SAT: Improving Adversarial Training via Curriculum-Based Loss Smoothing

Chawin Sitawarin*
University of California, Berkeley
Berkeley, California, USA
chawins@berkeley.edu

Supriyo Chakraborty
IBM T. J. Watson Research Center
Yorktown Heights, New York, USA
supriyo@us.ibm.com

David Wagner
University of California, Berkeley
Berkeley, California, USA
daw@cs.berkeley.edu

ABSTRACT

Adversarial training (AT) has become a popular choice for training robust networks. However, it tends to sacrifice clean accuracy heavily in favor of robustness and suffers from a large generalization error. To address these concerns, we propose Smooth Adversarial Training (SAT), guided by our analysis on the eigenspectrum of the loss Hessian. We find that curriculum learning, a scheme that emphasizes on starting "easy" and gradually ramping up on the "difficulty" of training, smooths the adversarial loss landscape for a suitably chosen difficulty metric. We present a general formulation for curriculum learning in the adversarial setting and propose two difficulty metrics based on the maximal Hessian eigenvalue (H-SAT) and the softmax probability (P-SAT). We demonstrate that SAT stabilizes network training even for a large perturbation norm and allows the network to operate at a better clean accuracy versus robustness trade-off curve compared to AT. This leads to a significant improvement in both clean accuracy and robustness compared to AT, TRADES, and other baselines. To highlight a few results, our best model improves normal and robust accuracy by 6% and 1% on CIFAR-100 compared to AT, respectively. On Imagenette, a ten-class subset of ImageNet, our model outperforms AT by 23% and 3% on normal and robust accuracy respectively.

CCS CONCEPTS

• **Security and privacy → Domain-specific security and privacy architectures**.

KEYWORDS

Adversarial Machine Learning, Adversarial Examples, Curriculum Learning

*Part of the work done while at IBM T. J. Watson Research Center.

1 INTRODUCTION

It is well-known that machine learning models are easily fooled by adversarial examples, generated by adding carefully crafted perturbation to normal input samples [4, 10, 28]. This raises serious safety concerns for systems and solutions that rely on machine learning as a crucial component (e.g., identity verification, malware detection, self-driving vehicles). Among numerous defenses proposed, Adversarial Training (AT) [23] is one of the most widely used algorithm to train neural networks that are robust to adversarial examples.

While the formulation of AT as a robust optimization problem is theoretically sound, solving it for neural networks is indeed tricky. Here, we focus on two problems from which models trained by AT tend to often suffer. First, it tends to sacrifice accuracy on benign samples, by a large margin, to gain robustness or accuracy on adversarial examples. This is undesirable for applications that requires high accuracy when operating in the normal settings. Second, models trained with AT often have a large generalization gap between their train and test adversarial accuracies. This gap is typically larger than the generalization error in non-adversarial settings. In some cases, neural networks are reduced to trivial classifiers that outputs a constant label for all inputs as they fail to learn any better robust decision boundary that actually exists.

Since the objective of AT is sound, we believe that these problems can be mitigated by optimization techniques. Intuitively, we posit that both the above problems stem from the model being presented with adversarial examples that are "too difficult to learn from" at the very beginning of the training which, in turn, causes the model to overfit to such samples. To this end, we observe that the concept of *curriculum learning* [3], which advocates that model training be initiated with "easy" samples before introducing the "hard" ones, can naturally help overcome the above problems.

Different from prior approaches used in curriculum-based adversarial training algorithms, we draw inspiration from the connection between small generalization error and smooth loss landscapes [13, 17] and design two difficulty metrics—*maximum eigenvalue of the Hessian matrix* and *softmax probability gap*—that directly and indirectly affect the sharpness of the adversarial loss landscape. By controlling these metrics, we can bias the network towards flatter regions on the adversarial loss landscape and hence, minimize the generalization gap. Following the metrics, we name these two algorithms, *Hessian-Based* Smooth Adversarial Training (H-SAT) and *Probability-Based* Smooth Adversarial Training (P-SAT). Following the technique from Li et al. [21], we visualize the loss landscapes (see Fig. 1) of the networks trained with AT, H-SAT, and P-SAT to confirm that both of our schemes do lead the networks to smoother regions.

(a) Adversarial Training (AT) **(b) H-SAT (ours)** **(c) P-SAT (ours)**
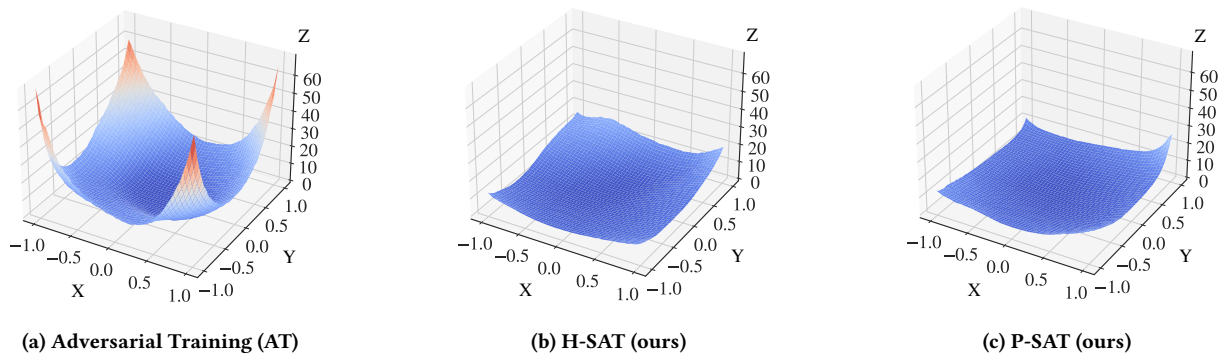
Figure 1: Loss landscapes of three PreAct-ResNet-20 models adversarially trained on CIFAR-10 using perturbation $\epsilon = 8/255$. The x-axis and y-axis represent two orthogonal random directions in the network parameter space. The z-axis represents the adversarial loss computed with the same normalization proposed by Li et al. [21]. Applying our algorithms, (b) H-SAT and (c) P-SAT, results in a much smoother loss landscape than (a) AT.

We make the following contributions. First, we unify the prior works on curriculum-based adversarial training under a single formulation. Second, we derive techniques to quickly approximate the two difficulty metrics used in H-SAT and P-SAT so that they can be combined with adversarial training efficiently. Finally, we systematically compare our proposed method against the baselines using multiple datasets (MNIST, CIFAR-10, CIFAR-100, Imagenette) and network architectures. Both our proposed schemes outperform state-of-the-art defenses including TRADES [37] and other curriculum-inspired algorithms [5, 7, 30] in term of robustness while maintaining competitive clean accuracy in most settings. Our best model improves normal and robust accuracy by 6% and 1% on CIFAR-100 compared to AT, respectively. On Imagenette with $\epsilon = 16/255$, our model outperforms AT by 23% and 3% on normal and robust accuracy respectively.

2 BACKGROUND AND RELATED WORK

Adversarial examples. Adversarial examples are a type of evasion attack against machine learning models generated by adding small perturbations to clean samples [4, 10, 28]. The desired perturbation, constrained to be within some ℓ_p-norm ball, is typically formulated as a solution to the following optimization problem:

$$x^{adv} = x + \delta^* \quad \text{where} \quad \delta^* = \underset{\delta:\|\delta\|_p \leq \epsilon}{\arg\max} \ \ell(x + \delta; \theta) \quad (1)$$

where $\ell(\cdot; \theta) : \mathbb{R}^d \to \mathbb{R}$ is the loss function of the target neural network, parameterized by θ, with respect to its input. The perturbation is bounded in an ℓ_p-norm ball of radius ϵ and treated as a proxy for imperceptibility of the perturbation. Projected gradient descent (PGD) is a popular technique used to solve Eqn. (1) [23].

Defenses against adversarial examples. In a nutshell, the adversarial training (AT) algorithm, iteratively generates adversarial examples corresponding to every batch of the training data by solving Eqn. (1) and then trains a model by minimizing the expected loss over the adversarial samples. AT is formulated as an optimization of the saddle point problem where θ are the model parameters, and

$\{(x_i, y_i)\}_{i=1}^n$ denotes the training set:

$$\underset{\theta}{\arg\min} \ \frac{1}{n} \sum_{i=1}^n \ell_\epsilon(x_i; \theta) \quad (2)$$

$$\text{where} \quad \ell_\epsilon(x; \theta) \coloneqq \underset{\delta:\|\delta\|_p \leq \epsilon}{\max} \ \ell(x + \delta; \theta) \quad (3)$$

Several prior works have attempted to improve AT in terms of computation time [26, 33] and robustness gain [11, 34]. Others have tried different loss functions that are better suited to adversarial training [9, 31, 37].

