On the Duality Between Sharpness-Aware Minimization and Adversarial Training

Yihao Zhang [1 *] **Hangzhou He** [1 *] **Jingyu Zhu** [1 2 *] **Huanran Chen** [3] **Yifei Wang** [4] **Zeming Wei** [1]

Abstract

Adversarial Training (AT), which adversarially perturb the input samples during training, has been acknowledged as one of the most effective defenses against adversarial attacks, yet suffers from inevitably decreased clean accuracy. Instead of perturbing the samples, Sharpness-Aware Minimization (SAM) perturbs the model weights during training to find a more flat loss landscape and improve generalization. However, as SAM is designed for better clean accuracy, its effectiveness in enhancing adversarial robustness remains unexplored. In this work, considering the duality between SAM and AT, we investigate the adversarial robustness derived from SAM. Intriguingly, we find that using SAM alone can improve adversarial robustness. To understand this unexpected property of SAM, we first provide empirical and theoretical insights into how SAM can implicitly learn more robust features, and conduct comprehensive experiments to show that SAM can improve adversarial robustness notably without sacrificing any clean accuracy, shedding light on the potential of SAM to be a substitute for AT when accuracy comes at a higher priority. Code is available at https://github.com/weizeming/SAM_AT.

1. Introduction

The existence of adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013) has raised serious safety concerns on the deployment of deep neural networks (DNNs). To defend against attacks of crafting adversarial examples, a variety of defense methods (Papernot et al., 2016; Xie et al., 2019; Cohen et al., 2019; Chen et al., 2023a) has been proposed. **Adversarial training (AT)** (Madry et al., 2017), which adds adversarial perturbations to the training samples in the training loop, has been acknowledged as one of the most effective adversarial defense paradigms (Athalye et al., 2018). Despite this success, AT suffers from an intrinsic limitation that decreases the clean accuracy, leading to a fundamental trade-off between accuracy and robustness (Tsipras et al., 2018; Zhang et al., 2019). A critical drawback of AT is that the perturbation of training samples makes the training distribution deviate from the natural data distribution.

Instead, **Sharpness-Aware Minimization (SAM)** (Foret et al., 2020) perturbs model weights yet keeps using the original samples during training. SAM is a novel training framework that regularizes the sharpness of the loss landscape by weight perturbation to improve generalization ability. So far, SAM has achieved remarkable success in modern machine learning (Bahri et al., 2021; Andriushchenko and Flammarion, 2022; Chen et al., 2023b) and many of its follow-up variants have been proposed (Kwon et al., 2021; Du et al., 2021; Chen et al., 2022). However, as SAM was originally designed for better natural generalization, its impact on adversarial robustness remains unexplored.

In this paper, considering the duality between SAM and AT, *i.e.* **perturbing samples (AT)** and **perturbing weights (SAM)**, we explore whether SAM can also achieve better adversarial robustness. Surprisingly, we find that using SAM alone can notably improve adversarial robustness compared to standard optimization methods like Adam, which is an unexpected benefit of SAM. As illustrated in Table 1, SAM can achieve both higher robust and clean accuracy than standard training. Compared to the improvement in clean accuracy, the enhancement of robust accuracy (evaluated by AutoAttack (Croce and Hein, 2020b)) is significantly prominent (**from 3.4% to 25.4%**). By contrast, though AT can exhibit better robustness, their decrease in natural accuracy is unaffordable in practical deployment. Motivated by this intriguing observation, we raise the two following questions and attempt to answer them:

(1) Why can SAM improve adversarial robustness, and

(2) Under what condition can SAM be deployed as a substitution for AT.

[*]Equal contribution [1]Peking University [2]University of California, Berkeley [3]Beijing Institute of Technology [4]MIT CSAIL. Correspondence to: Zeming Wei <weizeming@stu.pku.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Table 1. Examples of clean accuracy and robust accuracy with AutoAttack (AA.) comparison on standard training, SAM, and AT. The robustness is evaluated under ℓ_2 norm $\epsilon = 32/255$. More details in Section 5.

Method	Clean Acc.	AA. Rob. Acc.
Adam	76.0	3.4
SAM	**78.7**	**25.4**
ℓ_∞-AT	58.3	52.4
ℓ_2-AT	64.2	57.5

To study the two questions above, we first provide an intuitive understanding of why SAM can improve adversarial robustness without sacrificing natural accuracy. Specifically, we show that weight perturbation during training can help the model implicitly learn the robust features (Tsipras et al., 2018). To verify this notion, we also provide theoretical insights to support our intuitive understanding. Following the popular data distribution based on robust and non-robust features decomposition (Tsipras et al., 2018), we show that both SAM and AT can improve the robustness of the trained models by **biasing more weight on robust features**. To sum up, we answer the raised question **(1)** by concluding that SAM can improve adversarial robustness by implicitly learning robust features.

Furthermore, we conduct extensive experiments across different tasks, data modalities, and model architectures to evaluate the robustness improvement of using SAM, where we show that under various settings, SAM exhibits consistently better robustness than standard training methods while still maintaining higher natural performance. Besides, though AT can achieve higher robustness than SAM, it inevitably sacrifices natural performance which may be unaffordable for real-world applications. Therefore, we answer the raised question **(2)** with the conclusion that SAM can be used as a substitute for AT when accuracy is more important but better robustness is preferred.

