cases where all model predictions on $\mathtt{MCN}_\xi$ perturbed points are wrong.

many adversarial examples that reach such confidence using the weakest attack radius .01 in our experiments. This demonstrates that adversarial training prepares confidence in a nontrivial way as a discriminator. **(3)** Finally, we notice that for the adversarially trained model we used, the effectiveness of the rejection rule drops significantly as we approach the .03 boundary. In particular, we are able to find many more high-confidence adversarial points there than neighborhoods of smaller radius.

Defending against adversarial examples using $\mathtt{MCN}_\xi$. We implemented a version of $\mathtt{MCN}_\xi$ and test its performance against ℓ_∞ attacks. We sample a set $\mathcal{N}$ of 1000 random test points where the base model produces *correct* predictions. We use PGD attacks of 1000 iterations to increase the model confidence for target labels as much as possible. With the attack, we first generate 9000 adversarial examples $\mathbf{x}^\star$ for each $\mathbf{x} \in \mathcal{N}$ and each of the 9 incorrect target labels. For a number of cases, the attack achieves different prediction $C_F(\mathbf{x}^\star)$ (*label change*), and for the other cases, the attack fails to change the prediction but reduces prediction confidence of the correct label (*confidence reduction*).

Our implementation of $\mathtt{MCN}_\xi$ solves (5) using the PGD attack with a different setting (ℓ_∞ radius ξ, no random start, and 500 iterations). Table 4 gives results where we report top-2 accuracy. That is, let $\mathbf{x}_1, \ldots, \mathbf{x}_{10}$ be the 10 points perturbed by $\mathtt{MCN}_\xi$ for an input $\mathbf{x}$ with correct label y, sorted by the base model confidence $\|F(\mathbf{x}_1)\|_\infty \geq, \ldots, \geq \|F(\mathbf{x}_{10})\|_\infty$. We count the number of following two cases: **(i)** $y = C_F(\mathbf{x}_1)$, **(ii)** $y \neq C_F(\mathbf{x}_1)$ but $y = C_F(\mathbf{x}_2)$. In summary: **(1)** $\mathtt{MCN}_\xi$ recovers a nontrivial number of correct predictions from adversarial examples with changed labels, without changing the originally correct predictions of adversarial examples with reduced confidences. **(2)** Specifically, for $\eta = \xi = 0.010$, $\mathtt{MCN}_\xi$ produces correct predictions of 94 label-changed adversarial examples, and wrong predictions for 27 confidence-reduced adversarial examples. For $\eta = \xi = 0.015$, $\mathtt{MCN}_\xi$ produces correct predictions for 114 label-changed adversarial examples, and wrong predictions for 20 confidence-reduced adversarial examples. **(3)** Interestingly, for most cases (99.44% for $\eta = \xi = 0.010$, 98.94% for $\eta = \xi = 0.015$), the correct label y can be predicted from the perturbed points $\mathbf{x}_1, \mathbf{x}_2$ with two highest confidences.

Potential Problems with Madry et al.'s Formulation. Our experiments demonstrate that as attack radius increases, the effectiveness of confidence decreases, and it almost completely loses control at the .03 boundary. To this end, we first notice that ℓ_∞ ball of radius .03 is very large: with an attack of radius .03 it is possible to craft a good adversarial point, but one can also arrive at a almost random noise (by perturbing each position of the input image by .03 for example). Given such a large ball to enforce robustness, we observe two potential problems with Madry et al.'s formulation (1): **(1)** *gradient descent is not effective enough to search the space*. We note that the inner maximum of (1) is solved by gradient descent, so probably one can only search a small neighborhood, and make confidence good there. **(2)** *a more fundamental problem is that enforcing (1) in a large ℓ_∞ ball is too much to hope for, because this ball contains random noise anyway.* It seems more reasonable to us that one should only encourage (1) in a relatively small neighborhood (say .01), which builds a *trusted region*, and then encourage *low-confidence* outside the neighborhood, so following confidence information one can hope to go back to trusted regions.