Curriculum learning and adversarial training. In curriculum adversarial training (CAT18) [5], the authors create a curriculum by slowly increasing the number of PGD steps during training. But their empirical results suggest that curriculum alone is not effective and it must be combined with other techniques such as quantization and batch mixing. Wang et al. [30] (DAT) defined convergence score, motivated by the Frank-Wolfe optimality gap, as a metric to imitate a curriculum. Two recent works, Balaji et al. [2] (IAAT) and Cheng et al. [7] (CAT20), use an adaptive and sample-specific perturbation norm during training. Their motivation is that not all samples should be at a fixed distance from the decision boundary as encouraged by AT. Instead, margins should be flexible and data-dependent. In fact, this effect is a by-product of our scheme which perturbs *naturally* easier samples more than harder ones [1]. Concurrent to our work, Zhang et al. [38] proposed a formulation of an upper bound of the adversarial loss and, similarly to ours, used early stopping as a realization. However, their criterion is based on the number of PGD steps, while ours, inspired by curriculum learning, relies on a concrete difficulty metric. Furthermore, we also perform extensive evaluation over a number of bench marking datasets to validate our approach.

Smoothness of Loss Landscapes. There is a long-standing hypothesis regarding the correlation between smoothness of the loss surface and the network generalization error [13, 17]. Previous works have

---

[1]Naturally easy samples can be thought of as clean inputs that a given network classifies correctly with high confidence. In other words, easy samples are on the correct side of the decision boundary and are also far from it.

successfully improved generalization in both normal and adversarial settings by biasing the training process of the networks towards regions with smooth loss landscapes [6, 15, 34]. *In this work, we achieve a similar goal but follow a unique approach via curriculum learning, which has been shown to produce a smoothening effect on the objective [3].*

3 ADVERSARIAL LOSS LANDSCAPE

3.1 Smoothness and Hessian

Recently, Liu et al. [22] demonstrated a theoretical correlation between the *sharpness* of the adversarial loss landscape and the perturbation norm ϵ. A smaller ϵ implies a smoother loss landscape. We first briefly restate this result. In this setup, Liu et al. [22] assume that the normal loss function, $\ell(x; \theta)$, has Lipschitz continuous gradients w.r.t. θ and x with constants $L_{\theta\theta}$ and $L_{\theta x}$ respectively. Mathematically, this can be written as $\forall x, x_1, x_2 \in \mathbb{R}^d$ and $\forall \theta, \theta_1, \theta_2 \in \Omega$,

$$\|\nabla_\theta \ell(x; \theta_1) - \nabla_\theta \ell(x; \theta_2)\|_2 \le L_{\theta\theta} \|\theta_1 - \theta_2\|_2 \quad (4)$$

$$\|\nabla_\theta \ell(x_1; \theta) - \nabla_\theta \ell(x_2; \theta)\|_2 \le L_{\theta x} \|x_1 - x_2\|_2 \quad (5)$$

For adversarial loss, $\ell_\epsilon(x; \theta)$, the analogous expression includes an extra term that depends on the perturbation norm ϵ as follows [22]:

$$\|\nabla_\theta \ell_\epsilon(x; \theta_1) - \nabla_\theta \ell_\epsilon(x; \theta_2)\|_2 \le L_{\theta\theta} \|\theta_1 - \theta_2\|_2 + 2\epsilon L_{\theta x} \quad (6)$$

While Eqn. (6) reveals a relationship between ϵ and the smoothness of the adversarial loss (LHS of Eqn. 6), the upper bound is loose and difficult to compute. This is because Eqn. (6) holds globally and both $L_{\theta\theta}$ and $L_{\theta x}$ are also global quantities (they hold for all values of θ and x). Instead, we are only interested in the local smoothness at a particular θ in practice.

Prior works often quantified smoothness using the eigenspectrum of the Hessian matrix computed locally around a given θ. To see why it makes sense to use the Hessian, note that we can upper bound the LHS in Eqn. (6) by the spectral norm of the Hessian matrix using the mean value theorem. In particular, let $\Delta\theta = \theta_2 - \theta_1$ and $\theta(t) = \theta_1 + t\Delta\theta$, we know that $\exists t \in [0, 1]$ s.t.

$$\|\nabla_\theta \ell_\epsilon(x; \theta_1) - \nabla_\theta \ell_\epsilon(x; \theta_2)\|_2 \le \left\|\nabla_\theta^2 \ell_\epsilon(x; \theta(t))\right\|_{(2)} \|\theta_1 - \theta_2\|_2 \quad (7)$$

We use $\|\cdot\|_{(2)}$ to denote the spectral norm of a matrix to differentiate it from ℓ_2-norm of a vector. Now obviously, if we only consider θ_2 that are close to θ_1, simply choosing $t = 0$ yields a good estimate of the local smoothness around θ_1.

Here, we also choose to measure the smoothness locally. We use Hessian of the adversarial loss w.r.t. θ, which is defined as Hessian of the normal loss evaluated at the adversarial example for a given θ.[2] Specifically,

$$\textsc{Smoothness}(\theta) \approx \|H_\epsilon(x; \theta)\|_{(2)} \quad (8)$$

$$H_\epsilon(x; \theta) := \nabla_\theta^2 \ell(x^*; \theta) \text{ for } x^* \in \underset{z: \|z-x\|_p \le \epsilon}{\arg\max} \ \ell(z; \theta) \quad (9)$$

---

[2]This should be taken as an approximation rather than the true Hessian for the following reason. Note that the adversarial loss $\ell_\epsilon(x; \theta)$ is a maximal-value function of the normal loss over x (see Eqn. (2)). When the normal loss is convex, Danskin's Theorem states that we can use the substitution with x^* to compute first-order derivatives of an extreme-value function. However, here we are considering second-order derivatives of a non-convex function. For a more rigorous analysis, we may need a more generalized version of Danskin's Theorem for Hessian [27].

Note, the above measure of smoothness, as suggested previously in Liu et al. [22], also depend on both ϵ and the Hessian of $\ell(\cdot)$ w.r.t. θ. However, their relationship is differently expressed as in Eqn. (9) compared to that in Eqn. (6). Furthermore, notice that $\|H_\epsilon(x, \theta)\|_{(2)}$ is the absolute value of the largest eigenvalue of $H_\epsilon(x, \theta)$ since any Hessian matrix is symmetric. We will refer to this quantity as "maximal Hessian eigenvalue."

3.2 Curriculum Learning and Smoothness

We believe that smooth loss landscapes will benefit the adversarial training process in two ways: generalization and convergence. First, as mentioned in Section 2, many previous works have shown connections between generalization and smoothness of the loss landscape. Flat minima introduce an *implicit bias* for SGD on deep neural networks and has served as an explanation to the surprisingly good generalization of over-parameterized neural networks. Examples of factors that affect this implicit bias include batch size, learning rate, and architecture such as residual connections [16, 17, 21, 24]. Other works also aim to "artificially" create such flat local minima through more advanced training techniques [6, 15].

The second benefit of a smooth loss surface is a faster convergence. It is well-known that smoother loss (e.g., smaller Lipschitz gradient constant) allows for a large step size and hence, a faster convergence on both convex and non-convex problems including ERM on neural networks [18, 20]. Conversely, a sharp loss surface is harmful to the training process. Liu et al. [22] show that a large adversarial perturbation could increase the sharpness, i.e., make gradients large even near the minima, and hence, slow down the training.

In this work, we explore curriculum learning as a mechanism to encourage smoothness of the adversarial loss surface. The original intuition behind curriculum learning as outlined by Bengio et al. [3] was to help smooth the loss landscape in the initial phase of non-convex optimization. While this intuition has found acceptance in the community, to the best of our knowledge, this notion of smoothness has not been used for curriculum learning. Additionally, curriculum learning is particularly suitable in the adversarial context because it can, directly or indirectly, manipulate ϵ, which is a main factor affecting the smoothness as mentioned previously (Section 3.1). In Section 4, we make this connection explicit via our proposed algorithms.

4 SMOOTHED ADVERSARIAL TRAINING

4.1 Curriculum Learning and Difficulty Metric

It has been both empirically and theoretically (for a linear regression model) established that curriculum learning improves the early convergence rate as well as the final generalization performance, especially when the task is difficult (e.g., under-parameterized models, heavy regularization) [3, 12, 32]. The main challenge therefore lies in defining an effective difficulty metric to dictate the order of the training samples presented to the network. Luckily, the adversarial setting provides several intuitive difficulty metrics that can be easily controlled through the perturbation norm, ϵ.