To summarize, our main contributions in this paper are:

1. We uncover an intriguing property of SAM that it can notably enhance adversarial robustness while maintaining natural performance compared to standard training, which is an unexpected benefit.

2. We provide empirical and theoretical insights to understand how SAM can enhance adversarial robustness by showing that both input and weight perturbation can encourage the model to learn robust features.

3. We conduct extensive experiments to show the effectiveness of SAM in terms of enhancing robustness without sacrificing natural performance. We also suggest that SAM can be considered a lightweight substitute for AT when accuracy comes at a higher priority.

2. Background and Related Work

2.1. Sharpnes-Aware Minimization (SAM)

To improve generalization ability in traditional machine learning algorithms, (Hochreiter and Schmidhuber, 1994; 1997) respectively attempt to search for flat minima and penalize sharpness in the loss landscape, which obtains good results in generalization (Keskar et al., 2016; Neyshabur et al., 2017; Dziugaite and Roy, 2017). Inspired by this, a series of works focus on using the concept of *flatness* or *sharpness* in loss landscape to ensure better generalization, *e.g.* Entropy-SGD (Chaudhari et al., 2019) and Stochastic Weight Averaging (SWA) (Izmailov et al., 2018). Sharpness-Aware minimization (SAM) (Foret et al., 2020) also falls into this category, which simultaneously minimizes loss value and loss sharpness as described in (1).

The objective of SAM is to minimize the *sharpness* around the parameters, which can be formulated as

$$\min_w \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|\epsilon\|\le\rho;x,y} L(w+\epsilon) + \lambda\|w\|_2^2, \qquad (1)$$

where $\mathcal{D}$ is the data distribution, L is the loss function, w is the parameters of the model, $\|w\|_2^2$ is the regularization term and ρ controls the magnitude of weight perturbation. To solve the inner maximization process, one-step gradient descent is applied.

So far, SAM has become a powerful tool for enhancing the natural accuracy performance of machine learning models. There are also many applications of SAM in other fields of research like language models (Bahri et al., 2021) and decentralized SGD (Zhu et al., 2023), showing the scalability of SAM to various domains. In addition, many improvements of the algorithm SAM spring up, like Adaptive SAM (ASAM) (Kwon et al., 2021), Efficient SAM (ESAM) (Du et al., 2021), LookSAM (Liu et al., 2022), Sparse SAM (SSAM) (Mi et al., 2022), Fisher SAM (Kim et al., 2022) and SAM-ON (Mueller et al., 2023), which add some modifications on SAM and further improve the generalization ability of the model. However, while these existing works focus on the natural generalization goal, the effectiveness of SAM on adversarial robustness remains unexplored.

2.2. Adversarial Robustness and Adversarial Training

The adversarial robustness and adversarial training have become popular research topics since the discovery of adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014), which uncovers that DNNs can be easily fooled to make wrong decisions by adversarial examples that are crafted by adding small perturbations to normal examples. The malicious adversaries can conduct adversarial attacks by crafting adversarial examples, which cause serious safety concerns regarding the deployment of DNNs. So far, numerous defense approaches have been proposed (Xie et al., 2019; Bai et al., 2019; Cohen et al., 2019; Chen et al., 2024), among which Adversarial Training (**AT**) (Madry et al., 2017) has been considered the most promising defense method against adversarial attacks. AT can be formulated as the following optimization problem:

$$\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|\delta\|\leq\epsilon} L(w; x+\delta, y), \tag{2}$$

where $\mathcal{D}$ is the data distribution, ϵ is the margin of perturbation, w is the parameters of the model and L is the loss function (*e.g.* the cross-entropy loss). For the inner maximization process, the Projected Gradient Descent (PGD) (Madry et al., 2017) attack is commonly used to generate the adversarial example:

$$x^{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}(x^t + \alpha \cdot \text{sign}(\nabla_x \ell(\theta; x^t, y))), \tag{3}$$

where Π projects the adversarial example onto the perturbation bound $\mathcal{B}(x,\epsilon) = \{x' : \|x'-x\|_p \leq \epsilon\}$ and α represents the step size of gradient ascent.

Though improves adversarial robustness effectively, adversarial training has exposed several defects such as computational overhead (Shafahi et al., 2019), class-wise fairness (Xu et al., 2021; Wei et al., 2023b) and robust overfitting (Rice et al., 2020; Wang et al., 2023; Wei et al., 2023a) among which the decreased natural accuracy (Tsipras et al., 2018; Zhang et al., 2019) has become the major concern. One explanation for this drawback of AT is perturbing the samples during training leads the training sample distribution to deviate from the natural data (Ilyas et al., 2019).

In the context of adversarial robustness, several works also attempt to introduce a flat loss landscape in adversarial training (Wu et al., 2020; Yu et al., 2022a;b). The most representative one is Adversarial Weight Perturbation (AWP) (Wu et al., 2020), which simultaneously adds perturbation on examples and feature space to apply SAM and AT. However, AWP also suffers from a decrease in natural accuracy which is even lower than AT in some cases, which we assume is because perturbing both the inputs and parameters significantly raises the difficulty of robust learning. We compare SAM and AWP in our experiment section. Besides this thread of work, the adversarial robustness derived from SAM alone has not been explored.

3. Empirical Understanding

In this section, we provide an intuitive explanation to empirically understand how SAM improves adversarial robustness by demonstrating the duality between SAM and AT. Specifically, considering the arithmetical duality of the input and parameters to get the output in a specific layer, we can assume that the robustness against weight perturbation may also lead to robustness against input perturbation.

