Holistic Approach to Measure Sample-level Adversarial Vulnerability and its Utility in Building Trustworthy Systems

Gaurav Kumar Nayak* Ruchit Rawal* Rohit Lal* Himanshu Patil [1] Anirban Chakraborty
Indian Institute of Science, Bangalore, India
{gauravnayak, ruchitrawal, rohitlal, anirban}@iisc.ac.in [1]hipatil1998@gmail.com

Abstract

Adversarial attack perturbs an image with an imperceptible noise, leading to incorrect model prediction. Recently, a few works showed inherent bias associated with such attack (robustness bias), where certain subgroups in a dataset (e.g. based on class, gender, etc.) are less robust than others. This bias not only persists even after adversarial training, but often results in severe performance discrepancies across these subgroups. Existing works characterize the subgroup's robustness bias by only checking individual sample's proximity to the decision boundary. In this work, we argue that this measure alone is not sufficient and validate our argument via extensive experimental analysis. It has been observed that adversarial attacks often corrupt the high-frequency components of the input image. We, therefore, propose a holistic approach for quantifying adversarial vulnerability of a sample by combining these different perspectives, i.e., degree of model's reliance on high-frequency features and the (conventional) sample-distance to the decision boundary. We demonstrate that by reliably estimating adversarial vulnerability at the sample level using the proposed holistic metric, it is possible to develop a trustworthy system where humans can be alerted about the incoming samples that are highly likely to be misclassified at test time. This is achieved with better precision when our holistic metric is used over individual measures. To further corroborate the utility of the proposed holistic approach, we perform knowledge distillation in a limited-sample setting. We observe that the student network trained with the subset of samples selected using our combined metric performs better than both the competing baselines, viz., where samples are selected randomly or based on their distances to the decision boundary.

1. Introduction

Deep neural networks (DNNs) are becoming increasingly ubiquitous across a plethora of real-world applications such as object detection [33, 37], speech recognition [13, 24], remote sensing [18, 38], etc. However, these models are extremely brittle, as they yield incorrect predictions on samples corrupted by carefully crafted perturbations that are imperceptible to humans, popularly known as adversarial samples [8, 19, 29]. Vulnerability towards adversarial examples (adversarial vulnerability) in state-of-the-art DNNs is particularly worrisome in safety-critical applications such as medical imaging [1, 26], facial recognition [30, 35], self-driving cars [4, 22], etc. For example, adversarially manipulated traffic signs can be misclassified by a self-driving car leading to severe implications like loss of human life.

Recently, researchers have demonstrated [23, 25] that the notion of adversarial robustness also coincides with fairness considerations as certain subgroups (often corresponding to sensitive attributes like gender, race etc.) in a dataset are less robust in comparison to the rest of the data and thus are harmed disproportionately in case of an adversarial attack. A subgroup can broadly be defined as categorization of input-space into disjoint partitions based on certain criteria of interest such as class labels, sensitive attributes, etc. Nanda *et al*. [23] in their work, formulated the phenomena of different levels of robustness exhibited by different subgroups as 'robustness bias'. Moreover, robustness bias is present (and sometimes even amplified) [2, 34] after supposedly making the model 'robust' using state-of-the-art adversarial defense methods like adversarial training. Ideally, a fair model should ensure each (and any) subgroup in the dataset is equally robust. For example, in a face recognition system, the model should perform equally well across different ethnic subgroups.

Accurately measuring the adversarial vulnerability of a sample is the first step towards designing generalizable defense framework that can mitigate robustness bias, as the notion of subgroups may vary significantly depending on the tasks and datasets. Traditionally, this is achieved by using distance to decision boundary (DDB) as a proxy to quantify sample robustness i.e. samples that are farther away from decision boundary are assumed to have higher robustness and vice-versa. Proximity to the decision boundary

*equal contribution.
Webpage: https://sites.google.com/view/sample-adv-trustworthy/