We start by proposing a formulation of the curriculum-augmented adversarial loss, or *curriculum loss*, denoted by $\ell_{\psi, \epsilon}$ where $\psi : \mathbb{R}^d \to \mathbb{R}$ is a given difficulty metric acting as an additional constraint on

the adversarial loss. We call it the *curriculum constraint*.

$$\ell_{\psi,\epsilon}(x,\lambda) \quad = \quad \max_{\delta:\|\delta\|_\infty \le \epsilon} \quad \ell(x+\delta) \tag{10}$$
$$\text{s.t.} \quad \psi(x+\delta) \le \lambda$$

This general formulation above unifies prior works on curriculum-inspired adversarial training (see Table 4 in Appendix B). Without the curriculum constraint, the curriculum loss reduces to the normal adversarial loss in Eqn. (1). The difficulty parameter λ should be scheduled to increase as the training progresses such that it reaches its maximal value well before the end of training. This is to ensure that the curriculum loss converges to the original adversarial loss.

As mentioned in Section 3.2, we propose two difficulty metrics that directly aim to smoothen the loss landscape: *Maximal Eigenvalue of the Hessian* and *Softmax Probability Gap*.

4.2 Maximal Eigenvalue of the Hessian

To encourage smoothness, we can directly control the largest eigenvalue of the Hessian throughout the training. In Section 3.1, we described the correlation between the largest eigenvalue of the Hessian and adversarial strength and explained its usage as a curriculum constraint. We now use it as a difficulty metric,

$$\psi_H(x) := \|H_\epsilon(x;\theta)\|_{(2)} \tag{11}$$

for our *Hessian-Based Smooth Adversarial Training* (H-SAT).

This direct control on the Hessian, however, has two limitations. First, computing $\psi_H(x)$ requires calculating the maximal eigenvalue of the Hessian which is very expensive to execute at every PGD step or even training step. To mitigate this problem, we devise several approximations which significantly speed up the calculation (see Section 5). Second, the maximal Hessian eigenvalue is small not only when the input is easy but also when it is hard, i.e., when the probability of the correct class is close to either 1 (for easy) or 0 (for difficult). This might lead to an undesirable side effect where keeping the maximum Hessian eigenvalue small does not guarantee that only easy samples are presented in the early phases of training. We illustrates this effect for logistic regression in Fig. 2a. Note, the second derivative value is small both when the logit value is large and when it is small (a large negative number), corresponding to the green and orange arrows respectively.

(a) Second derivatives   **(b) True-class probability**
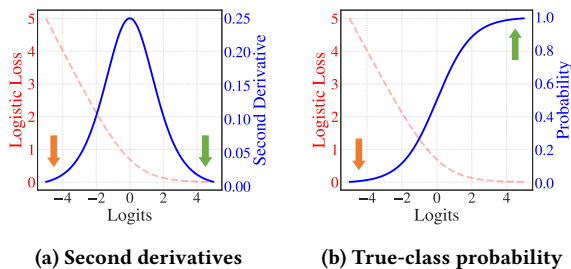
Figure 2: Relationship between logistic loss, (a) its second order derivative w.r.t. logit, and (b) the true-class (positive) probability. Small second derivative can correspond to difficult inputs (i.e., high loss, small logit, small probability, orange arrow) which is undesirable.

As shown in Fig. 2b, one way to circumvent this limitation of H-SAT is to directly control the class probability of a sample instead. If the class probabilities are large, it ensures that the samples are easy, and it is likely that the second derivative (and hence, the maximal Hessian eigenvalue) is also small. Since class probability for a sample is available after every PGD step, we will also solve the first problem simultaneously. We will explore this option for the difficulty metric in the next section below.

4.3 Softmax Probability Gap

To overcome some of the limitations of H-SAT, we propose *softmax probability gap*—difference between the true class probability and the largest softmax probability excluding the true class—as the second difficulty metric for our scheme. Formally, it is defined as

$$\psi_P(x) := \max_{j \ne y} f(x)_j - f(x)_y \tag{12}$$

where $y \in \{1, ..., c\}$ is the ground-truth label of x, and $f : \mathbb{R}^d \to \mathbb{R}^c$ is the softmax output of a neural network.

The probability gap has an intuitive interpretation and is bounded between -1 and 1. A perturbed input with a large gap, i.e., $\psi_P(x) \approx 1$, means that it has been incorrectly classified with high confidence by the network, suggesting that it is a "hard" sample. On the other hand, if $\psi_P(x) \approx -1$, the input must be "easy" since the network has classified it correctly with high confidence. When the gap is zero, the input is right on the decision boundary of the classifier.

When used as a curriculum constraint $\psi_P(x) \le \lambda$, softmax probability gap is also directly related to traditional adversarial training. When $\lambda = 1$, the constraint is always satisfied for any x and δ. Thus, the curriculum loss reduces to the normal adversarial loss. The curriculum loss is also a lower bound of the adversarial loss for any λ with equality for $\lambda \ge 1$.

$$\forall \lambda, \ \ell_{\psi,\epsilon}(x,\lambda) \le \ell_\epsilon(x) \quad \text{and} \tag{13}$$
$$\forall \lambda \ge 1, \ \ell_{\psi,\epsilon}(x,\lambda) = \ell_\epsilon(x) \tag{14}$$

With the above interpretation in mind, we create a curriculum for adversarial training by progressively increasing λ from 0 to 1 during the training (illustrated in Fig. 3). First, we expose the model to a weak adversary, or equivalently an easy objective, in the early stage of the training (small λ). Progressively, the curriculum objective becomes more difficulty (large λ) and eventually reach the original adversarial loss when $\lambda = 1$. This algorithm is named *Probability-Based Smooth Adversarial Training* (P-SAT).

4.4 Early Termination of PGD

Both the curriculum constraints are non-convex and thus it is difficult to solve Eqn. (10) directly as a constrained optimization problem. Conceptually, we can satisfy the constraint by terminating PGD as soon as the constraint is violated. This is a heuristic for most choices of the difficulty metric ψ, but in the case of P-SAT, the early termination solves the optimization exactly for the binary class case as stated in Proposition 1 below. For H-SAT, the early termination is slightly more complicated and is described in Appendix B.3.

Proposition 1. *In a binary-class problem, an optimum of the curriculum loss with $\psi_P(\cdot)$ as the curriculum constraint can be found by a projected gradient method that terminates when the curriculum constraint is violated.*
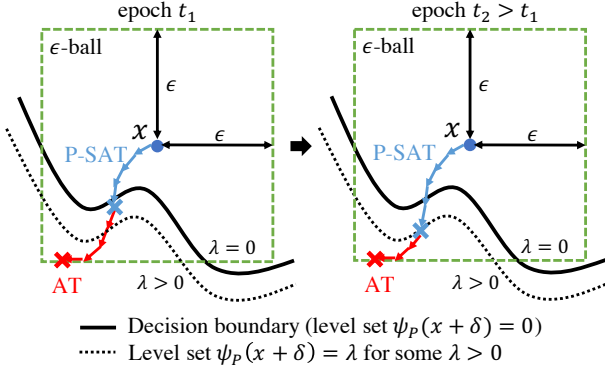
Figure 3: Comparison between the perturbed samples generated by AT and P-SAT. The blue and red arrows represents PGD updates. For P-SAT, the process stops when the curriculum constraint is violated (blue cross). As λ increases, P-SAT approaches AT.

Algorithm 1: P-SAT

Input : Training set (X, Y), neural network weights θ
Parameters : PGD steps K, step size η, difficulty scores $\{\lambda^t\}_{t=1}^T$, projection Π as a function of p and ϵ

1 **for** $t = 1,\dots,T$ **do**
2   Sample $\{(x_i, y_i)\}_{i=1}^B$ from (X, Y).
3   Initialize all δ_i's at random.
4   # Generating adversarial examples with curriculum
5   **for** $k = 1,\dots,K$ **do**
6   # Mask out updates that violate the constraint
7      $mask_i = \mathbb{1}\left\{\psi(x_i) \leq \lambda^t\right\}$
8   # Projection step
9      $\delta_i \leftarrow \Pi(x_i + \eta\nabla_x\ell(x_i)) - x_i$
10   # Update samples with the given mask
11      $x_i \leftarrow x_i + mask_i * \delta_i$
12   **end**
13   # Network training step
14   Compute $\ell(x_i)$ and update weights θ^t.
15 **end**

Below we will explain the intuition behind this proposition and defer the complete derivation to Appendix B.2 for improved readability. Note, the loss function (negative log-likelihood) is a monotonic function of the output class probability. This allows us to simplify the curriculum loss in Eqn. (10) as follows:

$$\ell_{\psi,\epsilon}(x, \lambda) = \max_{\delta:\|\delta\|_p \leq \epsilon} \min\{\ell(x + \delta), -\log\left((1-\lambda)/2\right)\} \quad (15)$$

The second term of the piecewise minimization in Eqn. (15) is just a constant. This modified problem can be solved with PGD, similarly to the adversarial loss, and terminated as soon as the first term is larger than the second—which is equivalent to the curriculum constraint being violated. The curriculum loss is slightly more complicated for the multi-class case, but we propose to solve it

approximately using the same early termination as a heuristic. Algorithm 1 summarizes the implementation of P-SAT.

