Toward Efficient Robust Training against Union of ℓ_p Threat Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

The overwhelming vulnerability of deep neural networks to carefully crafted perturbations known as adversarial attacks has led to the development of various training techniques to produce robust models. While the primary focus of existing approaches has been directed toward addressing the worst-case performance achieved under a single-threat model, it is imperative that safety-critical systems are robust with respect to multiple threat models simultaneously. Existing approaches that address worst-case performance under the union of such threat models (e.g., $\ell_\infty, \ell_2, \ell_1$) either utilize adversarial training methods that require multi-step attacks which are computationally expensive in practice, or rely upon fine-tuning of pre-trained models that are robust with respect to a single-threat model. In this work, we show that by carefully choosing the objective function used for robust training, it is possible to achieve similar, or even improved worst-case performance over a union of threat models while utilizing only single-step attacks during the training, thereby achieving a significant reduction in computational resources necessary for training. Furthermore, prior work showed that adversarial training against the ℓ_1 threat model is relatively difficult, to the extent that even multi-step adversarially trained models were shown to be prone to gradient-masking and catastrophic over-fitting. However, our proposed method—when applied on the ℓ_1 threat model specifically—enables us to obtain the first ℓ_1 robust model trained solely with single-step adversarial attacks. Finally, to demonstrate the merits of our approach, we utilize a modern set of attack evaluations to better estimate the worst-case performance under the considered union of threat models.

1 Introduction

Recent years have demonstrated the success of deep learning in solving machine learning tasks spanning across various domains—computer vision, natural language texts, speech, etc. In addition, it has even exceeded the human level performance for certain tasks [He et al., 2016, 2015]. However, despite their successes, these systems exhibit severe vulnerabilities: Deep learning models are very susceptible to imperceptible perturbations in the input at test time [Szegedy et al., 2013]. Such human-imperceptible noise, known as adversarial attacks, can be used to induce networks to confidently predict incorrect labels, and can thus have disastrous implication in safety critical applications such as autonomous navigation and identity verification. To make models robust against such vulnerabilities at test time, a paradigm of *adversarial robust training* of machine learning models has been developed in recent years [Goodfellow et al., 2015, Madry et al., 2018, Zhang et al., 2019].

These adversarial training procedures have primarily been used to train models robust to a single threat model—perturbations constrained within an ℓ_p-ball of ε_p radius for some p. For instance, the predominant threat model of interest that has been extensively studied in existing literature corresponds to the ℓ_∞ threat model (mostly $\varepsilon_\infty = 8/255$). However, human-imperceptible adversarial perturbations can be sourced from multiple threat-models; hence in practice, it is pertinent to ensure that networks are robust against perturbations from a union of threat models simultaneously. More so, it has been observed that robust training procedures for a chosen threat model are not effective against

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

attacks from other threat models [Tramer and Boneh, 2019, Maini et al., 2020], thus necessitating the development of adversarial defenses against multiple perturbation models simultaneously.

Over recent years, training procedures have been proposed to make systems simultaneously robust against perturbations constrained within a union of ℓ_∞, ℓ_1 and ℓ_2 balls. Systems trained in such manner are then evaluated over the worst-case performance across perturbations from all the threat-models. Tramer and Boneh [2019] proposed simple aggregations of different adversaries for adversarial training against multiple perturbation models utilizing multi-step adversarial attacks for robust training. Maini et al. [2020] further established SOTA for adversarial accuracy against union of (ℓ_∞, ℓ_1, ℓ_2) perturbations through the adversarial training procedure Multi Steepest Descent (or, MSD) that also uses multi-step ($k = 50$) adversarial attacks to generate adversaries for training. However, these methods, owing to their requirement of great number of adversarial training steps as compared to a regular setting for multi-step adversarial training procedure (10 steps), are computationally inefficient. This leads to our research question: Is it possible to achieve worst-case performance over a union of threat models that is similar to that of the SOTA methods, while utilizing training procedures that requires only single-step attacks to generate adversaries? We answer the same in affirmation: we first analyse failure modes of existing approaches during ℓ_1 based adversarial training, and thereby propose to use a dynamic curriculum schedule to effectively mitigate robust overfitting. Furthermore, we extend this approach to develop a training routine that utilizes a single-step adversarial training across a union of threat models to be robust against them simultaneously. In summary, we make the following contributions in this work:

- We demonstrate the first successful single-step robust training procedure, NCAT-ℓ_1, to achieve ℓ_1 robustness by using a curriculum schedule with Nuclear Norm based training.
- We extend this approach to propose a training procedure NCAT, that yields SOTA-like robust accuracy under the union of multiple ℓ_p threat models, while requiring only a single-step attack budget per minibatch.
- We further demonstrate that the proposed defense can scale-up to high-capacity networks and large-scale datasets such as ImageNet-100. Additionally, NCAT trained models generalize to unseen threat models, achieving near-SOTA robustness even on Perceptual Projected Gradient Descent (PPGD), which comprises one of the strongest attacks known to date.