We start by rewriting the optimization objective of SAM and AT in a unified form and omit the regularization term $\lambda\|w\|_2^2$ as follows:

$$\min_{\boldsymbol{w}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{||\boldsymbol{\epsilon}||<\rho} L(\boldsymbol{w}+\boldsymbol{\epsilon}; x, y) \quad \text{(SAM)} \tag{4}$$

and

$$\min_{\boldsymbol{w}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|\delta\|\leq\epsilon} L(\boldsymbol{w}; x+\delta, y) \quad \text{(AT)} \tag{5}$$

To illustrate their relation, we first emphasize that both techniques involve adding **perturbation** to make the output more robust *w.r.t.* input or weight changes since they both utilize one or more step gradient optimization to solve the inner maximization problem. However, as AT **explicitly** adds these perturbations to input examples x and transforms them into adversarial examples x_{adv}, the adversarial sample distribution learned through forward-backward passes deviated from the natural distribution, leading to an inevitable decrease when evaluating natural performance in the original distribution. By contrast, SAM applies weight perturbation to achieve this robustness but keeps the original samples during learning, which can implicitly bias more weight on robust features (Ilyas et al., 2019).

To be more formalized, we illustrate our understanding with a middle linear layer in a model, which extracts feature z from input x: $z = Wx$. During AT, we add perturbations directly to the input space, resulting in $x \leftarrow x+\delta$. However, in SAM, the perturbation is not directly applied to the input space, but to the parameter space as $W \leftarrow W + \delta$. This leads to $Wx + W\delta$ for input perturbation and $Wx + \delta x$ for parameter perturbation. If the weight W can keep Wx more robust against small perturbations around it in the subsequent layers, it will be also beneficial to improve sample robustness around x.

Besides, we discuss the attack (perturbation) strength of AT and SAM. For SAM, the perturbation is relatively more moderate, as its perturbations are conducted in the weight space and do not change the original input. On the other hand, in order to achieve the best robustness by eliminating the non-robust features (Ilyas et al., 2019), AT applies larger and more straightforward perturbations to the input space, leading to better robustness but a loss in natural accuracy, which is not the original goal of SAM, but it is the goal of AT.

In summary, our intuitive analysis suggests that SAM applies small perturbations implicitly to the feature space to maintain good natural accuracy performance, while AT utilizes direct input perturbations, which may result in a severe loss in natural accuracy. We provide more theoretical evidence to support these claims in the next section.

4. Theoretical Insights

In this section, we provide a theoretical analysis of SAM and the relation between SAM and AT. Following the *robust/non-robust feature* decomposition (Tsipras et al., 2018), we introduce a simple binary classification model, in which we show the implicit similarity and differences between SAM and AT. We first present the data distribution and hypothesis space, then present how SAM and AT work in this model respectively, and finally discuss their relations.

4.1. A Binary Classification Model

Following a series of theoretical work on adversarial robustness (Tsipras et al., 2018; Ilyas et al., 2019; Xu et al., 2021), we consider a similar binary classification task that the input-label pair $(\boldsymbol{x}, y)$ is sampled from $\boldsymbol{x} \in \{-1,+1\} \times \mathbb{R}^{n+1}$ and $y \in \{-1,+1\}$, and the distribution $\mathcal{D}$ is defined as follows.

$$\begin{aligned} & y \overset{\text{u.a.r}}{\sim} \{-1,+1\}, x_1 = \begin{cases} +y, & \text{w.p. } p, \\ -y, & \text{w.p. } 1-p, \end{cases} \\ & x_2, \ldots, x_{n+1} \overset{i.i.d}{\sim} \mathcal{N}(\eta y, 1), \end{aligned} \tag{6}$$

where $p \in (0.5, 1)$ is the accuracy of feature x_1, constant $\eta > 0$ is a small positive number. In this model, x_1 is called the *robust feature*, since any small perturbation can not change its sign. However, the robust feature is not perfect since $p < 1$. Correspondingly, the features $x_2, \cdots, x_{n+1}$ are useful for identifying y due to the consistency of sign, hence they can help classification in terms of natural accuracy. However, they can be easily perturbed to the contrary side (change their sign) since η is a small positive, which makes them called *non-robust features* (Ilyas et al., 2019).

Now consider a linear classifier model which predicts the label of a data point by computing $f_{\boldsymbol{w}}(\boldsymbol{x}) = \mathrm{sgn}(\boldsymbol{w} \cdot \boldsymbol{x})$, and optimize the parameters $w_1, w_2, \cdots, w_{n+1}$ to maximize $\mathbb{E}_{x.y\sim\mathcal{D}} \mathbf{1}(f_{\boldsymbol{w}}(x) = y)$. Note that value w_1 as the coefficient of the robust feature x_1, may have a strong correlation with the robustness of the model. Specifically, larger w_1 indicates that the model biases more weight on the robust feature x_1 and less weight on the non-robust features $x_2, \cdots, x_{n+1}$, leading to better robustness. Therefore, we consider the weight of parameter w_1 among all features as an indication of how many robust features are learned in the model. Therefore, we define the **robust feature weight** W_R of a given model as

$$W_R = \frac{w_1}{w_2 + w_3 + \cdots + w_{n+1}} \tag{7}$$

to **measure the weight of robust features involved** in the model prediction. We use the following lemma to justify the fundamental relationship between W_R and the adversarial robustness of the model:

Theorem 4.1. *The robust accuracy (R_A) of this model, defined as*

$$R_A = \mathbb{E}_{\mathbf{x},y\sim D} \min_{||\delta||<\epsilon} \mathbf{1}\{f_{\boldsymbol{w}}(x+\delta) = y\}. \tag{8}$$

is a ***monotonic increasing function*** *of W_R under condition $\epsilon < \eta$ and $0 < W_R < W_R^{AT}$(defined in (11)).*

In the following, we derive and compare the robust feature weight W_R of the trained model under standard training (ST), AT, and SAM respectively. To make our description clear, we denote the loss function $\mathcal{L}(\boldsymbol{x}, y, w)$ as $1 - \Pr(f_w(\boldsymbol{x}) = y)$ and for a given $\epsilon > 0$, we define the loss function of SAM $\mathcal{L}^{SAM}$ as $\max_{|\delta|\le\epsilon} \mathcal{L}(\boldsymbol{x}, y, w+\delta)$.

4.2. Standard Training (ST)

We first show that under standard training, the robust feature weight learned in this model can be derived from the following theorem:

Theorem 4.2 (Standard training)**.** *In the model above, under standard training, the robust feature weight W_R is*

$$W_R^* = \frac{\ln p - \ln(1-p)}{2n\eta}. \tag{9}$$

Therefore, W_R^* can be regarded as the measurement W_R returned by standard training with this model.

4.3. Adversarial Training (AT)

Now we consider when AT is applied to train the model. Recall that in this case, the loss function is no longer the standard one but the expected adversarial loss

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{||\boldsymbol{\delta}||_\infty \le \epsilon} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y; w) \right]. \tag{10}$$

Similar to standard training, we can derive the robust feature weight from the following theorem:

Theorem 4.3 (Adversarial training)**.** *In the classification problem above, under adversarial training with perturbation bound $\epsilon < \eta$, the robust feature weight*

$$W_R^{AT} = \frac{\ln p - \ln(1-p)}{2n(\eta - \epsilon)}. \tag{11}$$

We can see that W_R has been multiplied by $\frac{\eta}{\eta-\epsilon} > 1$, which has **increased** the dependence on the robust feature x_1 of the classifier. This shows the adversarially trained model pays more attention to robust features compared to the standard-trained one, which improves adversarial robustness.

4.4. Sharpness-Aware Minimization (SAM)

Now we consider the situation of SAM. Recall that the optimizing objective of SAM is

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\max_{|\delta|\leq\epsilon}\mathcal{L}(x,y;w+\delta)\right]. \tag{12}$$

We first explain why SAM could improve the adversarial robustness by proving that the measurement W_R trained with SAM W_R^{SAM} is also larger than W_R^*, which is stated as follows:

Theorem 4.4 (Sharpness-aware minimization)**.** *In the classification problem above, the robust feature weight for SAM training W_R^{SAM} satisfies that*

$$W_R^{SAM} > W_R^*. \tag{13}$$

From Theorem 4.3 and 4.4 we can see that both W_R^{AT} and W_R^{SAM} are greater than W_R^*, which indicates both SAM and AT encourages the trained model to learn more robust features. However, the qualitative relation is not sufficient to quantify how much robustness SAM achieves compared to adversarial training, and we attempt to step further by quantitatively estimating the W_R^{SAM} in the following theorem:

Theorem 4.5. *In the classification problem above, suppose that $\epsilon > 0$ is small, we have*

$$W_R^{SAM} \approx W_R^* + \frac{2}{3}W_R^*\epsilon^2. \tag{14}$$

4.5. Relation between SAM and AT

We further discuss the distinct attack (perturbation) strength between AT and SAM. Recall that in our empirical understanding in Section 3, the perturbation of SAM is more moderate and implicit than AT. Therefore, to reach the same robustness level (which is measured by the robust feature weight W_R), SAM requires a much larger perturbation range, while for AT, less perturbation over x is enough. Theoretically, the following theorem verifies our statement:

Theorem 4.6. *Denote the perturbation range ϵ of AT and SAM as ϵ_{AT} and ϵ_{SAM}, respectively. Then, when both methods return the same robust feature weight W_R, we have the following relation between ϵ_{AT} and ϵ_{SAM}:*

$$2 + \frac{3}{\epsilon_{SAM}^2} \approx \frac{2\eta}{\epsilon_{AT}} \tag{15}$$

From theorem 4.6, we can identify the different perturbation strengths of AT and SAM. It can be easily derive from Theorem 4.6 that ϵ_{SAM} is larger than ϵ_{AT} when (15) holds, since we assume η is a small positive, ϵ is small in Theorem 4.5 and $\epsilon_{AT} < \eta$ in Theorem 4.3. Therefore, to gain the same weight w_1 on robust features x_1, ϵ_{AT} only need to be chosen much smaller than ϵ_{SAM}. On the other hand, under the same perturbation bound $\epsilon_{AT} = \epsilon_{SAM}$, the model trained under AT has larger W_R than SAM, hence it focuses on more robustness yet decreases more natural accuracy.