5 COMPUTING MAXIMAL HESSIAN EIGENVALUES

In this section, we detail techniques to compute the maximal Hessian eigenvalue for $\psi_H(\cdot)$ efficiently. Computing the full Hessian matrix for neural networks with millions of parameters is computationally expensive and, in fact, unnecessary. We apply the power method [19] to compute the top eigenvalues of the Hessian directly. The method generally converges in a few iterations each of which requires two backward passes.

However, the power method is still too expensive to run for every PGD step and training iteration of the already computation-intensive adversarial training. Therefore, to significantly minimize the computational overhead, we approximate the maximal Hessian eigenvalue using the second-order Taylor's expansion,

$$\ell(\theta + \zeta) = \ell(\theta) + \zeta^\top \nabla_\theta \ell(\theta) + \frac{1}{2}\zeta^\top \nabla_\theta^2 \ell(\theta)\zeta + O(\|\zeta\|_2^3) \quad (16)$$

combined with the fact that

$$\frac{1}{2}\left\|\nabla_\theta^2 \ell(\theta)\right\|_{(2)} = \max_{\|\zeta\|_2=1} \frac{1}{2}|\zeta^\top \nabla_\theta^2 \ell(\theta)\zeta|. \quad (17)$$

As we are only interested in the largest positive eigenvalue, we can ignore the absolute value sign.[3] Letting $g := \nabla_\theta \ell(\theta)/\|\nabla_\theta \ell(\theta)\|_2$, we can now approximate the upper bound $(\overline{\sigma_1})$ and the lower bound $(\underline{\sigma_1})$ of the maximal eigenvalue of the Hessian (σ_1) by appropriately setting $\zeta = \pm \alpha g$ for some small constant α:

$$\underline{\sigma_1} \lessgtr \left\|\nabla_\theta^2 \ell(\theta)\right\|_{(2)} \lessgtr \overline{\sigma_1} \quad (18)$$

$$\underline{\sigma_1} = \frac{1}{\alpha}\max\{\ell(\theta + \alpha g) - \alpha\|\nabla_\theta \ell(\theta)\|_2, \quad (19)$$
$$\ell(\theta - \alpha g) + \alpha\|\nabla_\theta \ell(\theta)\|_2\} - \ell(\theta)$$

$$\overline{\sigma_1} = \frac{1}{\alpha}\left\{\ell(\theta + \alpha g) + \alpha\|\nabla_\theta \ell(\theta)\|_2\right\} - \ell(\theta) \quad (20)$$

Again, for improved readability, the full derivation and the implementation considerations are provided in Appendix C.1.

Evaluating Eqn. (20) and (19) at the adversarial example of x, gives us the upper and a lower bound respectively, for $\|H_\epsilon(x, \theta)\|_{(2)}$. It is also important to note that setting α sufficiently small makes our approximation more accurate in practice. In fact, we empirically validated that the lower bound is indeed very tight (within 15% from the true value).

6 EXPERIMENTS

6.1 Setup

We train and test the robustness of our proposed schemes as well as the baselines on four image datasets, namely MNIST, CIFAR-10, CIFAR-100, and Imagenette [14], a more realistic and higher-dimensional dataset containing a 10-class subset of the full ImageNet samples. We use a small CNN for MNIST, Pre-activation ResNet-20 (PRN-20) and WideResNet-34-10 (WRN-34-10) for CIFAR-10/100, and ResNet-34 for Imagenette. For evaluation, we use Auto-Attack [8], a novel attack based on an ensemble of four different

---

[3]Assuming that we are near local minima, most (if not all) eigenvalues should be positive with only a few very small negative eigenvalues.

Table 1: Clean and adversarial accuracy of the defenses on two models (PreAct-ResNet-20 and WideResNet-34-10) for the CIFAR-10 and CIFAR-100 datasets. "Sum" indicates the sum of the clean and the adversarial accuracy. The largest number in each column is shown in bold.

Defenses	CIFAR-10 (PRN-20)			CIFAR-10 (WRN-34-10)			CIFAR-100 (PRN-20)			CIFAR-100 (WRN-34-10)		
	Clean	Adv	Sum	Clean	Adv	Sum	Clean	Adv	Sum	Clean	Adv	Sum
AT [23]	80.67	45.19	125.86	86.18	49.72	135.90	51.76	21.93	73.69	60.77	24.54	85.31
TRADES [36]	80.50	**45.77**	126.27	88.08	45.83	133.91	54.73	20.17	74.90	58.27	23.57	81.84
DAT [30]	81.83	42.97	124.80	86.72	45.38	132.10	54.65	20.85	75.50	54.71	20.35	75.06
CAT20 [7]	**86.46**	21.69	108.15	**89.61**	34.78	124.39	49.29	13.13	62.42	62.84	16.82	79.66
H-SAT (ours)	81.85	44.88	126.73	85.56	47.25	132.81	56.64	22.25	78.89	61.33	**25.43**	86.76
P-SAT (ours)	83.99	44.54	**128.53**	86.84	**50.75**	**137.59**	57.90	22.93	80.83	62.95	24.56	**87.51**

Table 2: Clean and adversarial accuracy of the defenses on MNIST dataset. Adversarial accuracy is measured by AutoAttack for $\epsilon = 0.3$ and $\epsilon = 0.45$. The numbers in red indicate that the network is stuck in a sub-optimal local minimum.

Defenses	$\epsilon = 0.3$			$\epsilon = 0.45$		
	Clean	Adv	Sum	Clean	Adv	Sum
AT [23]	98.07	85.47	183.54	11.22	11.22	22.44
TRADES [36]	98.98	90.70	189.68	97.36	0.00	97.36
DAT [30]	98.93	**92.24**	**191.17**	97.98	65.71	163.69
CAT20 [7]	**99.46**	0.00	99.46	**99.39**	0.00	99.39
H-SAT (ours)	99.01	80.71	179.72	98.35	54.10	152.45
P-SAT (ours)	99.16	92.00	191.16	97.87	58.50	156.37

attacks that are collectively stronger than PGD and capable of avoiding gradient obfuscation issue [1]. We compare our H-SAT and P-SAT to four baselines: AT, DAT, CAT20, and TRADES.[4] For more details on the setup, see Appendix A.

6.2 Results

First, we compare the defenses in terms of their robustness, accuracy, and a combination of the two. We report the sum of the clean and the adversarial accuracy as a single metric to represent the robustness-accuracy trade-off. We do not argue that it is the only correct (or the best) metric but rather a simple one among many other options (e.g., a weighted sum or a non-linear function). We highlight higher performance gains of our schemes for more difficult tasks and show that they guide the networks towards smoother loss landscapes and improved local minima compared to AT.

6.2.1 Comparing clean and adversarial accuracy. On MNIST (shown in Table 2), with ϵ of 0.3, both the clean and the adversarial accuracy, fall in the same range, for all defenses except for CAT20. Our P-SAT has very similar accuracies to DAT and outperforms the rest of the baselines. CAT20 experiences gradient obfuscation [1], resulting in an over-estimated adversarial accuracy against PGD attack. However, the true robustness is very low as revealed by

the stronger AutoAttack. This issue persists on CAT20 for all the datasets we tested and is likely caused by their label smoothing.

For $\epsilon = 0.45$, the difference becomes significant. Apart from H-SAT, P-SAT, and DAT, none of the other defenses are robust, having adversarial accuracy of ~10% or lower. The clean and the adversarial accuracy of AT are the same and are close to a random guess because it outputs the same logit values for every input. This phenomenon happens when the training gets stuck in a sub-optimal local minimum where the network "finds an easy way out" and resorts to learning a trivial solution. Changing the optimizer, the learning rate, or the random seed can sometimes mitigate the problem, but we found that no combination of these changes allowed AT to effectively train on MNIST with $\epsilon \geq 0.35$. It is likely caused by the non-smooth loss landscape which is amplified by a large value of ϵ. Importantly, the fact that only curriculum-based adversarial training methods (H-SAT, P-SAT, and DAT) works in this case suggests that they help smoothen the loss surface.