2 Preliminaries

Here, we lay down the notations and conventions used in this work. We denote x to be a d-dimensional image from an N-class dataset $\mathcal{D}$, while its corresponding ground-truth label as a one-hot vector y. f_θ represents a Deep Neural Network with parameters θ, that maps an input image x to its pre-softmax output $f_\theta(x)$. The cross-entropy loss corresponding to the network prediction on a sample (x, y) is denoted as $\ell_{CE}(f_\theta(x), y)$. For a minibatch $B = \{(x_i, y_i)\}_{i=1}^{M}$, we denote X as the image matrix whose i^{th} row consists of flattened pixel intensities of the image x_i, and Y as the corresponding ground-truth array. Thus, X is a matrix of size $(M \times d)$, and Y is a matrix of size $(M \times N)$. Also, $\ell_{CE}(f_\theta(X), Y)$ now denotes the sum of cross-entropy losses over all data samples in the minibatch B. Further, for a matrix A, let $\|A\|_*$ denote the Nuclear Norm, the sum of the singular values, of A.

Adversarial Threat Model: In this work, we primarily consider the robustness of Deep Networks against the union of ℓ_∞, ℓ_1, and ℓ_2 constrained adversaries. Thus under the ℓ_∞ threat model with an ε_∞-constraint, for a given a clean image x, an adversarially perturbed counterpart $\tilde{x}$ can differ by at most ε at any given pixel location. In contrast, adversaries under the ℓ_1 constraint may differ from the original image such that the sum of pixel-wise absolute differences are capped by ε_1. Similarly, for adversaries under ℓ_2 constraint, the sum of squared pixel-wise differences are capped by ε_2^2. Further, a network f_θ is said to be ε_p-robust under a threat model ℓ_p on a clean sample x with label y, if $f_\theta(\tilde{x}) = y$, for all perturbations $\tilde{x}$ such that $\|x - \tilde{x}\|_p \leq \varepsilon_p$.

3 Related Works

In this section we briefly discuss the adversarial attacks and defences that builds up to efficient multi-step adversarial training procedures, work that introduces adversarial training against the union of multiple threat models, and their limitations that we propose to alleviate.

3.1 Early attempts towards Robust training

Early attempts of improving robustness to adversarial attacks could be broadly classified into input pre-processing based defenses and adversarial training methods. While the input pre-processing based defenses were computationally cheap, they primarily involved masking of gradients to prevent the generation of strong attacks. Such methods [Guo et al., 2018, Xie et al., 2018, Song et al., 2018] were later shown to be ineffective against attacks constructed using smooth approximations of the non-differentiable components [Athalye et al., 2018, Carlini et al., 2019].

3.2 Effectiveness of FGSM and its limitations

Amongst the latter methods, the most successful early defense which stood the test of time was the Projected Gradient Descent or PGD adversarial training [Madry et al., 2018]. This involved minimization of cross-entropy loss on the worst-case perturbations generated using multiple iterations of constrained optimization, leading to a significantly higher computational cost when compared to standard training. Multi-step defenses achieve the state-of-the-art robustness today and typically use 10-steps of optimization for attack generation, leading to 11 times higher forward and backward propagations. FGSM or the Fast Gradient Sign Method [Goodfellow et al., 2015] based adversarial training alleviates the computational cost by utilizing single-step adversarial samples for training. However, this method is contingent on the local linearity assumption of the loss surface, which is often compromised during the process of FGSM training, leading to generation of weaker adversaries as training progresses. Indeed, Kurakin et al. [2017] showed that such models exhibited the phenomenon of gradient masking, wherein stronger multi-step attacks were seen to reduce the robust accuracy drastically. Wong et al. [2020] proposed to incorporate early-stopping using R-FGSM based adversarial training [Tramèr et al., 2018], in order to identify the failure-point during robust training with single-step adversaries. However, the method was later shown to not be effective on large capacity networks such as the WideResNet [Zagoruyko and Komodakis, 2016] architecture in subsequent work [Sriramanan et al., 2020].

3.3 NuAT: Efficient Adversarial training for ℓ_∞ threat model

Sriramanan et al. [2021] proposes a Nuclear Norm regularizer to improve the adversarial robustness of Deep Networks through the use of single-step adversarial training under ℓ_∞ constraints. This Nuclear Norm Adversarial Training (NuAT) enforces function smoothing in the vicinity of clean samples by incorporating joint batch-statistics of adversarial samples, which results in enhanced robustness. Further, this limits the oscillation of function values and prevents the over-smoothing of loss surface uniformly in all directions, leading to a better robustness-accuracy trade-off.

