Revisiting Outer Optimization in Adversarial Training

Ali Dabouei, Fariborz Taherkhani, Sobhan Soleymani, and Nasser M. Nasrabadi

West Virginia University
{ad0046, ft0009, ssoleyma}@mix.wvu.edu, nasser.nasrabadi@mail.wvu.edu

Abstract. Despite the fundamental distinction between adversarial and natural training (AT and NT), AT methods generally adopt momentum SGD (MSGD) for the outer optimization. This paper aims to analyze this choice by investigating the overlooked role of outer optimization in AT. Our exploratory evaluations reveal that AT induces higher gradient norm and variance compared to NT. This phenomenon hinders the outer optimization in AT since the convergence rate of MSGD is highly dependent on the variance of the gradients. To this end, we propose an optimization method called ENGM which regularizes the contribution of each input example to the average mini-batch gradients. We prove that the convergence rate of ENGM is independent of the variance of the gradients, and thus, it is suitable for AT. We introduce a trick to reduce the computational cost of ENGM using empirical observations on the correlation between the norm of gradients w.r.t. the network parameters and input examples. Our extensive evaluations and ablation studies on CIFAR-10, CIFAR-100, and TinyImageNet demonstrate that ENGM and its variants consistently improve the performance of a wide range of AT methods. Furthermore, ENGM alleviates major shortcomings of AT including robust overfitting and high sensitivity to hyperparameter settings.

1 Introduction

Susceptibility of deep neural networks (DNNs) to manipulated inputs has raised critical concerns regarding their deployment in security-sensitive applications [4, 21, 24]. The worst-case manipulation can be characterized by *adversarial examples*: carefully crafted input examples that can easily alter the model prediction while remaining benign to the human perception [37, 15]. A principal approach to formalize the imperceptibility is to bound the perturbation using ℓ_p-norm. Hence, the problem of finding a model robust to adversarial manipulation reduces to finding

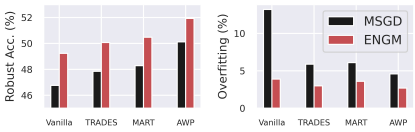

Fig. 1: Replacing MSGD with ENGM for outer optimization in AT results in consistent improvement of robust accuracy and generalization.

the one that generalizes well merely on the bounded neighborhood of the input example. Although this task seems effortless for humans, achieving such invariance is notoriously difficult for DNNs. The reason for this behavior has not been fully understood yet, but several factors have shown to be influential, including the high cardinality of the data space and non-zero test error of the classifier on noisy inputs [14, 7].

One of the most effective methods (defenses) to alleviate adversarial susceptibility is adversarial training (AT) which improves the robustness by training the model on the worst-case loss [15, 25]. Given the deep model $F_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$ and the surrogate loss function for the empirical adversarial risk L, the training objective of AT is defined as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[L^*\big(\mathbf{x},y;\boldsymbol{\theta}\big)\Big], \tag{1a}$$

$$L^*\big(\mathbf{x},y;\boldsymbol{\theta}\big) = \max_{||\mathbf{x}-\mathbf{x}'||_p\leq\epsilon} \; L\big(F_{\boldsymbol{\theta}}(\mathbf{x}'),y\big), \tag{1b}$$

where the input example $\mathbf{x}$ and the corresponding label y are a sample from the data distribution $\mathcal{D}$, $\mathbf{x}'$ is the adversarial equivalent of $\mathbf{x}$, and ϵ is the maximum ℓ_p-norm magnitude of the perturbation. Concretely, adversarial training consists of two simultaneous optimizations, referred to as the inner and outer optimizations. The inner optimization (Equation 1b) finds the worst-case adversarial example, and the outer optimization (Equation 1a) minimizes the empirical adversarial risk over the network parameters, $\boldsymbol{\theta}$.

Numerous efforts have been devoted to analyzing different aspects of AT, such as the inner optimization [25, 48, 36, 11, 10], adversarial objective [47, 40, 30, 12], computational cost [35, 50, 41], and evaluation methods [3, 26, 1, 13, 8]. Recent studies on the topic have revealed two major shortcomings of AT which contradicts common observations on NT. First, AT severely induces overfitting [34, 6], referred to as *robust overfitting*, whereas in NT overfitting is known to be less prominent especially in over-parameterized models [46, 28, 2]. Second, AT is highly sensitive to hyperparameter setting, *e.g.,* a slight change in the weight decay can deteriorate the robust performance [16, 29].

The majority of the previous works on AT have analyzed the inner optimization and its properties. However, the potential impact of outer optimization on the performance and shortcomings of AT has been critically overlooked. Furthermore, the success of the two recent state-of-the-art (SOTA) approaches of AT which indirectly affect the outer optimization by weight perturbations [42] or weight smoothing [6] advocates for further investigation on outer optimization. Based on these observations, we raise a fundamental question regarding outer optimization in AT and attempt to address it in this work:

Is the conventional MSGD, developed for non-convex optimization in NT, a proper choice for the outer optimization in AT? **If not**, *what modifications are required to make it suitable for the AT setup?*

To answer the first question, we empirically evaluate and compare two statistical parameters of gradients, namely expected norm and expected variance,

in NT and AT. Both these parameters are known to be major determinants of the performance of MSGD in NT [18, 23, 49]. We find that they are notably higher in AT compared to NT. Furthermore, after decaying the learning rate in NT, both the gradient norm and variance deteriorate suggesting convergence to local minima. However, in AT, they escalate after the learning rate decay. These observations highlight substantial disparities between the characteristics of the gradients in AT and NT. Consequently, we argue that MSGD, developed essentially for NT, is not the most proper choice for outer optimization in AT since it is not designed to be robust against high gradient norm and variance.