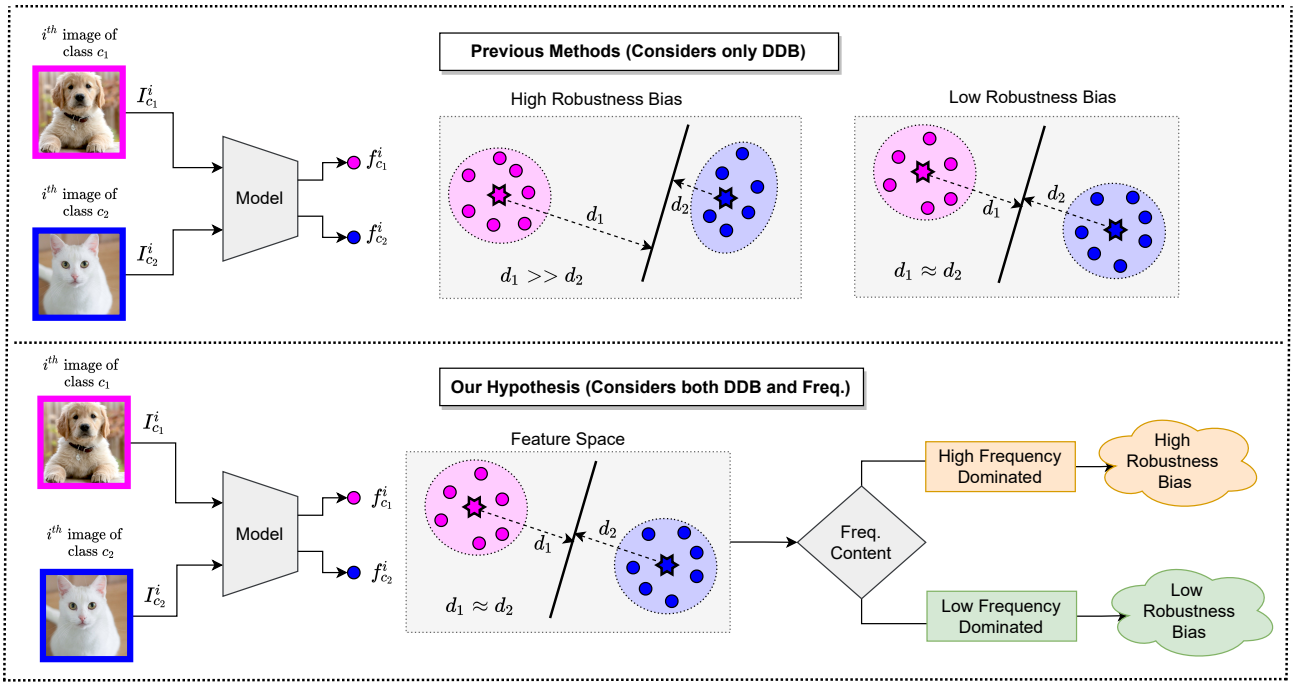

Figure 1. Comparison of existing methods vs our methods: Existing methods characterise robustness bias only via distance to decision boundary (DDB) as the only factor for describing adversarial vulnerability whereas our objective is to incorporate mutlitple perspectives which considers multiple factors (frequency and DDB). Subgroups (shown in pink and blue colour) lying at a similar distance to decision boundary has lower robustness bias compared to subgroups having different DDB (one subgroup lies closer while the other lies farther). Our hypothesis is that even samples lying at similar DDB can have high robustness bias if they are dominated by high frequencies.

is an intuitive measure as samples lying closer to the decision boundary would naturally be more prone to cross the decision boundary after perturbation, eventually leading to misclassification. However, in our analysis (Sec. 4), we find that the samples with similar DDB values can have different adversarial vulnerabilities, i.e., such instances can require varying minimal steps (by an iterative adversarial attack) to fool the model. We observe this trend to be consistent across adversarial attacks. Furthermore, we also provide explanations about these observations (Sec. 5). Hence, unlike [23], we argue based on experimental observations that DDB (although important) alone cannot be the only factor to measure adversarial vulnerability, motivating us to propose a holistic view for estimating adversarial vulnerability by (also) taking other factors into consideration.

Another perspective to quantify adversarial robustness is based on a trained model's reliance on high-frequency features in the dataset. While humans primarily rely on low-frequency (LF) features [32], the DNNs, on the contrary, not only focuses on LF features but can also extract 'useful' predictive features from the high-frequency (HF) components in the data to maximize their performance [31]. Interestingly, many state-of-the-art adversarial attacks primarily perturb the HF component of a sample in order to force erroneous predictions. Thus, the reliance of DNNs on HF features for better generalization makes them highly vulnerable to adversarial attacks [32]. These observations indicate that robustness to adversarial attacks is not only dependent on the distance to decision boundary but also on the nature of the features learned (discussed in Sec. 6).

This work aims to investigate the adversarial vulnerability of a sample from two distinct perspectives, i.e., a) proximity to decision boundary, b) reliance on high-frequency features. Our objective is to provide a holistic estimation, as only relying on proximity to decision boundary might give a false sense of robustness. The difference between existing works and ours is also shown in Figure 1.

Based on the above perspectives, we propose the idea of a trustworthy systems (Figure 5) where the trained models would also have provision to yield trust score besides regular class label predictions. The trust score (quantified using both DDB and HF factor) indicates the reliability of a model's prediction. A test sample is clustered either into 'Trust' or 'Non-Trust' cluster based on its trust score. The predictions are reliable only when the sample falls in the Trust cluster. Otherwise, samples in the Non-Trust cluster can be surfaced to a domain-expert for further inspection and annotation.

An analysis of multiple factors for quantifying adversarial vulnerability can also be utilized for different applications such as knowledge distillation. Specifically, a student net-

work can be trained efficiently in a limited-data setting by selecting the samples based on adversarial vulnerability factors (viz., distance to decision boundary and reliance on high-frequency). We empirically observe better performance by selecting samples based on our proposed criterion, compared to random-baseline and conventional methods (details in Sec. 9).

Our overall contributions are summarized as follows:

- We show that the traditional notion of quantifying adversarial vulnerability, i.e., 'proximity to the decision boundary', is insufficient, leading to a false sense of robustness. We verify it across multiple adversarial attacks and architectures, and provide suitable justification for the same.
- We incorporate multiple perspectives (distance to decision boundary and reliance on high frequency) to characterize robustness bias and thereby provide a more holistic view to estimate adversarial vulnerability. This is still unexplored to the best of our knowledge.
- We build a trustworthy system that predicts the sample-level trust scores based on the multiple factors of sample vulnerability. We further demonstrate that our proposed system alerts the human domain experts when the samples are highly likely to be misclassified (i.e. non-trustable) with better precision than the system designed by using individual factors only.
- As a case study on the task of knowledge distillation under limited data setting, we show the utility of multiple perspectives of adversarial vulnerability in sample selection for composing better transfer set. We obtain improvements in *Student*'s performance across different low-data settings against multiple baselines and this trend remains consistent even when the hyperparameters (e.g. temperature in distillation loss) are varied.