Formally, in a given minibatch B, if X is the matrix composed of row-wise vectorized pixel values of each image, Δ is a matrix of the same dimension as X consisting of independently sampled Bernoulli noise, and Y is the matrix containing the corresponding ground truth one-hot vectors, maximization of the following loss function that utilizes the pre-softmax values $f_\theta(\cdot)$ generates single-step adversaries:

$$\Delta^* = \arg\max_{\Delta} \left[\ell_{CE}\left(f_\theta(X+\Delta), Y\right) + \lambda \cdot ||f_\theta(X+\Delta) - f_\theta(X)||_*\right] \tag{1}$$

Subsequently, the following loss function is minimized during the NuAT training:

$$\min_{\theta} \left[\ell_{CE}\left(f_\theta(X), Y\right) + \lambda \cdot ||f_\theta(X+\Delta^*) - f_\theta(X)||_*\right] \tag{2}$$

3.4 Union of Threat Models

While above described works trains a target to be robust against a single threat model of ℓ_p-ball, there has been recent effort in the direction of making models robust against multiple threat models simultaneously. Tramer and Boneh [2019] study the theoretical and empirical trade-offs of adversarial robustness in various settings when defending against aggregations of multiple adversaries, proposing to train on the average (AVG) or maximizers of loss (MAX) amongst the different threat models considered for each minibatch of samples. Madaan et al. [2020] train using perturbations generated using a Meta-Noise Generator, and also propose a variant, Stochastic Adversarial Training wherein they utilize multi-step adversaries (10 steps for ℓ_∞ and ℓ_2, 20 steps for ℓ_1), though the authors note sub-optimal performance from the same. Croce and Hein [2019] propose a provable adversarial defense

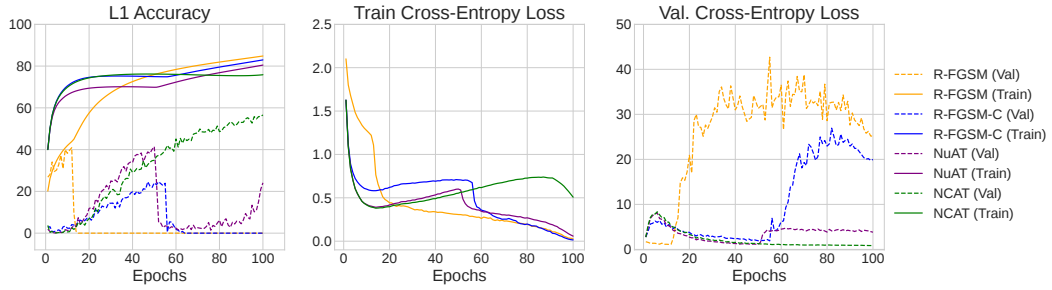

Figure 1: **Catastrophic Overfitting in ℓ_1 Adversarial Training:** To analyze the stability of single-step training, we plot accuracy (left) and cross-entropy losses (Right) over epochs of different single-step adversarially trained models. With R-FGSM based adversarial training [Wong et al., 2020], catastrophic overfitting occurs with extreme gradient masking (orange); adversarial accuracy (loss) is high (low) on the train set, while being close to zero (high) for validation images. More so, even using a curriculum schedule used for ℓ_1 adversaries during training only delays the overfitting (blue). In contrast, the proposed training approach NCAT (green) does not display catastrophic overfitting due to gradient masking.

against all ℓ_p norms for $p \geq 1$ using a regularization term for ReLU networks, by enforcing robustness against ℓ_∞ and ℓ_1 adversaries in particular. Lastly and most importantly, Maini et al. [2020] develops a generalization of the standard PGD-based approach to incorporate multiple perturbation models into a single attack by introducing a procedure called Multi Steepest Descent, or MSD, and further utilizing it to train standard architectures that are simultaneously robust against ℓ_∞, ℓ_1 and ℓ_2 adversaries.

3.4.1 Multi Steepest Descent (MSD)

The core idea that MSD [Maini et al., 2020] adopts, which helps establish better worst-case accuracies against the union of adversaries, is to create a single adversarial perturbation by simultaneously maximizing the worst-case loss over all perturbation models at each projected steepest descent step. Unlike previous approaches [Tramer and Boneh, 2019] that generate worst-case adversaries for each threat model, or augment adversaries from multiple threat models, MSD chooses a projected steepest descent direction in each iteration that maximizes the loss over all threat models. This has been established to be superior to the standard adversarial training and the simpler approaches that use comparatively myopic PGD subroutines that only use one perturbation model at a time. However, a clear limitation of MSD here is that it requires 50 adversarial attack steps for each training iteration. Additionally, for each training step, it performs three forward passes (one for each threat model) and a backward pass.

3.4.2 Extreme Norms Adversarial Training (EAT)

In order to achieve robustness against a union of ℓ_p threat models, Croce and Hein [2021b] propose to fine-tune models that were originally trained to be robust against a single ℓ_p norm threat model. The authors demonstrate that fine-tuning of robust models to previously unseen ℓ_p threat models is effective, in contrast to adversarial fine-tuning of normally trained networks which yields non-robust models. Furthermore, the authors propose to train solely on ℓ_1 and ℓ_∞ adversaries, such that other ℓ_p balls on interest are contained within the union of these two threat models (Extreme Norms Adversarial Training or EAT). However, this can place excessive restrictions during robust training if the perturbation budget of intermediate ℓ_p adversaries is large. As with MSD [Maini et al., 2020], EAT is computationally expensive in practice, since it relies upon multi-step adversarially pre-trained models, and further performs robust fine-tuning of such models using 10-step adversaries in the second phase.