Motivated by these observations, the current work attempts to develop an optimization method that is more suitable for AT, *i.e.,* less sensitive to the gradient norm and variance. The contributions of the paper are as follows:

- We investigate the effect of AT on gradient properties and provide empirical evidence that AT induces higher gradient norm and variance. We argue that this hinders the optimization since the convergence rate of MSGD is highly dependent on the variance of the gradients.
- We propose an optimization method tailored specifically for AT, termed ENGM, whose convergence rate is independent of the gradient variance.
- We empirically analyze the norm of gradients and provide insightful observations regarding their correlation in DNNs. Harnessing this, we develop a fast approximation to ENGM that significantly alleviates its computational complexity.
- Through extensive evaluations and ablation studies, we demonstrate that the proposed optimization technique consistently improves the performance and generalization of the SOTA AT methods.

2 Analyzing Outer Optimization in AT

We first investigate the disparities between the properties of gradients in AT and NT in Section 2.2. Then in Section 2.3, we draw connections between the observed disparities and poor performance of MSGD in AT by reviewing the previous theoretical analysis on the convergence of MSGD. In Section 2.4, we describe our proposed optimization technique whose convergence rate is more favorable for AT. Later in Section 2.5, we present an interesting observation that enables us to approximate a fast version of the proposed optimization technique.

2.1 Notations

Throughout the paper, we denote scalars, vectors, functions, and sets using lower case, lower case bold face, upper case, and upper case calligraphic symbols, respectively. We use notation $||\cdot||_p$ for the ℓ_p-norm and drop the subscript for $p = 2$. We employ the commonly used cross-entropy loss as the measure of empirical risk and denote the loss on i^{th} example, $L(F_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$, as L_i for the sake of brevity.

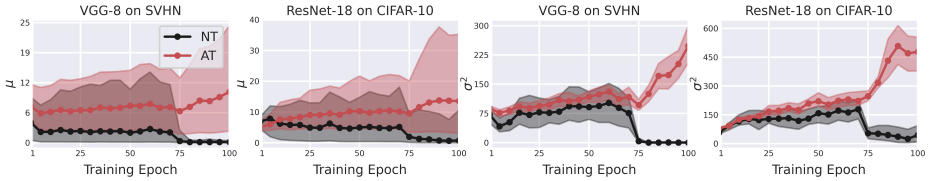

Fig. 2: Expected norm (μ) and variance (σ^2) of gradients during NT and AT. Learning rate is decayed from 10^{-1} to 10^{-2} at epoch 75. Note that the norm and variance in AT is higher than NT and escalates after learning rate decay.

2.2 Comparison of Gradient Properties

We experiment to analyze two statistical parameters of gradients which are major determinants of the performance of MSGD. The first parameter is the expected norm of gradients $\mu := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\big[||\nabla_{\boldsymbol{\theta}} L\big(F_{\boldsymbol{\theta}}(\hat{\mathbf{x}}), y\big)||\big]$, where $\hat{\mathbf{x}}$ is the natural example in NT and the adversarial example in AT. Change in the expected norm directly affects the learning rate, the most important hyperparameter in NT [18, 23]. The second parameter is the upper bound for the variance of gradients, and is defined as:

$$\sigma^2 := \sup_{\boldsymbol{\theta}} \;\; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\Big|\Big|\nabla_{\boldsymbol{\theta}} L\big(F_{\boldsymbol{\theta}}(\hat{\mathbf{x}}), y\big) - \bar{\mathbf{g}}\Big|\Big|^2\Big], \tag{2}$$

where $\bar{\mathbf{g}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\big[\nabla_{\boldsymbol{\theta}} L\big(F_{\boldsymbol{\theta}}(\hat{\mathbf{x}}), y\big)\big]$. It is shown that the convergence of MSGD is $O(\sigma^2)$ [44]. We roughly estimate both parameters during the training of ResNet-18 and VGG-8 on CIFAR-10 and SVHN datasets, respectively. Inner optimization in AT follows the standard setup, *i.e.,* 10 steps of ℓ_∞-norm PGD with $\epsilon = 8/255$ and step size $\epsilon/4$.

Figure 2 plots μ and σ^2 during 100 training epochs with the learning rate decay from 10^{-1} to 10^{-2} at epoch 75. We observe that the expected norm and variance of gradients is notably higher in AT. After learning rate decay, both parameters decrease significantly in NT suggesting the convergence to local minima. However in AT, the expected norm grows and the variance increases drastically. These findings highlight substantial disparities between the characteristics of the gradients in AT and NT. In the next section, we theoretically analyze how these differences can affect the convergence of MSGD.

2.3 Revisiting Stochastic Gradient Descent

In this part, we analyze the functionally and convergence of MSGD to identify modifications that improves its suitability for the AT setup. The update rule of MSGD at iteration t is as follows:

$$\mathbf{v}_{t+1} = \beta \mathbf{v}_t + \frac{1}{|\mathcal{I}_t|} \sum_{i\in\mathcal{I}_t} \nabla_{\boldsymbol{\theta}} L_i, \tag{3a}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{v}_{t+1}, \tag{3b}$$

Algorithm 1 Fast ENGM

Initialize $\tau > 0$, $\beta_\gamma \in [0, 1)$, $\alpha > 0$, $\gamma_0 = 0$, $\gamma_1 = 1$, Boolean parameter *Naive*.
for $t = 0 \dots t_1 - 1$ **do**
  Compute $L_i, \quad \forall i \in \mathcal{I}_t$; ▷ inner optimization
  Compute $\mathcal{G}_{\mathbf{x},t} = \{\nabla_{\mathbf{x}} L_i : i \in \mathcal{I}_t\}$; ▷ backprop. $\times 1$
  if mode$(t, \tau) = 0$ and *Naive* = False **then**
    Compute $\mathcal{G}_{\boldsymbol{\theta},t} = \{\nabla_{\boldsymbol{\theta}} L_i : i \in \mathcal{I}_t\}$; ▷ backprop. $\times n$ every τ iterations
    $\gamma_1', \gamma_0' = \text{LinearRegression}(\mathcal{G}_{\mathbf{x},t}, \mathcal{G}_{\boldsymbol{\theta},t})$ ▷ estimate slope and intercept
    $\gamma_0 \leftarrow \beta_\gamma \gamma_0 + (1 - \beta_\gamma)\gamma_0'$, and $\gamma_1 \leftarrow \beta_\gamma \gamma_1 + (1 - \beta_\gamma)\gamma_1'$;
  end if
  $\hat{w}_i \leftarrow \max(\frac{\alpha}{||\gamma_1 \nabla_{\mathbf{x}} L_i + \gamma_0||}, 1), \quad \forall i \in \mathcal{I}_t$;
  Update $\boldsymbol{\theta}$ with MSGD on the reweighted loss $\frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \hat{w}_i L_i$ ▷ backpropagation$\times 1$
end for