In Table 1, our scheme is more robust than all other defenses on both PRN-20 and WRN-34-10 for CIFAR-10 and CIFAR-100 datasets with an exception of PRN-20 on CIFAR-10 where TRADES is the most robust. P-SAT also has the highest sum of clean and adversarial accuracy in all cases. CAT20 generally has slightly higher clean accuracy than the other models, but it is the least robust by a large margin due to the false robustness issue previously mentioned. We note that on most settings, AT is a very strong baseline. It has roughly the same performance as TRADES and outperforms the other curriculum-based schemes in many experiments. This may seem surprising, but it is also observed by Rice et al. [25] and Gowal et al. [11] that when the training is stopped early to prevent overfitting (as in our experiments), AT performs as well as other more complicated defenses.

As expected, P-SAT outperforms H-SAT in most settings. This is likely due to the two limitations we mentioned in Section 4.2: (1) additional approximations introduced to H-SAT for practical consideration as well as (2) the fact that the Hessian eigenvalue does not always correspond to difficulty. While P-SAT controls the smoothness in a less direct manner, it does not suffer from these two issues. Consequently, it is computationally cheaper and also performs slightly better.

6.2.2 Larger Improvement with More Difficult Tasks. On CIFAR-10 with $\epsilon = 16/255$ and $24/255$ (Table 3), our schemes have higher clean accuracy compared to AT and higher adversarial accuracy than

---

[4]We do not include the results on Cai et al. [5] because ϵ-scheduling has been shown to perform worse than DAT. The scheme uses a non-differentiable component which makes evaluation more complicated and not comparable to the other baselines. We also found that it suffers from gradient obfuscation [1].

Table 3: Clean and adversarial accuracy of the defenses on CIFAR-10 and Imagenette datasets trained on PreAct-ResNet-20 and ResNet-34 respectively. Here, we use a larger perturbation norm ϵ of $16/255$ and $24/255$.

| Defenses | CIFAR-10 | | | | | | Imagenette | | | | | |
| | $\epsilon = 16/255$ | | | $\epsilon = 24/255$ | | | $\epsilon = 16/255$ | | | $\epsilon = 24/255$ | | |
	Clean	Adv	Sum	Clean	Adv	Sum	Clean	Adv	Sum	Clean	Adv	Sum
AT [23]	63.66	**23.78**	87.44	41.68	**15.21**	56.89	49.10	28.00	77.10	42.55	21.05	63.60
TRADES [36]	**68.70**	17.20	85.90	**58.83**	5.74	64.57	**78.05**	8.90	86.95	**68.50**	1.90	70.40
DAT [30]	57.28	18.94	76.22	31.21	14.36	45.57	66.20	30.30	96.50	52.50	24.50	77.00
H-SAT (ours)	64.27	23.19	87.46	49.85	13.35	63.20	69.10	**35.45**	**104.55**	47.50	**27.75**	75.25
P-SAT (ours)	66.99	22.42	**89.41**	52.58	12.24	**64.82**	72.20	31.25	103.45	62.15	20.00	**82.15**

(a) Maximal Hessian eigenvalue (b) Trace of Hessian (c) Gradient Norm
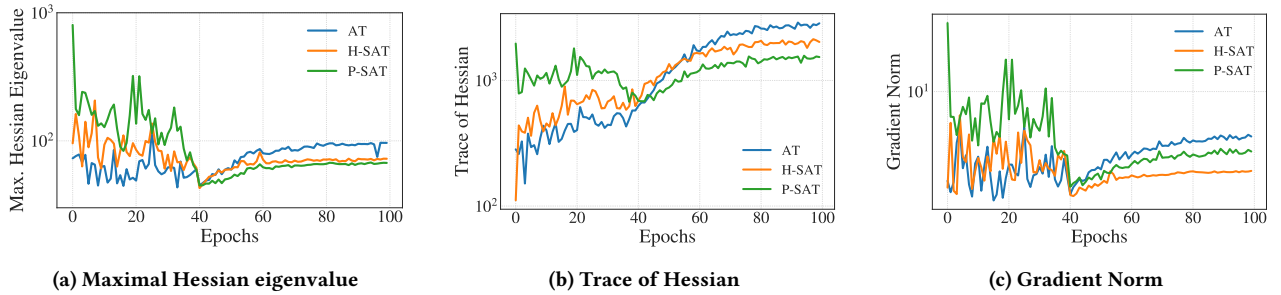
Figure 4: (CIFAR-10) Plots of three quantities typically used to measure the smoothness of the loss landscape of neural networks. Each is calculated exactly at each epoch during the training for the same subset of randomly chosen training samples.

TRADES. The difference has increased compared to the experiments with $\epsilon = 8/255$. The sum of the two accuracies of P-SAT also continues to beat that of the other defenses. In the same table, the benefit of our scheme is even more prominent on the more difficult and realistic classification task on Imagenette dataset. On this dataset, H-SAT and P-SAT outperform the baselines (we omit CAT20 since it is beaten by the other baselines) by a large margin, even larger than the results on CIFAR-10/100.

The improvement grows larger with an increase in the value of ϵ. This is because when ϵ is small, the early termination of PGD is not necessary, and the curriculum does not play a big role. Hence, P-SAT runs are very similar to those of AT. Alternatively, this also agrees with the observation made by Bengio et al. [3] and Weinshall et al. [32] that curriculum learning contributes more when the task is more difficult. A larger perturbation norm means a stronger adversary and a higher adversarial loss, representing a harder task.

6.2.3 Smoother Loss Landscape. We empirically confirm that our schemes, both H-SAT and P-SAT, increase smoothness of the adversarial loss landscape. To measure smoothness, we plot in Fig. 4 the maximal Hessian eigenvalue, trace of the Hessian, and norm of the gradients on the adversarial examples generated by PGD during training. There is a significant change in the trend of the three quantities at epoch 40 which is where the learning rate is reduced from 0.05 to 0.005. SGD almost converges to a local minima at this point, i.e., the loss no longer changes significantly. Prior to epoch 40, all three quantities behave erratically and vary abruptly.

However, after epoch 40, H-SAT and P-SAT consistently reach smoother local minima. Specifically, P-SAT has the smallest eigenvalue and trace of the Hessian. H-SAT is the second smallest while

AT comes in last. For the gradient norm, only H-SAT and P-SAT switch place, but the general trend remains the same. This plot is consistent with the visualization in Fig. 1 and helps explain why H-SAT and especially P-SAT perform better than the baselines.

We also measure the generalization gap by computing the difference between training and testing adversarial accuracies over the last 20 epochs of training. On CIFAR-10 with $\epsilon = 8/255$, the gaps are 2.42 ± 0.18, 2.08 ± 0.21, and 2.33 ± 0.24 for AT, H-SAT, and P-SAT, respectively, where the error denotes a 95%-confidence interval. The ordering of the generalization gaps is again consistent with that of spectral norm and trace of the Hessian in Fig. 4(a), (b).

7 CONCLUSION

We proposed a curriculum-based formulation of adversarial training. Keeping the optimization objective unmodified, we focused our analysis on difficulty measures that could guide the model training to smooth regions of the loss landscape, improving generalization. Towards this end, we proposed H-SAT and P-SAT, robust training algorithms that achieved high clean accuracy and a small generalization gap, addressing the two key problems of AT. Using extensive evaluation, we showed that H-SAT and P-SAT outperforms AT, TRADES, as well as other curriculum-inspired defenses on both clean and adversarial accuracies in most of the settings.

REFERENCES

[1] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 274–283.

[2] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. 2019. Instance Adaptive Adversarial Training: Improved Accuracy Tradeoffs in Neural Nets. *arXiv:1910.08051 [cs, stat]* (Oct. 2019). arXiv:1910.08051 [cs, stat]

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. ACM Press, Montreal, Quebec, Canada, 1–8. https://doi.org/10.1145/1553374.1553380

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 387–402.

[5] Qi-Zhi Cai, Chang Liu, and Dawn Song. 2018. Curriculum Adversarial Training. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 3740–3747. https://doi.org/10.24963/ijcai.2018/520

[6] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. 2017. Entropy-SGD: Biasing Gradient Descent into Wide Valleys. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017), 1–19.

[7] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. CAT: Customized Adversarial Training for Improved Robustness. *arXiv:2002.06789 [cs, stat]* (Feb. 2020). arXiv:2002.06789 [cs, stat]

[8] Francesco Croce and Matthias Hein. 2020. Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 2206–2216.

[9] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. 2020. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In *International Conference on Learning Representations*.

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

[11] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. 2021. Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples. *arXiv:2010.03593 [cs, stat]* (March 2021). arXiv:2010.03593 [cs, stat]

[12] Guy Hacohen and Daphna Weinshall. 2019. On the Power of Curriculum Learning in Training Deep Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2535–2544.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Flat Minima. *Neural Computation* 9, 1 (1997), 1–42.