4 Proposed Method

As noted by prior works [Madry et al., 2018, Croce and Hein, 2021a, Maini et al., 2020], robust training against the ℓ_1 threat model is significantly more complicated when compared to standard adversarial training techniques for ℓ_∞ or ℓ_2 threat models. Croce and Hein [2021a] note that even adversarial training using expensive 10-step adversaries generated from SLIDE [Tramer and Boneh,

2019] is prone to catastrophic overfitting [Wong et al., 2020]: Over the course of training, models overfit to the adversaries generated, leading to a false notion of being robust, while achieving close to 0% accuracy against stronger attacks during test evaluation. While such phenomena are frequently seen in single-step training [Goodfellow et al., 2015], the occurrence of such failure modes even with 10-step adversaries exhibits the difficulty in training ℓ_1 robust networks. Recent work [Wong et al., 2020, Sriramanan et al., 2020, 2021] has demonstrated that such failure modes can be mitigated through appropriate algorithmic choices such as validation-based early stopping using PGD adversaries, or relaxation terms for the overall loss to prevent collapse in training. Croce and Hein [2021a] demonstrate that using the 10-step APGD ℓ_1 attack, robust models can be trained by automatically tuning the sparsity level induced in the ℓ_1 perturbations seen during training. Building upon these in this work, we demonstrate the first successful instance of achieving non-trivial ℓ_1 robustness using single-step adversaries during training. Further, we extend the technique to achieve simultaneous robustness against the union of ℓ_p threat models using only single-step training.

4.1 Analyzing Robust Training with ℓ_1 Adversaries

As mentioned before, Croce and Hein [2021a] point out the intricacies involved with ℓ_1 adversarial training, in that even multi-step training methods can begin to catastrophically overfit. In this work, we seek to identify efficient yet effective single-step training routines that achieve robustness against ℓ_1 adversaries. In order to build upto that, we first focus on understanding the phenomenon of catastrophic overfitting under this setting and analyze what methods can help alleviate it in the single-step setting.

4.1.1 Nuclear Norm Attack and Curriculum Schedule

We begin by plotting the prediction accuracy and cross-entropy loss of different models over training and validation (Figure 1). We find that when trained with R-FGSM based adversaries, models suffer from catastrophic overfitting early on during the training. However, we make a crucial observation that dynamically varying the perturbation budget during training, effectively setting up a *curriculum*, helps immensely in improving overall stability of training. For instance, with the final ℓ_1 threat model of interest given by the ball of radius 12, we propose to linearly increase this parameter from 0 to 12 to prevent catastrophic overfitting. However, applying this curriculum to RFGSM-AT only leads to a delay in catastrophic failure, indicating the unsuitability of using R-FGSM adversaries for robust training.

Since the goal here is to utilize only single adversarial training step, it becomes imperative that the loss that we optimize over to generate the adversaries is smooth and does not showcase gradient masking. Hence, we build upon Nuclear-Norm Adversarial Training (NuAT) [Sriramanan et al., 2021] to generate single-step adversaries. NuAT proposed to generate ℓ_∞-adversaries by maximizing the Nuclear Norm regularized objective. We apply the same for generating adversaries like follows:

$$\widetilde{L} = \ell_{CE}\left(f_\theta(X+\Delta), Y\right) + \lambda \cdot ||f_\theta(X+\Delta) - f_\theta(X)||_* \tag{3}$$

Sriramanan et al. [2021] note that since the Nuclear norm forms a tight convex relaxation for the rank of the predicted matrix of logit values, the corresponding attack generates diverse adversaries in a given minibatch, which then helps mitigate robust overfitting. Crucially, we observe however that this supplemental attack diversity is not sufficient for single-step training on the ℓ_1 threat model, as even NuAT is observed to be susceptible to catastrophic failure in Fig-1. However, by utilizing the dynamic curriculum schedule, this phenomenon is successfully remedied in the proposed method, NCAT.

4.1.2 Steepest Ascent with Single-Step Optimization

Given a minibatch of samples X, we generate the Nuclear Norm attack by identifying a perturbation Δ that maximizes the loss $\widetilde{L}$ as in Eq-3 and conforms to the following two constraints:

$$||\Delta||_1 \leq \epsilon_1 \ , \ X + \Delta \in [0,1]^d$$

wherein the second constraint arises from the normalized range for pixel intensities of an image. Assuming a first-order Taylor series approximation for the loss incurred by the network f_θ, if $\nabla_\Delta \widetilde{L}$ represents the gradient direction, the steepest ascent direction Δ^* to maximize loss would be parallel to the same in the unconstrained setting. However, to incorporate the two constraints entailed above, we now derive the optimal perturbation (Eq-4). For an image x in the minibatch X, let g represent the corresponding gradient. Further, let M_i denote the deviation budget required at pixel i to saturate the same to the pixel constraint $[0,1]$ parallel to the gradient g_i. Formally, M can be defined as:

$$M_i = \begin{cases} 1 - x_i & \text{if } g_i \geq 0 \\ -x_i & \text{if } g_i < 0 \end{cases}$$