where η is the learning rate, $\mathbf{v}_{t+1}$ is the Polyak's momentum with the corresponding modulus β [33], $\mathcal{I}_t$ is the randomly selected set of indices for the mini-batch with size $|\mathcal{I}_t|$, and L_i is the objective for optimization computed on the i^{th} example. Assuming F has bounded variance of gradients according to Equation 2, and is smooth in $\boldsymbol{\theta}$, *i.e.,* $F_{\boldsymbol{\theta}_1}(\mathbf{x}) \leq F_{\boldsymbol{\theta}_2}(\mathbf{x}) + \langle \nabla_{\boldsymbol{\theta}} F_{\boldsymbol{\theta}_1}(\mathbf{x}), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 \rangle + \frac{c}{2}||\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1||^2$, Yu *et al.* [43, 44] have shown that the convergence rate of MSGD for non-convex optimization in DNNs is $O(\sigma^2)$. Hence, MSGD is not suitable for tasks with high gradient variance. Intuitively, higher variance implies that the gradients are not aligned with the average gradients which are being used to update the model parameters. This hinders the optimization process since the update is merely favorable for a portion of examples in the mini-batch.

One alternative to MSGD that is less sensitive to the variance of the gradients is stochastic normalized gradient descent with momentum (SNGM) [49]. SNGM is shown to provide better generalization for training with large batch size, *i.e.,* another cause of high gradient variance. Concretely, SNGM modifies Equation 3a as:

$$\mathbf{v}_{t+1} = \beta \mathbf{v}_t + \frac{\sum_{i \in \mathcal{I}_t} \nabla_{\boldsymbol{\theta}} L_i}{||\sum_{i \in \mathcal{I}_t} \nabla_{\boldsymbol{\theta}} L_i||}, \tag{4}$$

which limits the gradient norm by normalizing the magnitude of mini-batch gradients and considers only the direction of the average gradient. Zhao *et al.* [49] have shown that the convergence of SNGM is $O(\sigma)$, and therefore, is more suitable for tasks with induced gradient fluctuations. We also observe in Section 3.1 that SNGM improves the generalization in AT. This suggests that reducing the sensitivity of the optimizer to the gradient variance has a direct impact on the generalization and performance of the task with adversarial gradients.

2.4 Example-normalized Gradient Descent with Momentum

Although SNGM is less sensitive than MSGD to the variance of gradients, it does not impose any constraint on the variance. Hence, the variance can still

become large and impede the optimization. To address this, we introduce a transformation on gradient vectors that bounds the variance of the gradients in the mini-batch and makes the convergence rate of the optimizer independent of the variance.

Theorem 1. *For any arbitrary distribution $\mathcal{P}$ of random vectors, applying the transformation $T(\mathbf{a}) = \min(\frac{\alpha}{||\mathbf{a}||}, 1)\mathbf{a}$ with $\alpha > 0$ bounds the variance of vectors to $4\alpha^2$.*
(Proof is provided in Section 1 of Supp. material.)

We use the transformation in Theorem 1 to bound the variance of the gradients. To this aim, we rewrite Equation 3a as:

$$\mathbf{v}_{t+1} = \beta \mathbf{v}_t + \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} w_i \nabla_{\boldsymbol{\theta}} L_i, \tag{5a}$$

$$w_i = \min\big(\frac{\alpha}{||\nabla_{\boldsymbol{\theta}} L_i||}, 1\big), \tag{5b}$$

where w_i is the normalizing coefficient for $\nabla_{\boldsymbol{\theta}} L_i$, and α is the maximum allowed norm of gradients. This update rule limits the maximum norm of the gradients on each input example to α. Hence, it prevents high magnitude gradients from dominating the updating direction and magnitude in the mini-batch. It might be noted that α scales with the square root of the model size, and larger models require higher values of α. We refer to this approach as **e**xample-**n**ormalized stochastic **g**radient descent with **m**omentum (ENGM). ENGM recovers MSGD when $\alpha \gg 1$. The convergence properties of ENGM is analyzed in Theorem 2.

Theorem 2. *Let $A(\boldsymbol{\theta})$ be the average loss over all examples in the dataset, and assume that it is smooth in $\boldsymbol{\theta}$. For any $\alpha > 0$ and total iterations of t_1, optimizing $A(\boldsymbol{\theta})$ using ENGM (Equation 5) has the convergence of $O(\alpha)$. (Proof is provided in Section 1 of Supp. material.)*

Theorem 2 shows that the convergence rate of ENGM is $O(\alpha)$ and is independent of the variance of gradients. Hence, it is suitable for optimizing objectives with high gradient variance. Later in Section 3.1, we empirically validate this and show that the enhanced regularization of ENGM provides better optimization compared to SNGM and MSGD for AT. Despite the intrinsic merits of ENGM, it is computationally expensive since evaluating each w_i requires a dedicated backpropagation and cannot be implemented in parallel. In particular, Equation 5 requires $|\mathcal{I}_t|$ backpropagation for each mini-batch. In the next section, we present an empirical observation on the gradients of DNNs that enables us to estimate w_i and consequently Equation 5 using merely one additional backpropagation.