All proofs can be found in Appendix A. While we acknowledge that W_R is only analyzed under a simple model and cannot be directly generalized to multiple-layer networks, we believe the insights delivered from W_R can be generalized to DNNs. To sum up, we can conclude that AT utilizes explicit and direct perturbations for eliminating non-robust features, while SAM leverages implicit and moderate perturbations to learn robust features. This is consistent with our empirical understanding in Section 3 and we also verify these claims with experiments in the following section.

5. Experiment

In this section, we conduct extensive experiments to show the effectiveness of SAM in improving robustness while maintaining natural performance, across multiple tasks, data modalities, and various settings. We start with the classic image classification task, seconded by semantic segmentation (vision) and text classification (language).

5.1. Image Classification

5.1.1. Experimental settings

Training configurations. We mainly consider the vanilla SAM (Foret et al., 2020) optimizer with the perturbation hyper-parameter ρ from the range $\{0.1, 0.2, 0.3, 0.4\}$. In addition, we also explore two distinct variants of SAM including Adaptive Sharpness-Aware Minimization (ASAM) (Kwon et al., 2021) and Efficient Sharpness-aware Minimization (ESAM) (Du et al., 2022). Following Pang et al. (2020) and Wei et al. (2023b), we set the weight decay as `5e-4` and momentum as 0.9 and train 100 epochs with the learning rate initialized as 0.1 for SGD and `1e-3` for Adam, and is divided by 10 at the 75th and 90th epochs, respectively. For AT, we consider both ℓ_2 and ℓ_∞ norms.

Datasets and models. We examine the robustness of SAM on CIFAR-{10,100} (Krizhevsky et al., 2009) and TinyImageNet (Chrabaszcz et al., 2017) datasets. We mainly conduct our experiment with PreActResNet-18 (PRN-18) (He et al., 2016b). To demonstrate the scalability of SAM, we also include the Wider ResNet (WRN-28-10) (Zagoruyko and Komodakis, 2016) and the vision transformer architecture (DeiT)(Touvron et al., 2021). To evaluate the corruption robustness of various models, we also use the CIFAR-10C dataset (Hendrycks and Dietterich, 2019), a common corruptions dataset for CIFAR10 with 1000 randomly selected examples from CIFAR-10C and level-3 of perturbation severity for each corruption types (fog, snow, *etc*).

Table 2. Natural and robust accuracy evaluation on **CIFAR-10** dataset.

Method	Natural Accuracy	FGSM $\epsilon = \frac{1}{255}$	ℓ_∞-PGD $\epsilon = \frac{1}{255}$	ℓ_2-PGD $\epsilon = \frac{32}{255}$	ℓ_2-AA. $\epsilon = \frac{32}{255}$	StAdv	FAB	Pixle	Average Robustness
SGD	94.5	63.4	37.9	41.5	31.7	35.2	44.8	10.0	**37.8**
Adam	93.9	44.3	17.4	20.7	13.9	20.4	24.7	7.6	21.3
SAM ($\rho = 0.1$)	95.4	63.3	46.2	48.7	43.6	39.3	49.2	13.4	43.4
SAM ($\rho = 0.2$)	**95.5**	66.7	51.3	53.4	48.1	44.2	53.4	13.2	47.2
SAM ($\rho = 0.3$)	95.4	66.6	51.2	53.5	47.8	46.1	53.8	13.7	47.5
SAM ($\rho = 0.4$)	94.7	69.6	56.4	58.6	51.8	54.9	57.6	14.3	**51.9**
AT (ℓ_∞-$\epsilon = \frac{8}{255}$)	84.5	81.9	81.8	79.7	79.5	82.0	79.5	26.9	73.0
AT (ℓ_2-$\epsilon = \frac{128}{255}$)	89.2	84.1	84.1	84.8	84.8	80.4	84.8	32.0	76.4

Table 3. Natural and robust accuracy evaluation on **CIFAR-100** dataset.

Method	Natural Accuracy	FGSM $\epsilon = \frac{1}{255}$	ℓ_∞-PGD $\epsilon = \frac{1}{255}$	ℓ_2-PGD $\epsilon = \frac{32}{255}$	ℓ_2-AA. $\epsilon = \frac{32}{255}$	StAdv	FAB	Pixle	Average Robustness
SGD	76.5	30.9	13.3	17.0	11.4	14.5	16.9	1.8	**15.1**
Adam	76.0	20.5	5.3	5.8	3.4	7.4	8.4	1.3	7.4
SAM ($\rho = 0.1$)	77.7	36.3	20.3	24.1	19.6	20.9	24.5	3.4	21.3
SAM ($\rho = 0.2$)	**78.8**	38.1	22.9	25.6	20.2	23.7	25.2	3.7	22.8
SAM ($\rho = 0.3$)	78.7	40.2	25.5	28.1	22.6	26.4	27.5	4.3	24.9
SAM ($\rho = 0.4$)	78.7	41.8	29.5	31.8	25.4	29.4	30.0	5.4	**27.6**
AT (ℓ_∞-$\epsilon = \frac{8}{255}$)	58.3	55.1	55.0	53.8	52.4	50.6	52.7	7.3	46.7
AT (ℓ_2-$\epsilon = \frac{128}{255}$)	64.2	56.6	56.6	58.1	57.5	52.2	58.2	10.5	49.9