2. Related Works

Several hypothesis for explaining the existence of adversarial samples have been proposed in recent years [5–7, 12, 32]. Most prominently, researchers [12] have attributed the presence of adversarial samples to useful yet brittle i.e. non-robust features in the data that can be easily latched on to, by DNNs. On similar lines, Wang *et al*. [31] demonstrated that a DNN can learn well-generalizable features from high-frequency components in the data. These high-frequency components are shown to be most prone to perturbations (i.e. higher adversarial vulnerability) in case of state-of-the-art adversarial attacks [32].

Fairness approaches aim to ensure that no subgroup is disproportionately harmed (or benefited) due to decisions taken by an automated-decision making system [11, 21, 36]. Recently, Nanda *et al*. [23] demonstrated that adversarial robustness is closely connected to the notion of fairness as different subgroups possess different levels of robustness (robustness bias) leading to unfair scenarios in downstream applications. Furthermore, robustness bias persists [2, 34] even after making the model robust using adversarial defenses like adversarial training.

Traditionally, distance to decision boundary (of a sample) is used as a proxy for measuring it's adversarial vulnerability [23]. However, we argue that such a measure is inadequate as a sample could be adversarially vulnerable due to other factors as well. Thus, our objective is to holistically estimate such adversarial vulnerability by combining multiple perspectives (namely proximity to decision boundary and reliance on high-frequency features) and further show it as a better measure than the individual factors alone.

3. Preliminaries

Notations: The model M is trained on a labelled dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^{N}$ using a standard cross entropy loss ($\mathcal{L}_{ce}$). The dataset $\mathcal{D}$ contains images from p different classes i.e. the image set $I = \{I_{ck}\}_{k=1}^{p}$ and I_{ck} represents set of images from k^{th} class. Each class contains equal amount of samples i.e. N/p. $M(x^i)$ denotes the logits and the class prediction by the model M on the i^{th} input (x^i) is represented by $argmax(softmax(M(x^i)))$.

A_{adv} contains a set of s different adversarial attacks i.e. $A_{adv} = \{A_{adv}^j\}_{j=1}^{s}$. An i^{th} adversarial sample (i.e. $\hat{x}^i$) is obtained by perturbing the sample (x^i) with an objective to fool the network M.

The Discrete Cosine Transform and its inverse operations are denoted by DCT and $IDCT$ respectively. An i^{th} spatial sample (x^i) has f^i as its representation in frequency space.

The model M is made trustworthy where the trust score (denoted by T) of a sample is computed at test time and humans are alerted if the sample falls in the non-trustable cluster.

Threat model: The perturbations added via an adversarial attack $A_{adv}^j \in A_{adv}$ (also called adversarial noise) is constrained in a L_p norm ball to make them human imperceptible i.e. $\left\|\hat{x}^i - x^i\right\|_p \leq \epsilon$. L_∞ and L_2 are the two popular threat models. Besides the constraint on δ, the other objective is to enforce the model M to change its prediction on adversarial sample $\hat{x}^i$ i.e. $argmax(softmax(M(\hat{x}^i))) \neq argmax(softmax(M(x^i)))$. To achieve these, different optimization procedures are followed leading to different adversarial attacks such as PGD [19] and Deepfool [20].

Adversarial Training: At every iteration, the batch of samples from dataset $\mathcal{D}$ is augmented with corresponding adversarial samples, and together they are used to optimize the model parameters by minimizing the loss to obtain a robust model ($\hat{M}$). In PGD adversarial training, the loss taken is the standard cross-entropy loss.

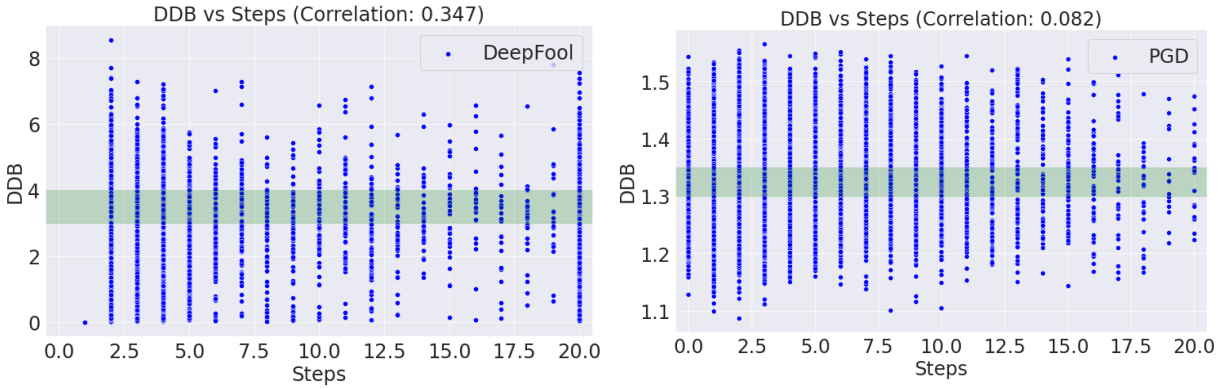

Figure 2. Plots of Distance to Decision Boundary (DDB) vs Steps to attack (Steps) for different attacks, DeepFool (left) and PGD (right) for CIFAR10 dataset on ResNet18 architecture. Even at a similar distance to decision boundary, number of steps required for the adverserial attack may vary from sample to sample.