2.5 Accelerating ENGM via Gradient Norm Approximation

During our evaluations, we observe an interesting phenomenon that enables us to develop a fast approximation to ENGM. Particularly, we observe that the

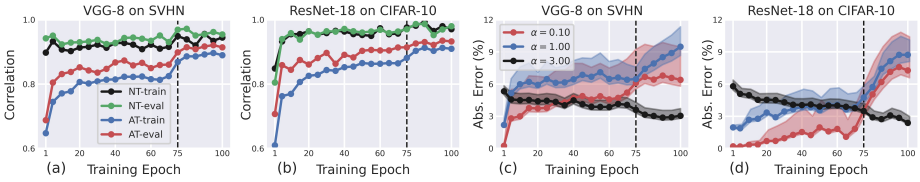

Fig. 3: (a,b): Characterizing the linear correlation between $||\nabla_{x_i} L_i||$ and $||\nabla_{\boldsymbol{\theta}} L_i||$ using Pearson correlation coefficient. (c, d): The absolute value of error (%) for estimating w_i using Equation 7. Dashed black line denotes the learning rate decay from 10^{-1} to 10^{-2}.

Method	MSGD	MSGD+GNC	SNGM	F-ENGM	N-ENGM	A-ENGM	ENGM
Ex. time (sec./iter)	0.60	0.61	0.63	5.05	0.75	0.83	5.06

Table 1: Execution time of the outer optimization methods. Experiments are conducted on an NVIDIA Titan-RTX GPU.

norm of gradients w.r.t. the network parameters, $||\nabla_{\boldsymbol{\theta}} L_i||$, is linearly correlated with the norm of the gradients w.r.t. the input example, $||\nabla_{\mathbf{x}_i} L_i||$. To illustrate this phenomenon, we track both gradient norms on $1,000$ training examples during NT and AT using VGG-8 on SVHN and ResNet-18 on CIFAR-10. We compute Pearson correlation coefficient to measure the correlation between the two norms. Figures 3a and 3b show the correlation coefficient during AT and NT with the model in the evaluation and training modes. We can see that there is a significant correlation between the two norms in DNNs which becomes stronger as the training proceeds. The correlation exists in both the training and evaluation modes of the model, and is slightly affected by the update in the statistics of the batch normalization modules.

Harnessing this phenomenon, we can estimate the norm of gradient w.r.t. the network parameters (computationally expensive) using the norm of gradients w.r.t. the inputs (computationally cheap) with a linear approximation as:

$$||\nabla_{\boldsymbol{\theta}} L_i|| \approx \gamma_1 ||\nabla_{\mathbf{x}_i} L_i|| + \gamma_0, \tag{6}$$

where γ_0 and γ_1 are coefficients for the slope and intercept of the linear estimation, respectively. Employing this estimation, we can approximate the functionality of ENGM by a simple modification of the loss on the i^{th} input example, L_i, and keeping the popular MSGD as the optimizer. This provides two benefits. First, there is no need to implement a new optimizer enhancing the applicability of the method. Second, the reweighting significantly reduces the computational cost of ENGM. To this aim, we use the estimated value for the norm of the gradients w.r.t. the input to normalize the gradients w.r.t. the network parameters indirectly by assigning a weight to the loss function computed on $\mathbf{x}_i$ as $\hat{L}_i := \hat{w}_i L_i$, where:

$$\hat{w}_i := \max\left(\frac{\alpha}{||\gamma_1 \nabla_{\mathbf{x}} L_i + \gamma_0||}, 1\right). \tag{7}$$

Here, optimizing the total loss $\frac{1}{|\mathcal{I}_t|}\sum_{i\in\mathcal{I}_t}\hat{L}_i$ using MSGD will approximately recover the functionality of ENGM on $\frac{1}{|\mathcal{I}_t|}\sum_{i\in\mathcal{I}_t}L_i$. To analyze the accuracy of estimating $\hat{w}_i$, we measure the average absolute value of the error during the training of the both models in AT and for three different values of $\alpha \in \{0.1, 1.0, 3.0\}$. Figures 3c and 3d visualize the error on two different datasets and network architectures. We observe that the maximum absolute value of error is less than 10% which advocates for the accuracy of estimating $\hat{w}_i$. For large values of α the error decreases during the training, while for small values of α the error increases. This points to a trade-off between the estimation error across the training process. It might be noted that the error is computed solely for AT since based on the evaluations in Figures 3a and 3d the correlation is stronger in NT.

Unlike $\nabla_{\boldsymbol{\theta}}L_i$, $\nabla_{\mathbf{x}}L_i$ can be computed in parallel for a batch of data using a single backpropagation. We consider two approaches for estimating γ_0 and γ_1 which result in two variations of ENGM. In the first approach, referred to as Approximated ENGM (A-ENGM), we evaluate $\nabla_{\boldsymbol{\theta}}L_i$ for a single mini-batch every τ iterations and use moving average to update the latest estimate. Then for the intermediate iterations, we use the estimate values of γ to approximate the norm of gradients using Equation 6. In comparison, A-ENGM reduces the required number of additional backpropagations from $|\mathcal{I}_t|$ (for ENGM) to $1 + |\mathcal{I}_t|/\tau$. In practice, we observe that the interval, τ, for estimating γ values can be conveniently large as investigated in Section 3.4. Furthermore, we consider a second approach in which we simply set $\gamma_0 = 0$ and merge γ_1 into α. We refer to this approach as Naive ENGM (N-ENGM) which solely requires a single additional backpropagation.

3 Experiments and Analysis

We evaluate ENGM on three datasets of CIFAR-10, CIFAR-100 [20], and Tiny-ImageNet [22]. Following the benchmark experimental setup for AT [42, 47, 40, 9], we conduct ablation studies and exploratory evaluations on ResNet-18 with 64 initial channels, originally developed for ImageNet. For SOTA evaluation, we use Wide ResNet-34 with depth factor 10 (WRN-34-10) [45].