Attacks for robustness evaluations. We assess the model's resistance through a multifaceted approach to thoroughly explore its robustness against different threat models. We apply the Fast Gradient Sign Method (**FGSM**) (Goodfellow et al., 2015) with a perturbation bound of ℓ_∞-$\epsilon = 1/255$, and a 10-step Projected Gradient Descent (**PGD**) attack under ℓ_∞ and ℓ_2 norms with bounds of $\epsilon = 1/255$ and $32/255$, respectively. Moreover, we incorporate the **AutoAttack** (AA.) (Croce and Hein, 2021) to ensure a reliable evaluation. Besides, we also consider other popular attacks like StAdv (Xiao et al., 2018), FAB (Croce and Hein, 2020a) (10 steps under ℓ_2-$\epsilon = 32/255$). For black-box assessment, we utilize Pixle (Pomponi et al., 2022), a strong pixel-rearranging black-box attack, constrained to a maximum of five iterations. The attacks are mostly implemented by the `torchattacks` (Kim, 2020) framework to ensure a reliable assessment. For corruption robustness (Hendrycks and Dietterich, 2019), which measures the robustness of DNNs against a wide range of real-world disturbances like noise, blur, or weather variations, we evaluate models on RobustBench (Croce et al., 2021), a standardized benchmark designed to evaluate the general robustness. We ran all experiments three times independently to report the average result and omitted the standard deviations since they are small (less than 0.5%) and do not affect our claims.

5.1.2. EXPERIMENTAL RESULTS

Comparison with standard training. As shown in Tables 2 to 4, all models trained with SAM exhibit significantly better natural accuracy and robustness compared to those trained with standard training (ST). In particular, higher robustness is achieved by using larger values of ρ with SAM. Taking the CIFAR-100 dataset as an example, the model trained with $\rho = 0.4$ demonstrates even multiple robust accuracy than ST, and its natural accuracy is still higher than that of ST. Compared to the improvement in natural accuracy (approximately 2%), the increase of robustness is more significant (**more than 10% in average**). Moreover, on large datasets like Tiny-ImageNet, SAM still surpasses ST by a large margin. Therefore, we conclude that SAM with a relatively larger weight perturbation bound ρ is a promising technique for enhancing adversarial robustness without sacrificing natural accuracy. Notably, our results are consistent with concurrent work showing that the robustness of SGD is better than Adam (Ma et al., 2023).

Comparison with AT. Regarding adversarially trained models, although there remains a large gap between the robustness obtained by SAM and AT, all adversarially trained models exhibit significant lower natural accuracy than standard training and SAM. Particularly, as demonstrated in

Table 4. Comparison on Tiny ImageNet.

Config	Natural	AA.	StAdv
SGD	57.4	2.5	3.4
SAM ($\rho = 0.4$)	**57.9**	10.3	**13.3**
SAM ($\rho = 0.5$)	57.7	**10.4**	12.7
ℓ_∞-AT($\epsilon = 8/255$)	32.3	27.7	25.1
ℓ_2-AT($\epsilon = 128/255$)	41.6	32.8	30.2

Table 5. Comparing SAM and AT with small ϵ on CIFAR-100 dataset.

Method	Config	Natural	AA.	Pixle
Standard	SGD	76.5	11.4	1.8
	Adam	76.0	3.4	1.3
SAM	$\rho = 0.4$	**78.7**	25.4	5.4
	$\rho = 0.5$	77.8	**30.1**	**8.1**
ℓ_∞-AT	$\epsilon = 1/255$	72.1	58.6	4.7
	$\epsilon = 2/255$	69.4	56.8	4.2
	$\epsilon = 4/255$	64.3	55.7	6.3
	$\epsilon = 8/255$	58.3	52.4	7.3
ℓ_2-AT	$\epsilon = 16/255$	74.3	57.9	5.7
	$\epsilon = 32/255$	72.5	57.2	5.2
	$\epsilon = 64/255$	69.0	57.3	8.2
	$\epsilon = 128/255$	64.2	57.5	10.5

Tables 5 and 6, even training with extremely small perturbation bound like $\epsilon = 1/255$ decreases natural accuracy at 3.8% for CIFAR-100 and 0.9% for CIFAR-10 datasets, respectively. The larger perturbation bound ϵ used in AT, the worse natural accuracy is obtained by the corresponding model. Therefore, a key benefit of using SAM instead of AT is that there is no decrease in clean accuracy. Additionally, note that AT requires significant computational overhead. Specifically, for 10-step PGD, all AT experiments require 10 times more computational cost compared to ST. Moreover, the cutting-edge efficient SAMs (Liu et al., 2022; Chen et al., 2022) are even faster than FastAT (Wong et al., 2020).

Scalability to various architectures. Table 6 presents the results for three distinct models: PRN-18, WRN, and DeiT. Regardless of the architecture, SAM consistently improved robustness against natural, adversarial, and StAdv attacks compared to standard SGD. On PRN-18, SAM yielded impressive gains on both AA and StAdv attacks, boosting robustness by 11.9% and 4.1%, respectively. Similar trends were observed with WRN, achieving robustness improvements of 2.4% and 4.7% for AA and StAdv attacks. Notably, even for DeiT, where SGD performance was lower, SAM still offered significant gains in robustness. These results convincingly demonstrate the broad applicability and scalability of SAM in enhancing robustness across diverse model architectures.

Table 6. Comparison of different model architectures on CIFAR-10 dataset with $\epsilon = 1/255$.