Distance to decision boundary (DDB): The accurate estimation of the distance from the nearest decision boundary $d_f(x^i)$ of a trained classifier (M or $\hat{M}$), for any input sample $x^i \in \mathcal{D}$, is an extremely challenging task. The strategy adopted by Nanda *et al.* [23] for computing $d_f(x^i)$ is to perturb x^i with the aim of transporting it in the input space to a different category than the original. This can be precisely achieved by a standard adversarial attack setup, wherein an adversarial attack $A^j_{adv} \in A_{adv}$ computes the adversarial sample $\hat{x}^i$. The difference in predicted labels for x^i and $\hat{x}^i$ by the model indicates that the decision boundary is atmost $\delta^i = \left\|\hat{x}^i - x^i\right\|_2$ distance away from x^i, establishing δ^i as a reliable estimate of $d_f(x^i)$.

Discrete Cosine Transform (*DCT*): *DCT* is used to transform the sample from spatial domain to frequency domain i.e. an i^{th} sample of $\mathcal{D}$ (x^i) gets converted to f^i. Unlike Discrete Fourier Transform which outputs a complex signal, DCT outputs only a real-valued signal which makes the analysis of images in the frequency domain much easier. In the frequency transformed image f^i, the top left corner represents the low-frequency content of the image, and the bottom right corner represents the high-frequency contents. In order to retrieve the spatial sample (x^i) back from its corresponding frequency domain representation in (f^i), we use the Inverse Discrete Cosine Transform (*IDCT*).

Flipping Frequency: In order to quantify the reliance of a sample on high-frequencies, we progressively remove high-frequency bands until the model changes it's prediction (w.r.t the prediction on original sample). For instance, if model predicts correctly on the sample containing frequency content till $(k+1)^{th}$ band, and mis-classifies on k^{th} band, this implies that model is dependent on $(k+1)^{th}$ band to make the correct prediction. Hence, if the flipping frequency band ($k+1$ in this case) is high it means that the model relies more on high-frequency content, if flipping frequency is low it means that the model relies on low-frequency.

4. DDB is insufficient to characterize robustness

Fig. 2 contains the plots for model M (Resnet-18) trained on CIFAR-10 dataset. The x-axis represents the minimal steps required by the adversarial attack to change the model prediction. Here, we also varied the adversarial attacks ($s = 2$ i.e. $A_{adv} = \{PGD, Deepfool\}$) which are used to compute steps to attack for each sample in the dataset. The y-axis represents the DDB values for all the samples i.e. $d_f(x^i)_{i=1}^N$. We can observe that samples having similar DDB values can require a varying number of attack steps (an instance is highlighted using a rectangular box in green color). This observation is consistent across different attacks.

More the number of steps taken by the adversarial attack to fool a network, the harder the sample is to attack. In other words, more steps imply less adversarial vulnerability (more robust) and vice versa. This along with preceding discussion also implies that the samples with similar DDB (either in close or far regions), can have different adversarial vulnerabilities. This contradicts the traditional assumption that samples that are far from the decision boundary would always be less vulnerable to adversarial attacks and samples with same DDB cannot have varying vulnerabilities. Hence, in contrast to that assumption, even samples far off from decision boundary can require fewer steps to attack (less vulnerable). Moreover, the correlation between DDB and steps to attack is low. Thus, it suggests that DDB is not a sufficient factor to characterize the robustness/vulnerability associated with a sample.

5. Explanation for DDB as an unreliable estimate of adversarial vulnerability

To compute DDB of a sample x^i (i.e. $d_f(x^i)$), $\delta^i = \left\|\hat{x}^i - x^i\right\|_2$ is chosen as an estimate of DDB. (Refer to Sec. 3). However, as shown in Figure 3, samples having similar δ values can take different optimization paths while

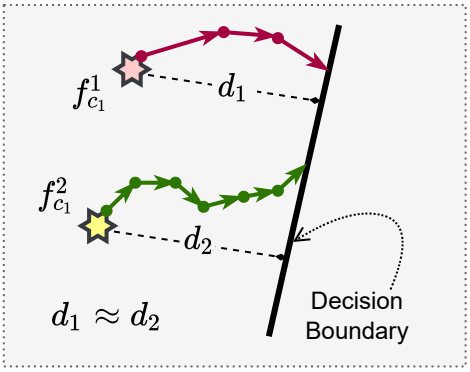

Figure 3. Samples lying to same distance to decision boundary ($f^1_{c_1}$ and $f^2_{c_1}$) can follow different number of optimization steps while generating adversarial samples through adversarial attack (as the path can be highly non-linear), Hence such samples can have diff adv vulnerabilities even after having similar DDB values.

estimating the minimal steps to flip the label via adversarial attack. Hence, samples with similar δ can have different counts of steps to attack, resulting in different adversarial vulnerabilities. We also note that DDB can become a good estimate if the sample path during optimization is linear. But in general, this is rather strong and improbable assumption and the optimization path is highly likely to be non-linear. So, intuitively it may seem that DDB may characterize robustness of a sample, but practically it is not a reliable estimate for deep networks.

6. Different Perspective: Model Reliance on High Frequency Features

In contrast to humans, DNNs heavily rely on high-frequency (HF) features in the data, which makes them susceptible to adversarial attacks [31, 32]. Along similar lines, Xu *et al*. [34] demonstrated that the vulnerability of certain classes (that are inherently more-vulnerable/less-robust to adversarial perturbations) is amplified after adversarial training. We hypothesize that such robust DNN must be relying more on high-frequency features for classes most vulnerable to adversarial attacks.