Training Setup. Except for evaluations involving ENGM, all the models are trained using MSGD with momentum 0.9, weight decay 5×10^{-4} [42, 29, 16], batch size equal to 128, and initial learning rate of 0.1. The learning rate is decayed by 0.1 at epochs 75, 90, and the total number of epochs is set to 120 unless otherwise noted. The standard data augmentation including random crop with padding size 4 and horizontal flip is applied for all datasets. All input images are normalized to $[0, 1]$. Based on ablation studies in Section 3.4, we set α for ENGM, A-ENGM, and N-ENGM to 5, 5, and 0.5, respectively. The momentum for A-ENGM is set to 0.7 based on empirical evaluations. PGD with 10 steps (PGD^{10}), $\epsilon = 8/255$, and step size $2/255$ is used as the attack to maximize the adversarial loss in ℓ_∞-norm ball. As suggested by Rice *et al.* [34], during the training we select the model with the highest robust accuracy against PGD^{20}

Optim.	Accuracy (%)				Overfit.
Method	Natural	Best	Last	AA	(%)
MSGD	**84.70**	50.87	44.15	46.77	13.2
MGNC	83.98	51.88	46.62	47.59	10.1
SNGM	83.73	51.95	46.80	47.75	9.9
F-ENGM	82.91	50.05	44.04	46.54	12.0
N-ENGM	84.36	52.19	48.79	48.06	6.5
A-ENGM	83.61	52.46	49.75	48.46	5.1
ENGM	83.44	**53.04**	**52.76**	**49.24**	**3.9**

Table 2: Comparison of ENGM with MSGD for outer optimization in AT (§3.1). 'Best' and 'Last' refer to the accuracy against PGD^{20} using the best and last checkpoints, respectively.

with $\epsilon = 8/255$ and step size $8/(255 \times 10)$ on a validation set of size $1,000$ as the best model. Only for PGD^{20}, we use margin loss instead of cross-entropy due to its better performance in evaluating the robustness of the model [39].

Evaluation Setup. We evaluate the model against two major attacks. First is the same PGD^{20} used in the training to find the best model. For a more rigorous evaluation of the robust performance, we follow the setup of the recent SOTA defense methods [47, 42, 48, 9, 19, 5, 36, 29] and use the benchmark adversarial robustness measure of AutoAttack (AA) [8]. AA has shown consistent superiority over other white box attacks such as JSMA [31], MIM [13], and CW [3][1]. Both attacks in evaluations are applied on the test set, separated from the validation set. Maximum norm of perturbation, ϵ, is set to 8/255 and 128/255 for ℓ_∞-norm and ℓ_2-norm threat models. In addition to the robust accuracy, the robust overfitting of the model is computed as the difference between the best and the last robust accuracies (PGD^{20}) normalized over the best robust accuracy. All results are the average of three independent runs.

3.1 Comparison of Optimization methods

In this section, we evaluate and compare the proposed method with other possible choices for outer optimization in AT. As the fist baseline, we employ the conventional MSGD which is the optimizer in all of the previous AT methods. A popular and well-known trick to bound the gradient norm especially in recurrent neural networks is Gradient Norm Clipping (GNC) [17, 32]. GNC clips the gradient norm when it is greater than a threshold. This threshold is similar to α in our method. However, instead of bounding the gradient norm on each individual input example, GNC bounds the norm of the average gradients of the mini-batch. We consider the combination of MSGD with GNC as our second baseline and refer to it as MGNC. The clipping threshold α for MSGD+GNC is set to 25 based on empirical evaluations. SNGM, discussed in Section 2.3, is used as the third baseline. For our method, we compare the original ENGM with

[1] *github.com/fra31/auto-attack.*

its accelerated versions, *i.e.,* A-ENGM and N-ENGM. The coefficients α and τ for our methods are set to the best-performing values from Section 3.4. As an additional baseline, we develop another version of ENGM in which instead of bounding the norm of gradients, we normalize them to the constant value α, *i.e.,* modifying Equation 5a to: $\mathbf{v}_{t+1} = \beta \mathbf{v}_t + \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \frac{\nabla_{\boldsymbol{\theta}} L_i}{||\nabla_{\boldsymbol{\theta}} L_i||}$. We refer to this method as Fixed ENGM (F-ENGM).

Table 2 presents the results for these comparisons. We can see that the simple GNC enhances robust accuracy providing the same performance as SNGM. These improvements caused by simple modifications further confirms the negative effect of high gradient norm and variance on outer optimization in AT. ENGM consistently improves the robust accuracy over baselines. In addition, robust overfitting in ENGM is significantly lower than other baselines. This suggests that a major cause of robust overfitting in AT is the high fluctuation of gradients and the incompetence of MSGD in addressing it. The learning curves (robust test accuracy) for different optimization methods are depicted in Figure 5h. We observe that after the learning rate decay, the robust performance of ENGM and its variants does not deteriorate which confirms that they alleviate robust overfitting. The best natural accuracy is provided by MSGD supporting the commonly observed trade-off between the natural and robust accuracies [38, 47]. Table 1 presents the execution time for the optimization methods. The execution time of ENGM is roughly 8.5× longer than MSGD. However, A-ENGM and N-ENGM achieve notable speed-up and robust performance. As expected, the performance of A-ENGM is between N-ENGM (lower-bound) and ENGM (upper-bound) and is controlled by the estimation interval τ. Hence, we use N-ENGM and ENGM for the major evaluations to clearly compare the two performance bounds.

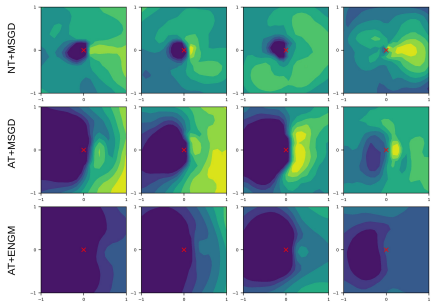

Fig. 4: Visualization of the loss landscape on four examples from CIFAR-10 (§3.2). The cross mark denotes the input example. Loss level sets are equalized on each column.