Model	Method	Natural	AA.	StAdv
PRN-18	SGD	94.5	31.7	35.2
	SAM	**95.4**	43.6	39.3
	ℓ_∞-AT	84.5	81.4	82.0
WRN	SGD	95.2	39.6	38.3
	SAM	**95.6**	42.0	43.0
	ℓ_∞-AT	87.2	79.7	76.6
DeiT	SGD	69.2	21.3	11.2
	SAM	**69.4**	24.2	19.6
	ℓ_∞-AT	62.2	51.9	54.2

Variants of SAM. As shown in Table 7, ASAM and ESAM generally outperform standard optimizers (SGD and Adam) in terms of robustness, demonstrating the effectiveness of adaptive momentum for adversarial defense. Notably, ESAM achieved the highest robustness against StAdv attacks (49.6% at $\rho = 0.4$), while SAM yielded the best robustness against the AA attack (58.4% at $\rho = 0.5$). These results highlight the robustness of various SAMs against adversarial threats.

Table 7. Comparison of variants of SAM on CIFAR-10 dataset.

Method	Config	Natural	AA.	StAdv
Standard	SGD	94.5	31.7	35.2
	Adam	93.9	13.9	20.4
SAM	$\rho = 0.3$	**95.4**	57.8	46.1
	$\rho = 0.4$	94.7	51.8	54.9
	$\rho = 0.5$	94.5	**58.4**	**55.8**
ESAM	$\rho = 0.3$	**95.2**	53.1	48.2
	$\rho = 0.4$	94.7	51.6	**49.6**
	$\rho = 0.5$	94.4	**53.6**	49.3
ASAM	$\rho = 0.3$	95.4	39.9	34.7
	$\rho = 0.4$	95.5	40.4	35.6
	$\rho = 0.5$	**95.7**	**44.2**	**36.6**

Corruption Robustness. Unlike robustness against adversarial attacks, common corruption robustness captures a wider spectrum of realistic disturbances beyond artificially crafted perturbations, showcasing how models can handle varied and unexpected environmental changes. We evaluated SAM and AT on corrupted images in CIFAR-10C and calculated the average accuracy across 9 different common corruptions. We present these experimental results in Table 8, with detailed accuracy for each corruption type available in Table 13 in Appendix B.

Unlike in adversarial attack scenarios where SAM cannot outperform AT in terms of robustness, models trained with

Table 8. Accuracy on common corruptions from CIFAR-10C.

Method	Natural	Corruption (Avg.)
SGD	94.5	34.91
Adam	93.9	29.60
SAM ($\rho = 0.3$)	**95.4**	32.60
ASAM ($\rho = 0.3$)	**95.4**	**36.69**
ESAM ($\rho = 0.3$)	95.2	36.29
ℓ_∞-AT($\epsilon = 8/255$)	84.5	15.67
ℓ_2-AT($\epsilon = 128/255$)	89.2	23.13

SAM have demonstrated significantly superior robustness compared to AT. As shown in the table, ℓ_p-norm-AT performs even worse than standard training (15.67% for ℓ_∞-AT with $\epsilon = 8/255$, compared to 34.91% for SGD). In contrast, SAM achieves similar results on corrupted images without significant performance degradation, showcasing the stability of SAM in realistic perturbation scenarios. Surprisingly, we found that variants of SAM can perform even better on corrupted images compared to standard training (with an accuracy increase of around 2%), highlighting the potential of the SAM method for addressing real-world robustness issues.

Comparison with AWP. Adversarial Weight Perturbation (AWP) (Wu et al., 2020) is an adversarial training method designed for better adversarial robustness by introducing a double-perturbation mechanism that adversarially perturbs both inputs and weights during training. Specifically, AWP can be regarded as a combination of SAM and AT. As AWP also leverages the weight perturbation paradigm, we provide a comparison with AWP and SAM to showcase that SAM still outperforms AWP in terms of natural accuracy.

The experimental results are shown in Table 9. Similar to the comparison between SAM and AT, we can see that although AWP outperforms both SAM and AT in terms of adversarial robustness, the natural accuracy with SAM is still higher than AWP even though weight perturbation was incorporated in it, indicating the importance of using natural data instead of adversarial examples for training to maintain the natural accuracy.

Table 9. Comparison of SAM, AT, and AWP on CIFAR-10.

Method	Natural	AA.
SAM ($\rho = 0.1$)	**95.4**	43.6
SAM ($\rho = 0.4$)	94.7	51.8
ℓ_∞-AT ($\epsilon = 8/255$)	84.5	79.5
ℓ_∞-AWP ($\epsilon = 8/255$)	82.0	80.1
ℓ_2-AT ($\epsilon = 128/255$)	89.2	84.8
ℓ_2-AWP ($\epsilon = 128/255$)	89.7	**86.3**

5.2. Semantic Segmentation

Experimental settings. We train DeepLabv3 (Chen et al., 2017) with randomly initialized ResNet-50 (He et al., 2016a) backbone on the Stanford background dataset (Gould et al.) with Cross-Entropy loss for 30000 iterations. For experiments on VOC2012 dataset (Everingham et al.), we use mobilenetv2 (Sandler et al., 2018) backbone, and the weights are initialized with pre-trained on ImageNet. The learning rate is initialized as 0.01 with a polynomial decay scheme. For the optimizer, the weight decay is set to `1e-4`, and the momentum is set to 0.9. We use the **mean Intersection over Union (mIoU)** (Everingham et al., 2015) to evaluate the segmentation results, where the IoU is calculated for each class at the pixel level as

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \times 100\%, \tag{16}$$

where TP, FN and FP represent true positive, false negative and false positive, respectively. The mIoU is then the mean value across all the classes in the dataset. For SAM, we select ρ from $\{0.01, 0.02, 0.03, 0.04, 0.05\}$.