Finally, to incorporate the ℓ_1 budget being constrained by ε_1, we observe the optimal perturbation prioritizes pixel locations corresponding to large gradient magnitudes. Let σ denote the sorted permutation of indices such that $a_{\sigma(1)} \geq a_{\sigma(2)} \geq \cdots \geq a_{\sigma(d)}$, where $a_i = |g_i|$ represents the gradient magnitudes at different pixel locations. Now, if the cumulative budget utilized is defined as $S_i = \Sigma_{j=0}^{i} |M_{\sigma(j)}|$ and $I_i = \max\{0, \varepsilon_1 - S_{i-1}\}$ denoting support variables, the optimal perturbation corresponding to image x can be defined as:

$$\delta^*_{\sigma(i)} = \begin{cases} M_{\sigma(i)} & \text{if } S_i \leq \varepsilon_1 \\ M_{\sigma(i)} \cdot I_i & \text{if } S_i > \varepsilon_1 \end{cases} \tag{4}$$

Thus, in an ℓ_1 constrained attack the gradients have to be sorted by their magnitude, which requires $O(d \cdot \log d)$ complexity where d represents the image dimensionality. However, in practice this overhead is observed to be exceedingly minimal relative to data loading times etc. Indeed, using this routine with Nuclear norm based training with the curriculum schedule (NCAT-ℓ_1), we demonstrate for the first time that single-step training can be effectively used to produce ℓ_1 robust models.

4.2 Single-Step Training for ℓ_∞ and ℓ_2 Robustness

Since the ℓ_∞ threat model is the most well-studied setting in existing literature, we rely upon prior works to obtain excellent baselines. To achieve robustness against ℓ_∞ constrained adversaries using single-step training, we utilize the current state-of-the-art method, Nuclear Norm Adversarial Training (NuAT) [Sriramanan et al., 2021]. We further seek to incorporate other threat models during training, in order to obtain models with non-trivial robustness against the union of the ℓ_1, ℓ_2 and ℓ_∞ threat models simultaneously. In order to efficiently train against ℓ_2 adversaries, we first propose to modify the NuAT training algorithm to utilize this constraint set, using ℓ_2 norm based projections. However, similar to Croce and Hein [2021b], we make the remarkable observation that models that are trained solely on ℓ_∞ adversaries achieve a great degree of robustness versus ℓ_2 adversaries on the test set. We observe similar transfer of robustness from ℓ_1 trained models toward the ℓ_2 threat model as well. Thus, the primary difficulty in achieving robustness to the union of threat models appears to be that of training networks robust to the ℓ_1 and ℓ_∞ threat models in particular.

4.3 Sampling Procedures to Improve Efficiency

To achieve robustness against the union of the three ℓ_p threat models considered, it is plausible that training with three distinct single-step attacks (constrained to ℓ_1, ℓ_2 and ℓ_∞) using the proposed approach in each minibatch will be effective. However, in this work, we primarily focus on reducing the training complexity further, in order to effectively utilise only a single-step attack for each minibatch. Awasthi et al. [2021] proposed to utilize the multiplicative weights algorithm, wherein the loss under different adversaries on a hold-out validation set guides the sampling procedure, using a set of exponential running weights w_i for each threat model. However, we find that this is contingent on the efficacy of adversaries utilized on the validation set, which can be restrictive in practice due to computational constraints. Building upon this, we find in practice that alternating between ℓ_∞ and ℓ_1 attacks across different minibatches with a fixed frequency is remarkably effective. Thus, the proposed defense, NCAT, uses nuclear norm based single-step training following a curriculum schedule, such that different threat models are selected for attack generation in different minibatches based on a pre-fixed frequency. We present a concise, summarised overview of the proposed training approaches in Algorithm-1. Here, the Decision Function D (L7, Alg-1) alternately outputs $p = 1$ or $p = \infty$ based on a predetermined frequency, since such models are observed to simultaneously achieve ℓ_2 robustness without explicit training. As observed in prior works [Chen et al., Sriramanan et al., 2021], maintaining a exponential running average of network weights (SWA [Izmailov et al., 2018]) helps improve robust performance overall as well, particularly so in this setup since different (random) minibatches are trained with adversarial perturbations arising from different threat models. Furthermore, this effectively reducing undesired bias to a particular threat model due to auto-correlations that arise in training. A modified version of the proposed approach, namely NCAT-AVG,