6. Related Work

(Szegedy et al., 2013) first observed the susceptibility of deep neural networks to adversarial perturbations. Since then, a large body of work have been devoted to study hardening neural networks for this problem (a small set of work in this direction is (Goodfellow et al., 2014; Papernot et al., 2016b; Miyato et al., 2017)). Simultaneously, another line of work have been devoted to devise more effective or efficient attacks (a small set of work in this direction is (Moosavi-Dezfooli et al., 2016; Papernot et al., 2016a; Carlini & Wagner, 2017a)). Unfortunately, there is still a large margin to defend against more sophisticated attacks, such as Carlini-Wagner attacks (Carlini & Wagner, 2017a). For example, while the recent robust residual network constructed by (Madry et al., 2017) achieves encouraging robustness results on MNIST, on CIFAR10 the accuracy against a strong adversary can be as low as 45.8%.

A line of recent research investigated using a generative model to "explicitly recognizing" natural manifolds to defend against adversarial perturbations. Examples of this include the robust manifold approach by (Ilyas et al., 2017), and an earlier proposal of MagNet by (Meng & Chen, 2017). Similar to our work, there is a discriminator that can distinguish between right and wrong predictions. However, there is a crucial difference: In generative model approach, one tries to distinguish between points on the manifolds and those that are not, and thus only *indirectly* discriminate between right and wrong predictions. By contrast, our confidence discriminator *directly* discriminates between right and wrong predictions. Therefore, one can think of our approach as a "discriminative" alternative to the "generative" approach studied in robust manifold approach and MagNet, and thus seems easier to achieve. In fact, MagNet has already been successfully attacked by (Carlini & Wagner, 2017b) by exploiting gaps between the generative discriminator and the underlying classifier.

7. Conclusion

In this work we take a first step to study structural information induced by an adversarial training to further reinforce adversarial robustness. We show that confidence is a provable property, once one solves the adversarial training objective of Madry et al. well, to distinguish between right and wrong predictions. We empirically validate our analysis and report encouraging results. Perhaps more importantly, our analysis and experiments also point to potential problems of Madry et al.'s training, which we hope, may stimulate further research in adversarial training.

8. Acknowledgments

This work was partially supported by ARO under contract number W911NF-1-0405.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, 2017a.

Nicholas Carlini and David A. Wagner. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *CoRR*, abs/1711.08478, 2017b.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, 2014.

Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G. Dimakis. The robust manifold defense: Adversarial training using generative models. *CoRR*, abs/1712.09196, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

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

Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In *CCS*, pp. 135–147. ACM, 2017.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *CoRR*, abs/1704.03976, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2574–2582, 2016.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization, second edition*. World Scientific, 2006.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Proceedings of the 1st IEEE European Symposium on Security and Privacy*, Saarbrucken, Germany, 2016a. IEEE.

Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pp. 582–597, 2016b.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

A. Bounding the probability for (p, q, δ)-separation

This section gives details of our estimation of (p, q, δ)-separation from statistics in Table 1. Note that event $\mathcal{E}_b$ corresponds to a Bernoulli trial. Let $X_1, \ldots, X_t$ be independent indicator random variables, where

$$X_i = \begin{cases} 1 & \text{if } \mathcal{E}_b \text{ happens,} \\ 0 & \text{otherwise} \end{cases},$$

and $X = (\sum_{i=1}^{t} X_i)/t$. Recall Chebyshev's inequality:

Theorem 1 (Chebyshev's Inequality)**.** *For independent random variables $X_1, \ldots, X_t$ bounded in $[0, 1]$, and $X = (\sum_{i=1}^{t} X_i)/t$, we have $\Pr[|X - \mathbb{E}[X]| \geq \varepsilon] \leq \frac{\mathrm{Var}[X]}{\varepsilon^2}$.*

In our case, $\mathbb{E}[X] = \mathbb{E}[X_1] = \cdots = \mathbb{E}[X_t]$ and let it be μ, and let the computed frequency be $\hat{\mu}$ (observed value). Thus $\Pr[|\hat{\mu} - \mu| \geq \varepsilon] \leq 1/(4\varepsilon^2 t)$ since $\mathrm{Var}[X] = \mu(1-\mu)/t < 1/4t$. We thus have the following proposition about (p, q, δ)-separation.

Proposition 5. *Let $\alpha, \varepsilon \in [0, 1]$. For sufficiently large t where $\frac{1}{4\varepsilon^2 t} \leq 1 - \alpha$ holds, we have:*

- (Upper bound) *With probability at least α, μ is smaller than $\hat{\mu} + \varepsilon$.*
- (Lower bound) *With probability at least α, μ is bigger than $\hat{\mu} - \varepsilon$.*

For example, we have guarantees for $\alpha = .9$ by putting $\varepsilon = .1$ and $t \geq 250$.