We empirically validate our hypothesis by calculating the average high-frequency bands required by the model in order to have predictions identical to those original samples for the most and least vulnerable classes (refer Table 1). For instance, class-3 (most-vulnerable class) of the CIFAR-10 dataset, on average, requires 32 (i.e. the maximum frequency band - k_{max}) to ≈ 9 HF bands to have predictions identical to those on original (full-frequency spectrum) samples. On the contrary, class-1 (least-vulnerable class) requires 32 to ≈ 1. Thus, the highly-vulnerable class primarily requires high-frequency information, whereas the least-vulnerable class relies more on low-frequency content. We perform additional analysis (refer Figure 4) to evaluate a robust model's performance on multiple high-frequency bands, i.e., we gradually add frequency bands, starting from the maximum frequency component (i.e. $k_{max} = 32$) to the lowest ($k_{min} = 0$). In line with our previous observations, we note that the robust model achieves fairly decent performance on the class-3 (most vulnerable) with only high-frequency content, compared to class-1 (least vulnerable), which also leverages low-frequency information to predict accurately.

Class	Clean Accuracy	PGD Accuracy	Avg. HF Band Req.
Class-1 (Least Vulnerable)	92.50	64.80	1.06
Class-3 (Most Vulnerable)	48.30	16.40	9.44

Table 1. Performance of least-vulnerable and most-vulnerable class for a Robust ResNet-18 model on the CIFAR-10 dataset. There is significant discrepancy in clean and adversarial performance of both the classes. The more-vulnerable class relies on high-frequency information to predict consistently (w.r.t original samples), whereas the least-vulnerable class relies on low-frequency content.

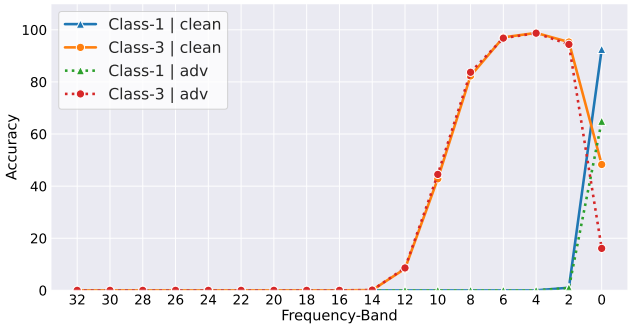

Figure 4. Evaluating robust ResNet-18's performance on multiple frequency bands by gradually adding high-frequency content from the maximum frequency component to the lowest.

7. Combined Study of DDB and Frequency

In the previous sections (Sec. 4, 5 and 6), we motivated the use of different perspectives such as DDB and Frequency towards vulnerability analysis. In order to come up with a more holistic measure to quantify adversarial vulnerability of a sample, we introduce a combined score (T) that appropriately combines the contribution from both the factors.

We first obtain the DDB $d_f(x^i)$ and Flipping Frequency $F(x^i)\ \forall i \in \{1, N\}$ (Refer Sec. 3). Since $d_f(x^i)$ and $F(x^i)$ can be of different range depending on datasets and other various factors, we therefore normalise both the metrics to a common range $[0, 1]$. We call these normalized values as Normalised DDB and Normalised Flipping Frequency respectively.

Normalised Distance to Decision Boundary: To obtain Normalised DDB or $\hat{d}_f(x^i)$ we simply scale the DDB value between 0 to 1 by using Eq. 1.

$$\hat{d}_f(x^i) = \frac{d_f(x^i) - min(D_f(x))}{max(D_f(x)) - min(D_f(x))} \tag{1}$$

Here $min(D_f(x))$ and $max(D_f(x))$ denotes minimum and maximum DDB values respectively, where $D_f(x) = \{d_f(x^1), d_f(x^2), \cdots, d_f(x^N)\}$ for a dataset. Sample farthest from the decision boundary will have $\hat{d}_f(x^i) = 1$ and sample nearest to decision boundary will have $\hat{d}_f(x^i)) = 0$. Samples having $\hat{d}_f(x^i)$ values close to 0 would generally be more prone to adversarial vulnerability.

Normalised Flipping Frequency: A higher value of Flipping Frequency $F(x^i)$ represents that the sample is more dependent on high frequency components and hence is more vulnerable (Refer Sec. 3). This is in contrast to DDB metric where higher value of DDB implies less vulnerability. To have a similar effect in case of frequency aspect as well (i.e. higher value representing lower vulnerability), we define a metric named Reversed Normalised Flipping Frequency which can be computed using Eq. 2).

$$\hat{F}(x^i) = 1 - \frac{F(x^i) - min(\tilde{F}(x))}{max(\tilde{F}(x)) - min(\tilde{F}(x))} \tag{2}$$

where $\tilde{F}(x) = \{F(x^1), F(x^2), \cdots, F(x^N)\}$. $\hat{F}(x^i)$ would be close to 0 for samples which majorly depends on high frequency components and are more prone to adversarial attacks. Similarly, the value of $\hat{F}(x^i)$ would be 1 if it is more adversarially robust.