[14] Jeremy Howard. 2021. Fastai/Imagenette. fast.ai.

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018 (34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018)*, Ricardo Silva, Amir Globerson, and Amir Globerson (Eds.). Association For Uncertainty in Artificial Intelligence (AUAI), 876–885.

[16] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. 2018. Three Factors Influencing Minima in SGD. *arXiv:1711.04623 [cs, stat]* (Sept. 2018). arXiv:1711.04623 [cs, stat]

[17] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv:1609.04836 [cs, math]* (Feb. 2017). arXiv:1609.04836 [cs, math]

[18] Koulik Khamaru and Martin J. Wainwright. 2019. Convergence Guarantees for a Class of Non-Convex and Non-Smooth Optimization Problems. *Journal of Machine Learning Research* 20, 154 (2019), 1–52.

[19] Yann LeCun, Patrice Y. Simard, and Barak Pearlmutter. 1992. Automatic Learning Rate Maximization by On-Line Estimation of the Hessian's Eigenvectors. In *Proceedings of the 5th International Conference on Neural Information Processing Systems (NIPS'92)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 156–163.

[20] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. 2016. Gradient Descent Only Converges to Minimizers. In *29th Annual Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 49)*, Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir (Eds.). PMLR, Columbia University, New York, New York, USA, 1246–1257.

[21] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. *Visualizing the Loss Landscape of Neural Nets*. Technical Report. 6389–6399 pages.

[22] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. 2020. On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them. In *Advances in Neural Information Processing Systems*.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[24] Rotem Mulayoff and Tomer Michaeli. 2020. Unique Properties of Flat Minima in Deep Networks. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 7108–7118.

[25] Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in Adversarially Robust Deep Learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 8093–8104.

[26] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial Training for Free!. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

[27] Alexander Shapiro. 1985. Second-Order Derivatives of Extremal-Value Functions and Optimality Conditions for Semi-Infinite Programs. *Mathematics of Operations Research* 10, 2 (1985), 207–219.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing Properties of Neural Networks. In *International Conference on Learning Representations*.

[29] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual Attention Network for Image Classification. *arXiv:1704.06904 [cs]* (April 2017). arXiv:1704.06904 [cs]

[30] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. 2019. On the Convergence and Robustness of Adversarial Training. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 6586–6595.

[31] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *International Conference on Learning Representations*.

[32] Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. *Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks*. Technical Report.

[33] Eric Wong, Leslie Rice, and J. Zico Kolter. 2020. Fast Is Better than Free: Revisiting Adversarial Training. In *International Conference on Learning Representations*.

[34] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. 2020. Adversarial Weight Perturbation Helps Robust Generalization. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 2958–2969.

[35] Sergey Zagoruyko and Nikos Komodakis. 2017. Wide Residual Networks. *arXiv:1605.07146 [cs]* (June 2017). arXiv:1605.07146 [cs]

[36] Yusen Zhan, Haitham Bou Ammar, and Matthew E. Taylor. 2016. Theoretically-Grounded Policy Advice from Multiple Teachers in Reinforcement Learning Settings with Applications to Negative Transfer. *IJCAI International Joint Conference on Artificial Intelligence* 2016-Janua, 7540 (2016), 2315–2321. https://doi.org/10.1038/nature14236

[37] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*.

[38] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 11278–11287.

We provide below detailed explanations, comparative empirical results, and theoretical analysis of our proposed approach. We begin with description of the experiments (datasets, model architecture, training parameters, and algorithm-specific parameters) in Section A. In Section B, we detail our curriculum loss framework which unifies the prior works, and then we justify our early termination techniques for P-SAT (Proposition 1) and H-SAT. Next, in Section C, we describe our approximation of the Hessian eigenvalue and additional experiments that compare smoothness of the loss landscape of multiple training schemes. Lastly, we finish with examples of the adversarial images from the Imagenette dataset in Section D.

A DETAILED DESCRIPTION OF THE EXPERIMENTS

The neural networks we experiment with all use ReLU as the activation function and are trained using SGD with a momentum of 0.9 and batch size of 128. We use early stopping during training, i.e., models are evaluated at the end of each epoch and only save the one with the highest adversarial validation accuracy thus far. Dataset-specific details of the training as well as brief descriptions of the model architectures are provided below.

MNIST. All experiments use a three-layer convolution network (8x8-filter with 64 channels, 6x6-filter with 128 channels, and 5x5-filter with 128 channels respectively) with one fully-connected layer. Models are trained for 70 epochs with a batch size of 128. The initial learning rate is set at 0.01 and is decreased by a factor of 10 at epochs $40, 50$, and 60. Weight decay is 5×10^{-4}. During training, we run PGD for 40 steps with a step size of 0.02 and use a uniform random initialization within the ℓ_∞-ball of radius ϵ.

CIFAR-10/CIFAR-100. We use both pre-activation ResNet-20 [29] and WideResNet-34-10 [35], which are trained for 100 epochs with a batch size of 128. We use 10-step PGD with step size of $2/255$ and random restart. The initial learning rate is set at 0.05 for ResNet and 0.1 for WideResNet and is decreased by a factor of 10 at epochs $40, 60$, and 80. Weight decay is set to 5×10^{-4} for CIFAR-10 and 2×10^{-4} for CIFAR-100. Standard data augmentation (random crop, flip, scaling, and brightness jitter) is also used.

Imagenette. The hyperparameters are almost identical to those for CIFAR-10. We train ResNet-34 for 100 epochs with a batch size of 128. The initial learning rate is 0.1 and is decreased by a factor of 10 at epochs $40, 60$, and 80. Weight decay is set to 5×10^{-4}.

For TRADES, we set $\beta = 6$ for all of the experiments on CIFAR-10, CIFAR-100, and Imagenette, which is the value suggested in the original paper. For MNIST, we did a grid search on the values of β including $\beta = 6$. However, we did not find any value of β that resulted in a robust model at $\epsilon = 0.45$ on MNIST, so we only report results for $\beta = 6$ in the table.

For DAT, we followed the schedule used in the original paper, i.e., the convergence score is initialized at 0.5 and is reduced to zero in 80 epochs for CIFAR-10 and CIFAR-100. For the other datasets, we found that the adversarial accuracy improved and is more comparable to the other defenses when the convergence score reached zero earlier. So we reduced the number of epochs over which the

(a) Step Schedules
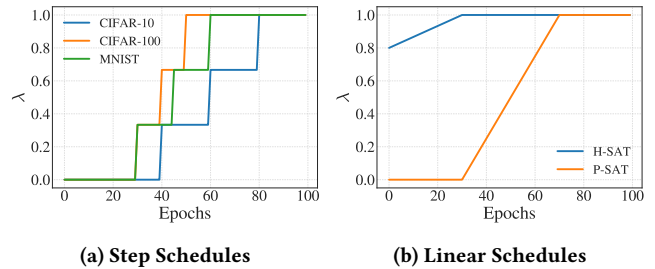
(b) Linear Schedules

Figure 5: Plots of multiple schedules of the difficulty parameter λ used by H-SAT and P-SAT. (a) displays three step schedules for MNIST, CIFAR-10, and CIFAR-100. (b) shows the linear schedules used in all datasets.

decay occurred from 80 to 30 for MNIST, 50 for CIFAR-10 with $\epsilon = 16/255, 24/255$, and 40 for Imagenette.

For CAT20, we used the original code which is provided by the authors, and we only experimented with the recommended default hyperparameters.

For H-SAT and P-SAT, we used slightly different schedules for the difficulty parameter λ depending on the setting (Fig. 5). For MNIST, we only used a step schedule wherein we increased λ is steps of 0.3333 at epochs 30, 45, and 60. On PRN-20, we used a similar step schedule, i.e., λ is increased at epochs 40, 60, and 80. On WRN-34-10, we found that it is beneficial to increase λ in earlier epochs. So we increased λ at epochs 30, 40, and 50 instead. In addition to the step schedules, we also use two linear schedules, one for H-SAT and the other for P-SAT. For H-SAT, λ starts off at 0.8 and increase to 1 by epoch 30. For P-SAT, λ increases from 0 to 1 between epoch 30 and 70.

All of the codes are written in PyTorch and run on servers with multiple Nvidia 1080ti and 1080 GPUs. On a single Nvidia 1080ti GPU with 12 cores of Intel i7-6850K CPU (3.60GHz) and 64 GB of memory, P-SAT with an MNIST model takes 2 hours to train, PRN-20 takes about 6 hours, and WRN-34-10 uses about 24 hours. P-SAT is slightly faster than AT due to the early termination. On the contrary, H-SAT uses approximately 50% additional computation time because of the Hessian eigenvalue computation.