3.2 Combination with Benchmark AT Methods

In the this section, we incorporate the proposed optimization approaches into the benchmark AT methods including the vanilla method [25], TRADES [47], MART [40], and AWP [42]. Here, AWP represents the weight perturbation method applied on top of TRADES. The coefficient for the self-distillation loss in TRADES and MART is set to 6, and the maximum magnitude of weight perturbation for AWP is set to 5×10^{-3}. The rest of the training setups are set to the best setup reported by the original papers. However, the total training epochs for all methods is set to 200 (learning rate decays by 0.1 at epochs 100 and 150) for the sake of consistency.

	AT Method	Optim. Method	Accuracy (%) Natural	PGD^{20}	AA	Overfit. (%)
CIFAR-10	Vanilla	MSGD	**84.70**	50.87	46.77	13.2
		ENGM	83.44	53.04	49.24	3.9
	TRADES	MSGD	82.40	50.94	47.85	5.9
		ENGM	82.33	53.46	50.07	3.0
	MART	MSGD	83.68	51.05	48.29	6.1
		ENGM	83.03	53.56	50.48	4.6
	AWP	MSGD	82.98	52.55	50.12	4.6
		ENGM	83.10	**54.07**	**51.93**	**2.7**
CIFAR-100	Vanilla	MSGD	**57.75**	26.11	24.45	20.9
		ENGM	56.91	28.43	26.60	7.4
	TRADES	MSGD	56.00	29.04	26.93	10.6
		ENGM	55.65	30.68	29.20	7.1
	MART	MSGD	56.52	29.41	27.18	11.8
		ENGM	56.20	30.89	29.30	8.6
	AWP	MSGD	56.22	30.36	28.43	7.3
		ENGM	56.82	**31.24**	**30.46**	**6.3**
Tiny-ImageNet	Vanilla	MSGD	35.71	7.47	6.92	26.37
		ENGM	29.78	11.29	8.54	10.10
	TRADES	MSGD	**37.26**	14.13	10.95	14.79
		ENGM	36.30	16.88	12.65	8.74
	MART	MSGD	37.06	13.79	10.08	15.94
		ENGM	36.53	16.90	12.99	8.20
	AWP	MSGD	36.13	16.29	13.09	10.67
		ENGM	36.81	**19.14**	**16.02**	**7.97**

Table 3: Comparison of MSGD and ENGM on different AT methods (§3.2). Note that ENGM consistently outperforms MSGD.

Table 3 presents the results for ℓ_∞-norm threat model. For results on ℓ_2-norm threat model please refer to Section 2 in Supp. material. We observe that ENGM consistently outperforms MSGD on robust performance. The average improvement in robustness against AA is 2.15% and 1.16% in ℓ_∞-norm and ℓ_2-norm, respectively. This suggests that the *amount* of perturbation in AT affects the convergence of the outer optimization. Consider the ℓ_2-norm as the unified metric, the amount of noise in ℓ_∞-norm threat model is roughly $3\times$ the norm of noise in the counterpart threat model. Combining these results with the evaluations in Figure 2 advocates that the improvement offered by ENGM over MSGD depends on the norm of perturbation. This observation is further investigated in Section 3.4.

AWP is previously shown to alleviate robust overfitting [42]. Interestingly, we find that TRADES and MART also reduce the robust overfitting independent of the optimization method. This suggests that the AT method can affect the robust overfitting. ENGM results in the lowest overfitting and consistently surpasses MSGD. On vanilla AT, replacing MSGD with ENGM results in 9.3%, 13.5%, and 16.2% reduction of overfitting on CIFAR-10, CIFAR-100, and TinyImageNet, respectively. These results advocate that, in addition to the AT method, the outer optimization method also affects the overfitting and limiting the sensitivity of the optimization method to the variance of the gradients can alleviate the robust overfitting.

As the last evaluation in this part, we visualize the loss landscape on networks optimized by MSGD and ENGM in Figure 4. This figure plots the loss values for the space spanned by the adversarial perturbation (PGD^{20}) and random

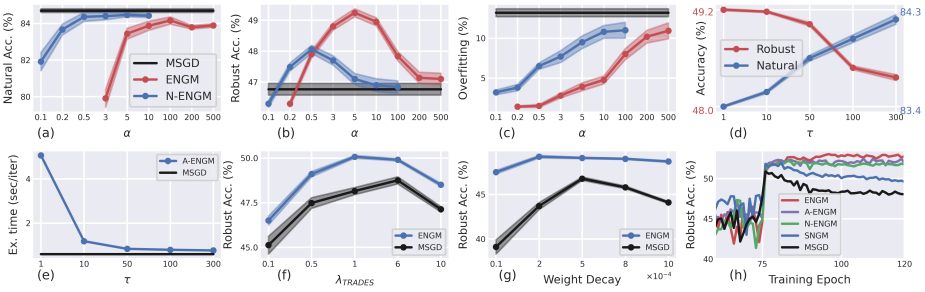

Fig. 5: (a-g): Ablation studies on α, τ, λ_{TRADES}, and weight decay (§3.4). Note that α of ENGM scales to that of N-ENGM with $1/\gamma_1$. Robust accuracy is measured using AutoAttack [8]. (h): learning curves (robust test accuracy) for AT with different outer optimization methods. Results on last 60 epochs are plotted for better visualization of the robust overfitting. Robust accuracy is measured using PGD^{20}.

noise, orthogonalized to the perturbation via Gram-Schmidt. We can see that ENGM results in a smoother loss landscape, known as an empirical evidence of the robustness [27]. This qualitative analysis further validates the effectiveness of ENGM for outer optimization in AT.