Algorithm 1 Nuclear Curriculum Adversarial Training for ℓ_p Norm Robustness

Input: Network f_θ with parameters θ, Weight Averaged Network f_ω with parameters ω, Training Data $\mathcal{D}$ with input images of dimension d, Minibatch Size M, Attack Size ε_p for each ℓ_p threat model, Epochs E, Learning Rate η, Decision Function D, Curriculum Schedule $\mathcal{C}$
for $epoch = 1$ **to** E **do**
  $\varepsilon_p = \mathcal{C}(p)$
  for minibatch $\{(x_i, y_i)\}_{i=1}^{M} \subset \mathcal{D}$ **do**
    $X = \begin{bmatrix} \dots & x_1 & \dots \\ \dots & \vdots & \dots \\ \dots & x_M & \dots \end{bmatrix}, \quad \Delta = \begin{bmatrix} \dots & \delta_1 & \dots \\ \dots & \vdots & \dots \\ \dots & \delta_M & \dots \end{bmatrix}, \quad \delta_i \sim Bern^d(-\alpha, \alpha), \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$
    $\widetilde{L} = \ell_{CE}\left(f_\theta(X + \Delta), Y\right) + \lambda \cdot ||f_\theta(X + \Delta) - f_\theta(X)||_*$
    for p in $\mathrm{D}(\theta)$ **do**
      $\Delta = \Delta + \varepsilon_p \cdot \mathrm{Proj}\Big(\nabla_\Delta \widetilde{L}, B_p(\varepsilon_p)\Big)$
      $\widetilde{X} = Clamp\left(X + \Delta, 0, 1\right)$
    end for
    $L = \ell_{CE}(f_\theta(X), Y) + \lambda \cdot ||f_\theta(\widetilde{X}) - f_\theta(X)||_*$
    $\theta = \theta - \frac{1}{M} \cdot \eta \cdot \nabla_\theta L$
    $\omega = \tau \cdot \omega + (1 - \tau) \cdot \theta$
  end for
end for

uses a Decision Function D that outputs the collection of $p = \{1, 2, \infty\}$, and effectively uses a budget of three single-step attacks, one for each threat model. We present results obtained using NCAT-AVG and other ablations in the Supplementary Material.

5 Experiments and Analysis

In this work, we primarily consider the CIFAR-10 [Krizhevsky et al., 2009] and ImageNet-100 [Russakovsky et al., 2014] datasets, since they have come to form the benchmark for comparative analysis of adversarially robust models. Following prior works [Maini et al., 2020], we consider constraint sets given by the ℓ_∞ ball of radius $8/255$, ℓ_2 ball of radius 0.5 and ℓ_1 ball of radius 12 as the threat models of interest, and as explained previously, we attempt to train models that achieve non-trivial worst-case accuracy against the union of such ℓ_p threat models. For the ImageNet-100 dataset, the corresponding radii for ℓ_1, ℓ_2 and ℓ_∞ threat models are 255, 1200/255 and 4/255 respectively, following the constraints considered by Laidlaw et al. [2021].

We present results in the white-box setting, wherein the adversary is cognizant of the model weights, architecture and training scheme employed. To accurately estimate worst-case performance, we focus our evaluation pipeline to incorporate state-of-the-art attacks such as AutoAttack [Croce and Hein, 2020] for each ℓ_p threat model. Furthermore, AutoAttack includes strong ℓ_1 attack evaluation baselines using techniques proposed by Croce and Hein [2021a], wherein the authors note that significant improvement in attack efficacy as compared to prior works such as SLIDE Attack [Tramer and Boneh, 2019], B&B Attack [Brendel et al., 2017] and Pointwise Attack [Schott et al., 2018]. Due to paucity of space, we include black-box evaluations, generalization to unseen domains, gradient masking checks and adaptive attacks in the Supplementary Material. We however make the important note that the suite of white-box attacks considered using AutoAttack are *strictly stronger* than the former set of attack evaluations, indicating the absence of gradient masking.

We first present results obtained using the ResNet-18 [He et al., 2016] architecture on CIFAR-10 in Table-1. In the first partition of the table, we present models trained solely on the ℓ_1 threat model. The current state-of-the-art is achieved by APGD-ℓ_1 [Croce and Hein, 2021a], which performs a 10-step APGD attack during training in order to mitigate gradient masking and catastrophic overfitting for ℓ_1 constrained adversaries. On the other hand, our method, NCAT-ℓ_1 which uses just a single-step attack for adversarial training achieves ℓ_1 robustness much more efficiently. We note that while the multi-step approach has higher ℓ_1 robustness (+3.6%), the single-step NCAT trained model has

Table 1: **Consolidated Results on CIFAR-10:** Prediction accuracy (%) of ResNet-18 models trained using different methods under various threat models. Robust evaluations are presented under the constraint sets given by the ℓ_1 ball of radius 12, ℓ_2 ball of radius 0.5 and ℓ_∞ball of radius $8/255$ comprising the individual threat models of interest, along with worst-case and average-case performance under the union of these threat models.