Combined Score: The essence of computing a combined score T is to capture robustness information from both frequency and DDB. For any sample x^i, the combined score $T(x^i)$ can be simply computed using the harmonic mean of Normalised Distance to Decision Boundary $\hat{d}_f(x^i)$ and Reversed Normalised Flipping Frequency $\hat{F}(x^i)$ (Eq. 3). Here $\epsilon = 10^{-5}$ is a small quantity to ensure numerical stability.

$$T(x^i) = \frac{2 \times \hat{d}_f(x^i) \times \hat{F}(x^i)}{\hat{d}_f(x^i) + \hat{F}(x^i) + \epsilon} \tag{3}$$

The combined score T follows a similar trend compared to other metric, i.e. higher combined score means a lower adversarial vulnerability. We claim this score to be superior than its individual counterparts (verified experimentally in Sec. 8) as the harmonic mean is able to captures the combined effects of fluctuations between these two quantities and hence gives a better estimate of adversarial vulnerability.

8. Building Trustworthy Systems

The combined score as discussed in the previous section is used to quantify the combined effect of both the factors (DDB and frequency). As these factors can better describe adversarial vulnerability of a sample, hence they can be used to determine the trustworthiness of a sample.

High combined score implies high normalized DDB $\hat{d}_f$ (i.e. far from decision boundary) and high reverse normalized flipping frequency ($\hat{F}$) (i.e. more dependency on low frequency features), which in turn leads to high trust. Hence combined score T can be treated as **trust score**. Keeping this in mind, we design a trustworthy system (shown in Figure 5) where the model is allowed to only predict when the samples have high trust (i.e. ensuring reliable predictions from the model). If the trust score is low, the system raises an alert signal to human for further investigation.

To achieve the above objectives i.e. separating out high and low trust samples, we simply use K-means clustering [17] with 2 clusters. The samples with high combined score (trust score) are clustered into the 'trust' cluster and samples with low trust score are assigned to 'non-trust' cluster. Moreover, this approach allows easy integration with any pretrained model.

Overall, during test time, combined score (trust score) is computed for the test sample using which it gets clustered into one of the clusters (either trust cluster or non-trust cluster). If the test sample belongs to trust cluster, then the model outputs the prediction, otherwise the sample falls into non-trust cluster and alert signal is raised to human domain expert for investigation.

To validate our system, we perform experiments on CIFAR-10 [15] (containing 10 classes). We conduct experiments by estimating $\hat{d}_f$ using different adversarial attacks (i.e. PGD and DeepFool). We also vary the architectures for robust (Resnet-18, obtained via adversarial training) and non-robust (Resnet-18 and VGG) settings.

We also compare our performance (combined multiple perspectives) against the trustworthy system when designed using only one factor (i.e. single perspective of adversarial vulnerability such as DDB).

We evaluate our method and other baselines using flagging accuracy which is defined as:

$$\frac{\text{The number of incorrect samples in 'non-trust' cluster}}{\text{Total number of samples in 'non-trust' cluster}} \times 100 \tag{4}$$

A higher flagging accuracy is desired. This implies that most of the samples that are sent to humans were wrongly predicted by the model. This can help in reducing human effort and time as human in loop would mostly get those samples which the model can't handle. The results are reported in Table 2. We can observe that T-Score consistently outperforms its individual counterparts by a significant margin across different attacks (PGD and DeepFool) and architectures (Resnet-18 and VGG-11). Moreover, on an average over different attacks and architectures, our method flags nearly 77% of incorrect predictions which is $\approx 10\%$ higher than the average performance observed on normalized DDB.

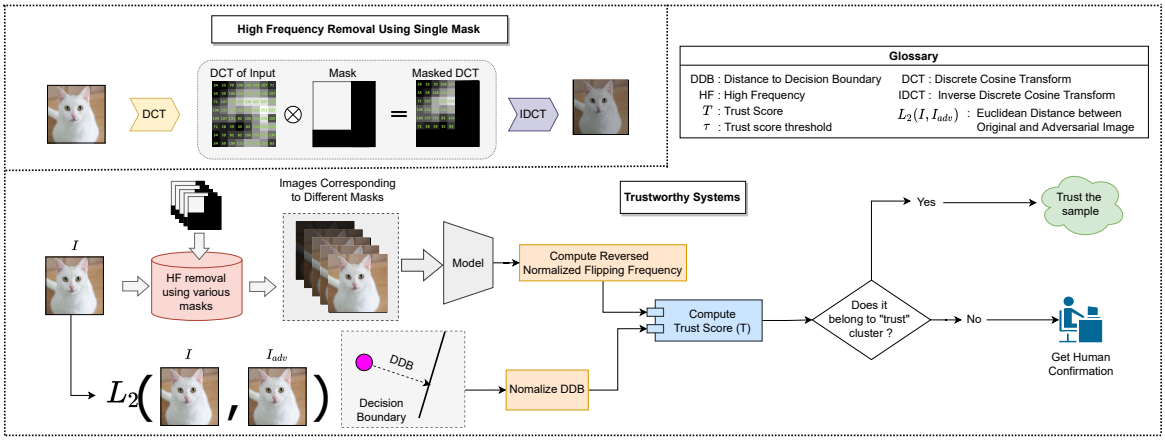

Figure 5. Detailed steps involved in our proposed design of trustworthy systems. (*Top Row*) Method to remove high frequency components using DCT and IDCT. The image is first transformed to frequency domain using DCT and then a binary mask is applied which selects the required frequencies. (*Bottom Row*) For a test sample we compute Reversed Normalized Flipping Frequency (Eq. 2) and Normalized Distance to Decision to Boundary (Eq. 1). Finally, Trust score (Eq. 3) is computed, based on which the test sample is clustered either into "Trust cluster" or "Non-Trust Cluster". Samples that fall into trust cluster are allowed to predict by model, and sample which fall in non-trust cluster are sent to human in form of alert signals for further investigation.