3.3 Comparison with SOTA

Here, we evaluate ENGM in the benchmark of AT, *i.e.,* WRN-34-10 on CIFAR-10 dataset [47, 42, 48, 9, 19, 5, 36]. For training using ENGM, we set $\alpha = 10.4$ which is obtained by scaling the best α for ResNet-18 with the factor of 2.08, square root of the ratio of the total parameters of the two models ($48.2M$ for WRN-34-10 vs. $11.1M$ for ResNet-18). To achieve SOTA performance, we consider AWP as the AT scheme. We train the model for 200 epochs with learning rate decay by 0.1 at epochs 100 and 150. The rest of the setting is the same as our previous evaluations. Table 4 presents the results for this experiment. AWP combined with ENGM and N-ENGM surpasses the previous SOTA by 1.28% and 0.94%, respectively. This validates the effectiveness of ENGM on large models. We also find that ENGM results in higher natural accuracy on AWP. This suggests that although AWP indirectly improves the outer optimization, its impact is orthogonal to ENGM.

3.4 Ablation Studies

We conduct ablation studies to investigate the impact of hyperparameters on the performance of ENGM and its two variants using ResNet-18 on CIFAR-10.
Impact of α: We measure the natural accuracy, robust accuracy (AA), and overfitting versus α. We conduct this experiment on ENGM/N-ENGM since they upper/lower bound the performance of A-ENGM.

Method	Optim.	Nat. Acc. (%)	AA (%)
ATES [36]	MSGD	86.84	50.72
BS [5]	MSGD	85.32	51.12
LBGAT [9]	MSGD	**88.22**	52.86
TRADES [47]	MSGD	84.92	53.08
MART [40]	MSGD	84.98	53.17
BERM [19]	MSGD	83.48	53.34
FAT [48]	MSGD	84.52	53.51
AWP [42]	MSGD	85.36	56.17
AWP	N-ENGM	85.40	57.11
AWP	ENGM	86.12	**57.45**

Table 4: Comparison of the benchmark robustness on WRN.

Figures 5a, 5b, and 5c present the results for these evaluations. As expected, for large values of α all three values converge to that obtained by MSGD. Small values of α can be interpreted as training with a very small learning rate causing both the natural and robust accuracies to drop. Interestingly, we observe that the overfitting decreases significantly for small values of α. This confirms that the high variance of gradients in AT negatively affects the functionality of MSGD, *i.e.,* ENGM with large α. We find that ENGM and N-ENGM achieve their optimal performance on ResNet-18 at α equal to 5 and 0.5, respectively. We select these as the optimal values for training the models in other experiments. Note that the optimal value of α is expected to be the same for ENGM and A-ENGM but different for N-ENGM. This is because the formulation of ENGM and A-ENGM is the same except that A-ENGM estimates the norm of gradients every τ iterations, and setting $\tau = 1$ recovers the exact ENGM. However, in N-ENGM, α is scaled by $1/\gamma_1$ according to the discussion in Section 2.5. The optimal α is scaled for other networks based on their capacity.

Impact of τ: We conduct experiments to evaluate the role of τ in AT setup with A-ENGM ($\alpha = 5$) as the optimizer and $\tau \in \{1, 10, 50, 100, 300\}$. It might be noted that each epoch in CIFAR-10 consists of 390 mini-batches of size 128. Hence, $\tau = 300$ is roughly equivalent to estimating the correlation at the end of each epoch. Figures 5d and 5e present the results for these evaluations. As expected, for small and large values of τ A-ENGM converges to ENGM and N-ENGM, respectively. For $\tau = 50$, obtained robustness is roughly 85% of the robustness obtained by ENGM while the training time is significantly lower (0.83 vs. 5.06) because the extra gradient computation is being performed every 50 iterations. Furthermore, we can see that τ controls the trade-off between the natural and robust accuracies.

Perturbation norm: As an initial exploration in this paper, we observed that AT induces higher gradient norm and variance. We also noticed in Section 3.2 that ENGM seems to outperform MSGD with a larger margin when the magnitude of perturbations is higher. Here, we further analyze the impact of the magnitude of perturbations on the gradient norm and variance induced by AT.

This allows us to identify the extent of suitability of MSGD and ENGM for NT and AT. We train models in AT setup with ℓ_∞-norm threat model and varied size of perturbation, $\epsilon \in \{0, 2/255, 4/255, 6/255, 8/255, 10/255\}$. Both MSDG and ENGM are utilized for the outer optimization in these evaluations. We measure the average norm and variance of gradients across all training epochs. For a fair comparison, we compute the expected distance to the closest decision boundary as the unified robustness measure: $\rho := E_{\mathbf{x}}[||\mathbf{x} - \mathbf{x}^*||]$, where $\mathbf{x}^*$ is the closest adversary to x computed using DeepFool [26].

Table 5 presents the results for this experiment. In NT (AT with $\epsilon = 0$), MSGD provides slightly better performance than ENGM. This is because in NT the norm and variance of gradients are naturally limited. As the ϵ increases, the expected norm and variance of the gradients also increase. This confirms our initial observation that AT induces higher gradient norm and variance. Consequently as expected, we find that in AT with larger magnitude of perturbations ENGM works better than MSGD.

	Magnitude of Perturbation, ϵ ($\times \frac{1}{255}$)					
	0	2	4	6	8	10
μ	4.25	5.10	6.09	7.73	10.04	14.21
σ^2	118.1	118.7	121.8	141.7	185.2	253.5
ρ_{MSGD}	0.33	0.41	0.57	0.93	1.15	1.24
ρ_{ENGM}	0.30	0.42	0.61	1.08	1.35	1.49

Table 5: Analyzing the impact of the perturbation magnitude on gradient properties and final robustness obtained by MSGD and ENGM (§3.2). AT with $\epsilon = 0$ is equivalent to NT.

Sensitivity to hyperparameters: One intriguing shortcoming of AT is sensitivity to hyperparameter setting. Several works have shown that a slight change in the modulus of the ℓ_2-norm regularization, *i.e.,* weight decay, results in drastic changes in robust performance [29, 16]. Here, we analyze the sensitivity of the proposed optimization method and compare it with that of MSGD. Figure 5g presents the results for this evaluation. We observe that ENGM exhibits significantly less sensitivity to changes in weight decay compared to MSGD. We hypothesis that high weight decay helps MSGD to prevent the bias from input examples with high gradient magnitude. ENGM achieves this goal by explicitly limiting the gradient magnitudes, and thus, is less sensitive to weight decay. We believe this phenomenon calls for more in depth analysis and defer it to future studies.