B CURRICULUM CONSTRAINTS

B.1 The Curriculum Loss Framework

Our formulation of the curriculum loss and in particular, the curriculum constraint generalize the curriculum learning approach used by the prior works. Table 4 lists the different curriculum constraints used by each approach together with their method of choice to satisfy the corresponding constraints. CAT18 controls the difficulty by setting the number of PGD steps for generating adversarial examples. DAT terminates PGD early once the convergence score is lower than the specified value.

The maximum perturbation norm is also an intuitive difficulty metric that can be scheduled manually or can be automatically adjusted based on some condition. Both IAAT and CAT20 explicitly use sample-specific perturbation norm, $\epsilon^*(x)$, to control the difficulty. In addition, both methods approximate and schedule $\epsilon^*(x)$

Table 4: Comparison of constraints and implementations of different curriculum-based AT schemes unified under the form of Eqn. (10). **Note that CAT18 uses the number of PGD steps as the constraint which cannot be written analytically.**

Schemes	Curriculum Constraints	Implemented Methods
Perturbation norm	$\|\delta\|_\infty \le \lambda$	Projection: $\|\delta\|_\infty \le \lambda$
CAT18 [5]	n/a	Setting PGD steps
DAT [30]	$\langle \delta, \nabla\ell(x+\delta)\rangle - \epsilon \,\|\nabla\ell(x+\delta)\|_1 \le \lambda$	Early termination
IAAT [2]	$\psi_P(x+\delta) \le 0$	Projection: $\|\delta\|_\infty \le \epsilon^*(x)$
CAT20 [7]	$\psi_P(x+\delta) \le 0$	Projection: $\|\delta\|_\infty \le \epsilon^*(x)$
H-SAT (ours)	$\psi_H(x+\delta) \le \lambda$	Subset updates
P-SAT (ours)	$\psi_P(x+\delta) \le \lambda$	Early termination
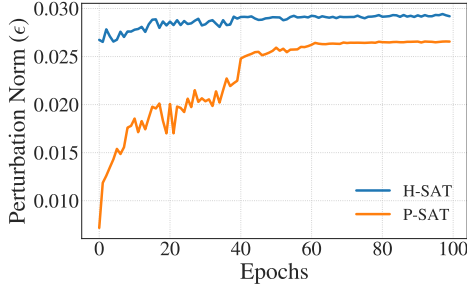
Figure 6: (CIFAR-10) Mean of the perturbation norm (ϵ) of the adversarial examples generated by H-SAT and P-SAT. max. ϵ is set to $8/255$, and no random restart is used to emphasize the trend. Note that H-SAT and P-SAT use different linear schedules on λ.

based on the true curriculum constraint (which is designed to keep the perturbed sample on the decision boundary and not push it further inside the boundary of an incorrect class, $\psi_P(x+\delta) \le 0$).

Initially, $\epsilon^*(x)$ is set to a small value. If the perturbed sample is correctly classified, then $\epsilon^*(x)$ increase and adds difficulty for the next epoch. Conversely, if the perturbed sample is incorrectly classified, $\epsilon^*(x)$ is decreased for IAAT or left unchanged for CAT20. If this approximation works well, then $\epsilon^*(x)$ should be close to the shortest distance from the decision boundary for each training sample. Also due to the choice of the constraint $\psi_P(x+\delta) \le 0$, CAT20 to have high accuracy but low robustness. The other difference between IAAT and CAT20 is that CAT20 also uses label smoothing which could be the cause of the gradient obfuscation that we observed in our experiments.

We claimed that H-SAT and P-SAT implicitly and adaptively increased the effective perturbation norm as training progressed. Intuitively, the perturbation norm increases for two reasons. First, when the difficulty parameter is increased, the curriculum constraint is relaxed and thus, the samples can be perturbed more on average before the constraint is violated or before PGD is terminated. Second, as the network becomes more robust, a larger perturbation is required to generate adversarial examples with the same level of difficulty. We empirically verify this statement by training networks using H-SAT and P-SAT on CIFAR-10 with $\epsilon = 8/255$ and tracking the mean of the perturbation norm for 10 batches each with 128 randomly chosen training samples (Fig. 6).

B.2 Early Termination for P-SAT

Here, we justify the early termination of PGD as a way to solve the curriculum loss for P-SAT. First, we restate and prove Proposition 1 for the binary-class case. Then, we extend it to our heuristic for the multi-class case.

Proposition 1. *In a binary-class problem, an optimum of the curriculum loss with $\psi_P(\cdot)$ as the curriculum constraint can be found by a projected gradient method that terminates when the curriculum constraint is violated.*

PROOF. First, we rewrite the curriculum objective such that it includes the normal loss function. We consider a fixed θ here so we drop the dependence on θ to declutter the notation.

$$\psi_P(x+\delta) \le \lambda \tag{21}$$

$$\iff \max_{j \ne y} f(x+\delta)_j - f(x+\delta)_y \le \lambda \tag{22}$$

$$\iff 1 - f(x+\delta)_y - f(x+\delta)_y \le \lambda \tag{23}$$

$$\iff f(x+\delta)_y \ge (1-\lambda)/2 \tag{24}$$

$$\iff -\log\left(f(x+\delta)_y\right) \le -\log\left((1-\lambda)/2\right) \tag{25}$$

$$\iff \ell(x+\delta) \le \gamma \tag{26}$$

$$\gamma := -\log\left((1-\lambda)/2\right) \tag{27}$$

Now we can rewrite the optimization in Eqn. (10) as

$$\max_{\delta:\|\delta\|_p \le \epsilon} \ell(x+\delta) = \max_{\delta:\|\delta\|_p \le \epsilon} \ell(x+\delta) \tag{28}$$

$$\text{s.t.} \quad \psi_P(x+\delta) \le \lambda \qquad \text{s.t.} \quad \ell(x+\delta) \le \gamma \tag{29}$$

$$= \max_{\delta:\|\delta\|_p \le \epsilon} \min\{\ell(x+\delta), \gamma\} \tag{30}$$

Now we have arrived at the modified optimization problem that only has one convex constraint on the norm of δ. This problem can then be solved with PGD (we call it PGD, but it is a projected gradient ascent). Given a particular sample (x, y), let's consider two possible cases. First, $\nexists \delta$ such that $\|\delta\|_\infty \le \epsilon$ and $\ell(x+\delta) \ge \gamma$. In this case, the new problem reduces to the adversarial loss whose local optima can be found by PGD with a sufficient number of iterations.

The second scenario, $\exists \delta$ such that $\|\delta\|_\infty \le \epsilon$ and $\ell(x+\delta) \ge \gamma$, can be solved if we can find any δ that satisfy the two conditions. Since projected gradient methods satisfy the first condition by default, we only have to run PGD until the second condition is satisfied. Equivalently, PGD can be terminated as soon as the curriculum constraint is violated. □

For the multi-class case, γ is dependent on δ so Eqn. (10) does not reduce to a similar simple form.

$$\psi_P(x + \delta) \leq \lambda \tag{31}$$

$$\iff \max_{j \neq y} f(x + \delta)_j - f(x + \delta)_y \leq \lambda \tag{32}$$

$$\iff f(x + \delta)_y \geq \max_{j \neq y} f(x + \delta)_j - \lambda \tag{33}$$

$$\iff -\log\left(f(x + \delta)_y\right) \leq -\log\left(\max_{j \neq y} f(x + \delta)_j - \lambda\right) \tag{34}$$

$$\iff \ell(x + \delta) \leq \gamma(x + \delta, \lambda) \tag{35}$$

$$\gamma(x + \delta, \lambda) := -\log\left(\max_{j \neq y} f(x + \delta)_j - \lambda\right) \tag{36}$$

Note that we can assume that the RHS on line 2, $\max_{j \neq y} f(x+\delta)_j - \lambda$, is positive. Otherwise, the constraint is automatically satisfied and can be ignored because the LHS $f(x + \delta)_y$ is always non-negative. Similarly to the binary-class case, Eqn. (10) can be rewritten as:

$$\max_{\delta: \|\delta\|_p \leq \epsilon} \ell(x + \delta) \tag{37}$$

$$\text{s.t.} \quad (x + \delta) \leq \lambda \tag{38}$$

$$= \max_{\delta: \|\delta\|_p \leq \epsilon} \ell(x + \delta) \tag{39}$$

$$\text{s.t.} \quad \ell(x + \delta) \leq \gamma(x + \delta, \lambda) \tag{40}$$

$$= \max_{\delta: \|\delta\|_p \leq \epsilon} \min\left\{\ell(x + \delta), \gamma(x + \delta, \lambda)\right\} \tag{41}$$

This objective is difficult to optimized in a few steps of PGD because of the piecewise min and max as well as the fact that the two terms are inversely proportional to the other. Alternatively, we propose a heuristic to approximate the objective by treating the second term as a constant and so not computing its gradients (we still update it as δ changes).