Method	**Number of AT Steps**	**Clean Acc**	**Worst-Case Acc**	**Average Acc**	ℓ_1 **Acc**	ℓ_2 **Acc**	ℓ_∞ **Acc**
			ℓ_1 Training Alone				
APGD-ℓ_1	10	85.9	22.1	48.8	59.5	64.9	22.1
NCAT-ℓ_1	1	81.1	37.9	53.6	55.9	67.0	38.0
			Training under Union of Threat Models				
SAT	$13.33^\dagger$	83.9	40.4	54.2	54.0	68.0	40.7
AVG	30	84.6	40.1	53.8	52.1	68.4	40.8
MAX	30	80.4	44.0	53.4	48.6	66.0	45.7
MSD	50	81.1	43.9	53.4	49.5	65.9	44.9
EAT	$10^{\dagger\dagger}$	82.2	42.4	54.6	53.6	67.5	42.7
NCAT	1	80.3	42.6	53.3	46.9	67.0	46.0
NCAT$^+$	1	77.5	43.7	53.4	48.4	65.7	46.1

Table 2: **Consolidated Results on CIFAR-10:** Prediction accuracy (%) of WideResNet-28-10 models trained using different methods under various threat models. Robust evaluations are presented under the constraint sets given by the ℓ_1 ball of radius 12, ℓ_2 ball of radius 0.5 and ℓ_∞ball of radius $8/255$ comprising the individual threat models of interest, along with worst-case and average-case performance under the union of these threat models.

Method	**Number of AT Steps**	**Clean Acc**	**Worst-Case Acc**	**Average Acc**	ℓ_1 **Acc**	ℓ_2 **Acc**	ℓ_∞ **Acc**
			ℓ_1 Training Alone				
APGD-ℓ_1	10	83.7	30.7	52.5	61.6	65.1	30.7
NCAT-ℓ_1	1	80.7	39.2	54.6	56.1	68.6	39.3
			Training under Union of Threat Models				
SAT	$13.33^\dagger$	80.5	45.7	56.2	55.9	66.7	45.9
AVG	30	82.5	45.1	56.1	55.0	68.0	45.4
MAX	30	79.9	47.4	54.6	50.2	65.3	48.4
MSD	50	80.6	46.9	55.1	51.7	65.6	48.0
EAT	$10^{\dagger\dagger}$	79.9	46.4	56.3	56.0	66.2	46.6
NCAT	1	81.5	44.6	54.8	49.9	68.3	46.3

significantly better worst-case accuracy over all three threat models (+15.8). Indeed, we note once again that NCAT represents the first-ever successful single-step adversarial training on the ℓ_1 threat model, which also generalizes well to the unseen ℓ_∞ and ℓ_2 threat models simultaneously.

In the second partition of Table-1, we present models that are explicitly trained to be robust under the union of the ℓ_∞, ℓ_2 and ℓ_1 threat models. Namely, we present comparative analysis with respect to existing multi-step adversarially trained defenses such as AVG and MAX [Tramer and Boneh, 2019], MSD [Maini et al., 2020], EAT [Croce and Hein, 2021b] and SAT [Madaan et al., 2020]. For these methods, we primarily utilize robust evaluations as presented by Croce and Hein [2021b] to enable fair comparisons, which comprise of re-implemented models that obtain higher accuracies as compared to values reported in the original papers. We first note that SAT† requires 13.33 adversarial attack steps during training, since it utilizes 10-step attacks for ℓ_∞ and ℓ_2 adversaries, and 20 attack steps for ℓ_1 adversaries to mitigate gradient masking, indicating the considerable difficulty involved in achieving ℓ_1 robustness. In contrast, EAT†† relies upon 10-step fine-tuning of a network that is already robust against a single threat-model. The current state-of-the-art approaches comprise

Table 3: **Results on ImageNet-100:** Prediction accuracy (%) of models trained using different methods under various threat models. Robust evaluations are presented under the constraint sets given by the ℓ_1 ball of radius 255, ℓ_2 ball of radius $1200/255$ and ℓ_∞ball of radius $4/255$ comprising the individual threat models of interest, along with worst-case and average-case performance under the union of these threat models.

Method	**Number of AT Steps**	**Arch /**	**Clean Acc**	**Worst-Case Acc**	**Average Acc**	ℓ_1 **Acc**	ℓ_2 **Acc**	ℓ_∞ **Acc**	**PPGD Acc**
ℓ_∞-AT	10	RN50	81.7	0.8	20.7	0.8	3.7	55.7	1.5
PAT	10	RN50	72.6	37.8	41.2	41.2	37.7	45.0	29.2
NCAT-ℓ_1	1	RN18	64.9	41.1	43.9	48.3	41.4	42.1	26.6
NCAT	1	RN18	63.9	41.5	44.8	46.8	41.9	45.7	29.1

of MSD and MAX that achieve 44% worst-case accuracy, while utilizing a budget of 50 and 30 attack steps respectively during training. We observe that the proposed approach, NCAT achieves comparable worst-case and average-case performance over the threat models considered, while requiring a significantly smaller computational footprint during training. Furthermore, the proposed method facilitates the trade-off between clean accuracy and worst-case performance as indicated by NCAT$^+$ which achieves near-SOTA robust performance.