Scores	PGD	DeepFool
$\hat{d}_f$	23.5	32.7
$\hat{F}$	29.8	29.8
T (Ours)	**35.2**	**36.2**

(a) Robust Resnet-18

Scores	PGD	DeepFool
$\hat{d}_f$	8.6	11.5
$\hat{F}$	11.7	11.7
T (Ours)	**12.4**	**13.9**

(b) Resnet-18

Scores	PGD	DeepFool
$\hat{d}_f$	10.2	12.2
$\hat{F}$	14.3	14.3
T (Ours)	**17.6**	**14.6**

(c) VGG-11

Table 2. Flagging accuracy across various architectures (DDB estimated using PGD and DeepFool adversarial attacks) for Normalised DDB ($\hat{d}_f$), Reversed Normalised Flipping Frequency ($\hat{F}$) and T-score (T) on CIFAR10 dataset. It can be observed that T-score outperforms other metrics by a significant margin across various settings.

9. Time Efficient Training of Lightweight models using Knowledge Distillation

Knowledge Distillation (KD) is a common technique to transfer knowledge from a deep network (called *Teacher*) to a shallow network (called *Student*). This is done by matching certain statistics between *Teacher* and *Student* networks. The statistics can be logits [3], temperature raised softmaxes [10], intermediate feature responses [27], Jacobian [28] etc. It has been popularly used in model compression for classification tasks. The networks after compression become lightweight (less memory footprint) and suitable (less computation and less inference time) to be deployed onto the portable devices.

Most of the existing works use the entire training data to conduct knowledge transfer. As a consequence of this, the training time is quite high. This problem becomes even compounded when large scale training sets are used for training on large architectures. Many times in such situations, sophisticated hardware would be needed for training. Moreover, finding the optimal hyperparameters by running the model several times can become infeasible at times. Therefore, there is a need to do fast training to overcome such issues.

A child quickly learns from just a few examples. Based on this analogy, one way of reducing time required is to do training with few selected samples. But if we randomly select those samples, it can result in a significant drop in performance. Therefore, we need to intelligently select the required samples so that we can optimally trade off between training time and distillation accuracy. In other words, our goal is to do *"Quick Training for Fast Knowledge Distillation by only utilizing few training samples"*. There are few works where distillation is performed using few training examples by modifying the *Student*'s architecture [16] or augmenting pseudo data with few training samples [14]. But unlike these works, we do not aim to modify *Student*'s architecture or generate pseudo samples which may bring good performance at the cost of large training time.

$$\mathcal{L} = \sum_{(x,y)\in\hat{\mathbb{D}}} (1-\lambda)\mathcal{L}_{kd}(St(x;\theta_S,\tau), Te(x;\theta_T,\tau)) + \lambda\mathcal{L}_{ce}(\hat{y}_S, y) \tag{5}$$

Let the optimal transfer set ($\hat{D}$) contain the selected training samples which is $k\%$ of entire training data such that the car-

dinality of entire dataset (D) would be much greater than the cardinality of $\hat{D}$ i.e. $|D| \gg |\hat{D}| = k\%$ of D. The distillation can then be performed on samples of $\hat{D}$ as in eq. 5. Here, Te and St denotes *Teacher* and *Student* networks with parameters θ_T and θ_S respectively. We use a robust ResNet-18 network as our *Teacher* model (Te) and MobileNet-V2 as our *Student* network (St). L_{kd} represents distillation loss while cross entropy is denoted by L_{ce}. The temperature (τ) and λ are hyperparameters.

In order to carefully select only a subset of training samples and use them as a transfer set to conduct the distillation, we take into account the factors discussed in previous subsections i.e. sample's distance to decision boundary and model's frequency reliance on that sample for its prediction. For a limited data budget, we rank all the samples in the training data according to our trust-score metric (described in Sec 8 ; with respect to the teacher model) and select a fixed amount of most-trustworthy samples for each class. The student model is then trained on those trust-worthy samples directly using standard distillation losses (described in Eq. 5). We adopt a (uniform) random sample selection strategy as our baseline, where we randomly select a fixed number of samples from each class. Furthermore, the samples lying close to the teacher model's decision boundary are often important for the student network to successfully mimic the teacher's decision boundary [9]. Hence, we also report the student model's performance when the samples closest (for each class separately) to the decision boundary are selected. We perform experiments with multiple sample selection budgets (i.e. 100, 120 and 150 samples per class).

In Table 3 we observe that our proposed trust-worthy sample selection strategy consistently outperforms both the random baseline and the conventional distance to decision boundary across multiple sample selection budgets. In Figure 6 we further analyze the flagging accuracy performance (normalized DDB and our trust score metrics) for the best performing *Student* models (i.e. models distilled via our trust-worthy sample selection strategy) across different sample-budgets. We observe that even on a specific application (KD in this case), our flagging accuracy using trust score is consistently better (or atleast similar) than the performance on normalized DDB. Apart from this, even the models trained via suboptimal sample selection strategies (such as random and closest to DDB), we obtained better performance. For instance, we notice an improvement in flagging accuracy from 74.0% to 75.83% on random baseline and 73.41% to 75.94% on closest to DDB (estimated using PGD) with a budget of 100 samples per class. Similar trend is followed across other sample budgets.