B.3 Early Termination for H-SAT

The early termination can also be applied in this case. However, the approximation of the max eigenvalue of the Hessian vary significantly across models and datasets. Thus, setting a hard threshold is cumbersome and requires a lot fine-tuning. Instead, we propose the following scheme that leverages the high rank-correlation between the true and the approximated eigenvalue to circumvent the above issue of setting hard thresholds.

We choose λ as fraction of samples with the smallest Hessian eigenvalue from each batch to perturb in a given PGD step. For example, when $\lambda = 0.3$, only 30% of the samples in the batch with the smallest $\|H_\epsilon(x, \theta)\|_2$ are perturbed. This method adapts to samples in the batch and is more flexible than a hard threshold. Now, similarly to the probability gap, λ starts off with a small value and increases to 1, which is equivalent to AT, at the end of training. Setting $\lambda = 0$ is equivalent to normal training as no samples are perturbed.

C MAXIMAL HESSIAN EIGENVALUE

C.1 Maximal Eigenvalue Approximation

Below we outline a series of approximations for minimizing the overhead of computing the max. Hessian eigenvalue. We use the second-order Taylor's expansion to approximate the Hessian:

$$\ell(\theta + \zeta) = \ell(\theta) + \zeta^\top \nabla_\theta \ell(\theta) + \frac{1}{2}\zeta^\top \nabla_\theta^2 \ell(\theta)\zeta + O(\|\zeta\|_2^3) \tag{42}$$

Note, we can now maximize over ζ to obtain the max. eigenvalue of the Hessian. The absolute value can be omitted because we are only concerned with the positive eigenvalues.

$$\frac{1}{2}\left\|\nabla_\theta^2 \ell(\theta)\right\|_2 = \max_{\|\zeta\|_2=1} \frac{1}{2}\zeta^\top \nabla_\theta^2 \ell(\theta)\zeta \tag{43}$$

$$= \max_{\|\zeta\|_2=1} \left[\ell(\theta + \zeta) - \zeta^\top \nabla_\theta \ell(\theta)\right] - \ell(\theta) \tag{44}$$

$$\leq \max_{\|\zeta\|_2=1} \ell(\theta + \zeta) - \min_{\|\zeta\|_2=1} \zeta^\top \nabla_\theta \ell(\theta) - \ell(\theta) \tag{45}$$

$$\approx \frac{1}{\alpha}\left\{\ell(\theta + \alpha g) + \alpha \|\nabla_\theta \ell(\theta)\|_2\right\} - \ell(\theta) \tag{46}$$

where g is a shorthand notation of $g(x; \theta) := \nabla_\theta \ell(x; \theta)/\|\nabla_\theta \ell(x; \theta)\|_2$. The first and the second terms in Eqn. (46) approximate the maximization and the minimization in Eqn. (45) by taking a one-step projected gradient update. We can also use a similar approximation to lower bound Eqn. (44) by substituting ζ with αg and $-\alpha g$ and take the maximum between the two:

$$\frac{1}{2}\left\|\nabla_\theta^2 \ell(\theta)\right\|_2 \gtrapprox \frac{1}{\alpha}\max\{\ell(\theta + \alpha g) - \alpha \|\nabla_\theta \ell(\theta)\|_2, \tag{47}$$
$$\ell(\theta - \alpha g) + \alpha \|\nabla_\theta \ell(\theta)\|_2\} - \ell(\theta)$$

Now we have arrived at the upper and the lower bounds as stated in Section 5. Evaluating Eqns. (46) and (47) at the adversarial example of x, gives us the upper and a lower bound respectively, for $\|H_\epsilon(x, \theta)\|_2$. Choosing an appropriate value of α makes the Taylor's series based approximation and the maximization using a single gradient step much more accurate in practice. We choose it to be 1% of the gradient norm so the precision also automatically adapts to the current scale. Choosing α much smaller is not recommended because it can blow up small numerical errors

C.2 Implementation Consideration

Note that the smoothness analysis typically computes eigenvalue of the Hessian matrix of the loss averaged over the entire training set, but for the purpose of curriculum learning, we want to control the Hessian eigenvalue for individual samples. The former quantity can be upper bounded by the average of the latter as follows:

$$\left\|\frac{1}{n}\sum_{i=1}^n \nabla_\theta^2 \ell_\epsilon(x_i; \theta)\right\|_{(2)} \leq \frac{1}{n}\sum_{i=1}^n \|H_\epsilon(x_i, \theta)\|_{(2)} \tag{48}$$

This shows that Hessian eigenvalue can be used as a difficulty metric for curriculum learning and still controls the smoothness of the loss landscape.

We have derived the lower/upper bounds of the maximal Hessian eigenvalue in Section 5 and Appendix C.1. Nonetheless, we face with some difficulty for combining it with AT in practice. To enable a fine-grained sample-wise control on the difficulty metric, we must approximate the Hessian eigenvalue per sample at every PGD step of AT. This is an issue in practice as the automatic differentiation software (e.g., PyTorch) does not provide an easy way to access $\nabla_\theta \ell(x; \theta)$, and evaluation of $\ell(x; \theta + \nabla_\theta \ell(x; \theta))$ cannot be parallelized due to the fact that gradients are different for each sample x. If we ignore the parallelization and compute the perturbed loss for
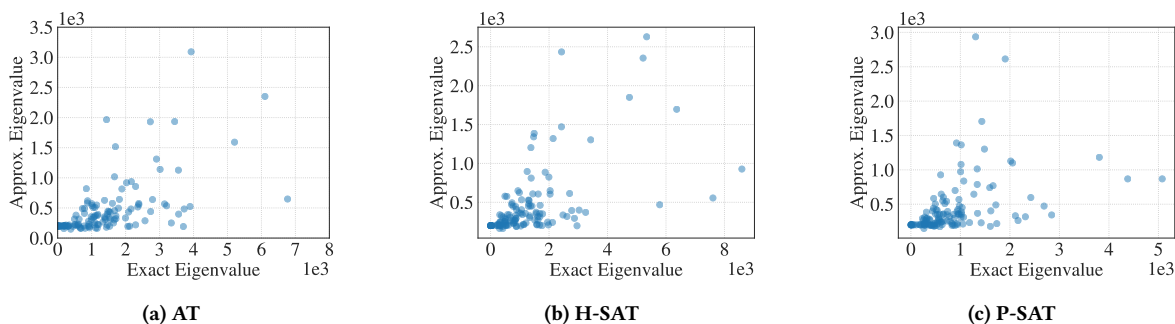
Figure 7: (CIFAR-10) The exact and the approximate maximal eigenvalue of the Hessian computed on three ResNet models: (a) AT, (b) H-SAT, and (c) P-SAT. Each point corresponds to one randomly chosen training sample.

Figure 8: (Imagenette) Randomly selected images from the Imagenette dataset. The first row is the original images while the second and the third are adversarial examples generated with ϵ of 16/255 and 24/255 respectively.

every x in the batch in a sequential manner, the computation time becomes prohibitively expensive (linear in minibatch size).

To reduce the computation, we approximate gradients of individual samples with the minibatch gradient. Obviously, if the minibatch size were set to one, this approximation is exact. The smaller the minibatch size, the more accurate and more expensive this gradient approximation becomes. However, we want to keep the minibatch size fixed across all the defenses we experiment with for a fair comparison. Thus, we avoid the issue by fixing the minibatch size for the weight update to be 128 (same as the other schemes) but using a smaller minibatch size for computing the Hessian eigenvalue.

We determine that a minibatch size of 32 for the Hessian computation is sufficiently accurate and does not introduce too much overhead. We measure the Spearman rank-correlation for the samplewise Hessian eigenvalue computed exactly by the power method and approximately by the upper bound and the heuristic we introduced above. The correlation is above 0.6 in all the cases we test. In Fig. 7, we plot the eigenvalue for 10 batches each with 128 randomly chosen training samples computed on three ResNet's trained with AT, H-SAT, and P-SAT. The correlations are 0.6419, 0.6097, and 0.6456 for the three models respectively.

D EXAMPLES OF ADVERSARIAL IMAGES

Fig. 8 shows 10 examples of the images from the Imagenette dataset, which is a ten-class subset of the ImageNet dataset. The images in the second and the third rows are adversarial examples that are perturbed with ϵ of 16/255 and 24/255 respectively. This figure illustrates that while the choices of perturbation norm we experiment with may seem large compared to 8/255 for CIFAR-10/100, they are very much imperceptible to humans because the images are of much higher resolution (224 by 224 pixels).