In Table-2, we present results on models trained with the WideResNet-28-10 [Zagoruyko and Komodakis, 2016] architecture to demonstrate the scalability of the proposed defense to high-capacity networks. We thus establish the efficacy of the curriculum schedule combined with nuclear norm based training in mitigating catastrophic overfitting, enabling efficient training of these large networks. In Table-3, we present evaluations on the ImageNet-100 dataset, wherein we utilize ResNet-18 networks to reduce computational demands. We observe that NCAT-ℓ_1 attains remarkable robust accuracy on unseen ℓ_∞ and ℓ_2 adversaries, even achieving 26.6% accuracy against the Perceptual Projected Gradient Descent (PPGD) attack [Laidlaw et al., 2021], which forms one of the strongest attacks known to date. Further, we observe that the NCAT trained achieves state-of-the-art worst-case ℓ_p accuracy, while attaining robustness similiar to that of Perceptual Adversarial Training [Laidlaw et al., 2021] for the PPGD attack, with the latter being a model that was explicitly trained on such adversaries.

6 Conclusions

In this work, we develop an efficient adversarial training procedure, NCAT, to train networks that are robust against a union of ℓ_p threat models, namely ℓ_∞, ℓ_1 and ℓ_2. To do so, we first focus on developing an efficient, yet effective robust training procedure for the ℓ_1 threat model, by incorporating a curriculum schedule to mitigate catastrophic overfitting. Indeed, in this work we present the first ℓ_1 constrained robust model trained solely using single-step adversaries, achieving robustness similar to that of multi-step SOTA approaches. Furthermore, we extend the proposed method to achieve worst-case robustness under multiple ℓ_p norm constraints simultaneously. Compared to the current SOTA that uses 30 adversarial attack steps for its training procedure to achieve 44% robust accuracy on CIFAR-10, our method yields 43.7% robustness while solely utilizing single-step adversaries during the training routine. This thereby greatly reduces the computational requirements needed to achieve SOTA-equivalent robust performance.

7 Limitations and Societal Impact

The proposed approach, NCAT, facilitates efficient yet effective training of robust models, that achieve non-trivial worst-case accuracy under the union of several ℓ_p norm threat models. Furthermore, NCAT trained models are seen to robust against unseen attacks and adversaries that might potentially be encountered in the real world. Thus, given that NCAT helps train reliable and trustworthy models using a small computational footprint, this work indeed has potential in creating a positive impact on society. Further, we hope that future works could extend NCAT to include provable certificates for different ℓ_p threat models and their union, and possibly provide stricter guarantees for the absence of gradient masking and catastrophic failure with the use of theoretically-motivated curriculum schedules.

References

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018. 3

P. Awasthi, G. Yu, C.-S. Ferng, A. Tomkins, and D.-C. Juan. Adversarial robustness across representation spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7608–7616, 2021. 6

W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 7

N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 3

T. Chen, Z. Zhang, S. Liu, S. Chang, and Z. Wang. Robust overfitting may be mitigated by properly learned smoothening. 6

F. Croce and M. Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. *arXiv preprint arXiv:1905.11213*, 2019. 3

F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 7

F. Croce and M. Hein. Mind the box: l_1-APGD for sparse adversarial attacks on image classifiers. 139: 2201–2211, 2021a. 4, 5, 7

F. Croce and M. Hein. Adversarial robustness against multiple l_p-threat models at the price of one and how to quickly fine-tune robust models to another threat model. May 2021b. 4, 6, 8

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 3, 5

C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018. 3

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv: Arxiv-1502.01852*, 2015. 1

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 7

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 6

A. Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 7

A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017. 3

C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021. 7, 9

D. Madaan, J. Shin, and S. J. Hwang. Learning to generate noise for multi-attack robustness, 2020. URL https://arxiv.org/abs/2006.12135. 3, 8

A. Madry, A. Makelov, L. Schmidt, T. Dimitris, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 4

P. Maini, E. Wong, and J. Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1909.04068. 2, 4, 7, 8

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL http://arxiv.org/abs/1409.0575. 7

L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on MNIST. Sept. 2018. 7

Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018. 3

G. Sriramanan, S. Addepalli, A. Baburaj, and R. Venkatesh Babu. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 5

G. Sriramanan, S. Addepalli, A. Baburaj, and V. B. Radhakrishnan. Towards efficient and effective adversarial training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kuK2VARZGnI. 3, 5, 6

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013. 1

F. Tramer and D. Boneh. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3, 4, 7, 8

F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018. 3

E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020. 3, 4, 5

C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations (ICLR)*, 2018. 3

S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3, 9

H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019. 1

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section-6,7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section-6,7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Further details included in the Supplementary Material
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Details included in the Supplementary Material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details included in the Supplementary Material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Details included in the Supplementary Material
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details included in the Supplementary Material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Further details included in the Supplementary Material
   (b) Did you mention the license of the assets? [Yes] Details included in the Supplementary Material
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Details included in the Supplementary Material

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]