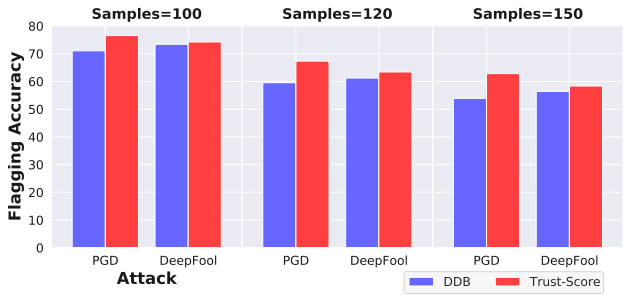

Figure 6. Flagging accuracy comparison for (Normalized) DDB score v/s (Our Proposed) Trust-Score (flagging) metrics across various attacks (PGD and DeepFool) and multiple per-class sample-selection budgets (100, 120, 150) for the model trained via our Trust Score based sample selection strategy.

Samples Selection (Per Class)	**Sample Selection Criteria**	$\tau = 30.0$ $\lambda = 0.2$	$\tau = 8.0$ $\lambda = 0.2$
100	Random	27.06	25.42
	Closest to DDB (Conventional)	24.19	23.4
	Ours (Trust Score based)	**41.63**	**36.65**
120	Random	37.48	22.98
	Closest to DDB (Conventional)	24.34	20.90
	Ours (Trust Score based)	**44.64**	**44.52**
150	Random	46.10	41.59
	Closest to DDB (Conventional)	22.87	24.92
	Ours (Trust Score based)	**50.38**	**50.73**

Table 3. Performance of *Student* (MobileNet-V2) when different budgets of samples (per-class) are selected, using different sample selection strategies on CIFAR-10 dataset.

10. Conclusion

We presented the traditional factor for quantifying adversarial vulnerability (i.e. distance to decision boundary) and discussed its limitations (i.e. samples having similar distance from decision boundary can have different vulnerabilities). To overcome this, we proposed a holistic view by considering other factors such as model reliance on high frequency features for its prediction. We combined the multiple perspectives to propose a holistic metric that can be used to quantify sample trust. Hence, we designed trustworthy systems using the holistic metric that yields reliable predictions and provides alert signals to human domain-expert whenever encountering non-trustable samples with better precision than individual factors considered alone. We also showed the utility of multiple factors that characterise adversarial vulnerability in a knowledge distillation setup, where the student network can be trained efficiently in a limited data-setting using the samples selected based on the vulnerability factors. In the future, we plan to extend this work by exploring other aspects of adversarial vulnerability beyond distance to decision boundary and high-frequency reliance, and investigating their utility across different applications.

References

[1] Andrea Barucci and Emanuele Neri. Adversarial radiomics: the rising of potential risks in medical imaging from adversarial learning, 2020. 1

[2] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. In Luca Bertinetto, João F. Henriques, Samuel Albanie, Michela Paganini, and Gül Varol, editors, *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, volume 148 of *Proceedings of Machine Learning Research*, pages 325–342. PMLR, 11 Dec 2021. 1, 3

[3] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 7

[4] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 1

[5] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. *arXiv preprint arXiv:1802.08686*, 2018. 3

[6] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 3

[7] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018. 3

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[9] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3771–3778, 2019. 8

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 7

[11] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020. 3

[12] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 3

[13] Uday Kamath, John Liu, and James Whitaker. *Deep learning for NLP and speech recognition*, volume 84. Springer, 2019. 1

[14] A Kimura, Z Ghahramani, K Takeuchi, T Iwata, and N Ueda. Few-shot learning of neural networks from scratch by pseudo example optimization. In *British Machine Vision Conference 2018, BMVC 2018*, 2019. 7

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[16] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14639–14647, 2020. 7

[17] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 6

[18] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS journal of photogrammetry and remote sensing*, 152:166–177, 2019. 1

[19] 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. 1, 3

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 3

[21] Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ruben Tolosana. Sensitivenets: Learning agnostic representations with application to face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):2158–2164, 2020. 3

[22] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019. 1

[23] Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 466–477, 2021. 1, 2, 3, 4

[24] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737, 2015. 1

[25] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1

[26] Magdalini Paschali, Sailesh Conjeti, Fernando Navarro, and Nassir Navab. Generalizability vs. robustness: investigating medical imaging networks using adversarial examples. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 493–501. Springer, 2018. 1

[27] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 7

[28] Suraj Srinivas and Francois Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning*, pages 4723–4731, 2018. 7

[29] 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. 1

[30] Fatemeh Vakhshiteh, Ahmad Nickabadi, and Raghavendra Ramachandra. Adversarial attacks against face recognition: A comprehensive study. *IEEE Access*, 9:92735–92756, 2021. 1

[31] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020. 2, 3, 5

[32] Zifan Wang, Yilin Yang, Ankit Shrivastava, Varun Rawal, and Zihao Ding. Towards frequency-based explanation for robust cnn. *arXiv preprint arXiv:2005.03141*, 2020. 2, 3, 5

[33] Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. Recent advances in deep learning for object detection. *Neurocomputing*, 396:39–64, 2020. 1

[34] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11492–11501. PMLR, 18–24 Jul 2021. 1, 3, 5

[35] Lu Yang, Qing Song, and Yingqi Wu. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia Tools and Applications*, 80(1):855–875, 2021. 1

[36] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019. 3

[37] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. 1

[38] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017. 1