4 Conclusion

In this paper, we studied the role of outer optimization in AT. We empirically observed that AT induces higher gradient norm and variance which degrades the performance of the conventional optimizer, *i.e.,* MSGD. To address this issue, we developed an optimization method robust to the variance of gradients called ENGM. We provided two approximations to ENGM with significantly reduced computational complexity. Our evaluations validated the effectiveness of ENGM and its fast variants in AT setup. We also observed that ENGM alleviates shortcomings of AT including the robust overfitting and sensitivity to hyperparameters.

References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International conference on machine learning. pp. 274–283. PMLR (2018)
2. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences **116**(32), 15849–15854 (2019)
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
4. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE international conference on computer vision. pp. 2722–2730 (2015)
5. Chen, J., Cheng, Y., Gan, Z., Gu, Q., Liu, J.: Efficient robust training via backward smoothing. arXiv preprint arXiv:2010.01278 (2020)
6. Chen, T., Zhang, Z., Liu, S., Chang, S., Wang, Z.: Robust overfitting may be mitigated by properly learned smoothening. In: International Conference on Learning Representations (2020)
7. Cohen, J., Rosenfeld, E., Kolter, Z.: Certified adversarial robustness via randomized smoothing. In: International Conference on Machine Learning. pp. 1310–1320. PMLR (2019)
8. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. pp. 2206–2216. PMLR (2020)
9. Cui, J., Liu, S., Wang, L., Jia, J.: Learnable boundary guided adversarial training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15721–15730 (2021)
10. Dabouei, A., Soleymani, S., Dawson, J., Nasrabadi, N.: Fast geometrically-perturbed adversarial faces. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1979–1988. IEEE (2019)
11. Dabouei, A., Soleymani, S., Taherkhani, F., Dawson, J., Nasrabadi, N.: Smoothfool: An efficient framework for computing smooth adversarial perturbations. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2665–2674 (2020)
12. Dabouei, A., Soleymani, S., Taherkhani, F., Dawson, J., Nasrabadi, N.M.: Exploiting joint robustness to adversarial perturbations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1122–1131 (2020)
13. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018)
14. Ford, N., Gilmer, J., Carlini, N., Cubuk, D.: Adversarial examples are a natural consequence of test error in noise. arXiv preprint arXiv:1901.10513 (2019)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
16. Gowal, S., Qin, C., Uesato, J., Mann, T., Kohli, P.: Uncovering the limits of adversarial training against norm-bounded adversarial examples. arXiv preprint arXiv:2010.03593 (2020)
17. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)

18. He, F., Liu, T., Tao, D.: Control batch size and learning rate to generalize well: Theoretical and empirical evidence. Advances in Neural Information Processing Systems **32**, 1143–1152 (2019)
19. Huang, L., Zhang, C., Zhang, H.: Self-adaptive training: beyond empirical risk minimization. Advances in Neural Information Processing Systems **33** (2020)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
21. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016)
22. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
23. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:1908.03265 (2019)
24. Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., Lu, F.: Understanding adversarial attacks on deep learning based medical image analysis systems. Pattern Recognition **110**, 107332 (2021)
25. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
26. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
27. Moosavi-Dezfooli, S.M., Fawzi, A., Uesato, J., Frossard, P.: Robustness via curvature regularization, and vice versa. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9078–9086 (2019)
28. Neyshabur, B., Bhojanapalli, S., Mcallester, D., Srebro, N.: Exploring generalization in deep learning. Advances in Neural Information Processing Systems **30**, 5947–5956 (2017)
29. Pang, T., Yang, X., Dong, Y., Su, H., Zhu, J.: Bag of tricks for adversarial training. arXiv preprint arXiv:2010.00467 (2020)
30. Pang, T., Zhang, H., He, D., Dong, Y., Su, H., Chen, W., Zhu, J., Liu, T.Y.: Adversarial training with rectified rejection. arXiv preprint arXiv:2105.14785 (2021)
31. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P). pp. 372–387. IEEE (2016)
32. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318. PMLR (2013)
33. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. Ussr computational mathematics and mathematical physics **4**(5), 1–17 (1964)
34. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning. pp. 8093–8104. PMLR (2020)
35. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
36. Sitawarin, C., Chakraborty, S., Wagner, D.: Improving adversarial robustness through progressive hardening. arXiv preprint arXiv:2003.09347 (2020)
37. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)

38. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2018)
39. Uesato, J., O'donoghue, B., Kohli, P., Oord, A.: Adversarial risk and the dangers of evaluating against weak attacks. In: International Conference on Machine Learning. pp. 5025–5034. PMLR (2018)
40. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: International Conference on Learning Representations (2019)
41. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994 (2020)
42. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. arXiv preprint arXiv:2004.05884 (2020)
43. Yan, Y., Yang, T., Li, Z., Lin, Q., Yang, Y.: A unified analysis of stochastic momentum methods for deep learning. arXiv preprint arXiv:1808.10396 (2018)
44. Yu, H., Jin, R., Yang, S.: On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In: International Conference on Machine Learning. pp. 7184–7193. PMLR (2019)
45. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
46. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization (2016). arXiv preprint arXiv:1611.03530 (2017)
47. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning. pp. 7472–7482. PMLR (2019)
48. Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., Kankanhalli, M.: Attacks which do not kill training make adversarial learning stronger. In: International Conference on Machine Learning. pp. 11278–11287. PMLR (2020)
49. Zhao, S.Y., Xie, Y.P., Li, W.J.: Stochastic normalized gradient descent with momentum for large batch training. arXiv preprint arXiv:2007.13985 (2020)
50. Zheng, H., Zhang, Z., Gu, J., Lee, H., Prakash, A.: Efficient adversarial training with transferable adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1181–1190 (2020)