Table 10. Comparing SAM and AT for semantic segmentation on Stanford background dataset, mIoU is used for evaluation.

Method	Config	Natural	ℓ_∞-PGD	ℓ_2-PGD
Standard	SGD	64.0	57.3	57.0
	Adam	62.2	57.1	56.8
SAM	$\rho = 0.01$	64.3	58.2	58.0
	$\rho = 0.02$	**64.8**	**58.7**	57.6
	$\rho = 0.03$	64.5	58.3	**58.0**
	$\rho = 0.04$	64.6	57.9	57.6
	$\rho = 0.05$	64.6	57.8	57.4
ℓ_∞-AT	$\epsilon = 1/255$	59.9	**59.0**	**59.2**
	$\epsilon = 2/255$	58.2	57.7	57.8
	$\epsilon = 4/255$	55.3	54.9	55.1
	$\epsilon = 8/255$	57.5	55.1	54.9

Experimental results. The experiment results are summarized in Table 10 and Table 11, where we use a 10-step PGD attack under ℓ_∞-$\epsilon = 1/255$ and ℓ_2-$\epsilon = 1$ for robustness evaluation, respectively. As shown in the table, models trained using SAM consistently achieve better segmentation performance than ST and significantly better than AT, and exhibit notably better robustness than ST, showcasing the robust generalization ability of SAM in various visual tasks. We also observed that the mIoU decrease of segmentation tasks is not as significant as the accuracy decrease of classification under adversarial attacks, which we assume is because the segmentation task is a highly interpretive task and many low-level features also play an important role in the segmentation results (Ren and Malik, 2003; Zhu et al., 2021), making segmentation models inherently more robust than classification models.

Table 11. Comparing SAM and AT for semantic segmentation on VOC2012 dataset, mIoU on the validation set is used for evaluation.

Method	Config	Natural	ℓ_∞-PGD	ℓ_2-PGD
Standard	SGD	66.9	37.7	47.1
	Adam	65.2	28.9	36.9
SAM	$\rho = 0.01$	67.9	38.9	48.6
	$\rho = 0.02$	67.7	40.2	50.1
	$\rho = 0.03$	68.3	38.8	49.9
	$\rho = 0.04$	67.5	39.1	49.1
	$\rho = 0.05$	**68.6**	**40.4**	**50.6**
ℓ_∞-AT	$\epsilon = 1/255$	50.8	50.2	50.4
	$\epsilon = 2/255$	49.2	48.7	49.0
ℓ_2-AT	$\epsilon = 64/255$	60.3	**56.9**	**58.4**
	$\epsilon = 128/255$	56.5	54.3	55.2
	$\epsilon = 255/255$	53.7	52.3	52.8

5.3. Text Classification

Language models are also shown to be vulnerable against adversarial attacks (Morris et al., 2020; Wei et al., 2024). While adversarial attack (Zou et al., 2023; Dong et al., 2023; Wei et al., 2023c; Zhang and Wei, 2024) and defense (Xie et al., 2023; Piet et al., 2023; Mo et al., 2024; Wang et al., 2024) on Large Language Models (LLMs) have become emergent research topics recently, it is worth noting that adversarial training on LLMs may not be helpful and practical (Jain et al., 2023), further underscoring the potential of deploying SAM a defense for language models.

Experimental settings. In this part, we explore the effectiveness of SAM in enhancing the robustness of text sequence classification models against adversarial attacks. Utilizing the Rotten Tomatoes dataset (Pang and Lee, 2005) for sentiment analysis, we study the word-level adversarial robustness of models trained with different methods. Specifically, we consider two popular attacks: the Improved Genetic Algorithm (IGA) (Wang et al., 2021) and the Particle Swarm Optimization (PSO) (Zang et al., 2020), both of which are officially supported by the TextAttack framework (Morris et al., 2020). For the model under test, we selected the widely recognized `distilbert-base-uncased` model from Hugging Face's transformers library (Sanh et al., 2019), fine-tuning it for 3 epochs on the target dataset. The optimization for ST was performed using the AdamW optimizer with a learning rate of `5e-5`, and the SAM optimization strategy was applied with the same base learning rate for AdamW and $\rho = 0.05$. We also compare SAM with AT which incorporates adversarial examples identified in various adversarial attacks into the training dataset. Adversarial attacks such as IGA, PSO, and Textfooler (Jin et al., 2020) have been chosen for data augmentation purposes. The official implementation in TextAttack provides specific details for training in this regard.

Experimental results. The experimental results, summarized in Table 12, show the nuanced performance differences between SAM and ST. Although both SAM and ST achieved comparable natural accuracy (84.6% *v.s.* 84.4%), the distinction in attack success rates (ASR) is pronounced, *e.g.* a reduced ASR of 17.1% under SAM compared to 22.7% under ST, indicating a significant improvement in adversarial robustness. As for AT, adding adversarial examples to the training set has been found to lower natural accuracy without significantly improving adversarial robustness. On average, the ASR for models trained with AT is 2.9% higher with PSO attack and 2.6% higher with IGA attack, indicating that adversarially trained models hurt overall performance against adversarial attacks, although they may be effective at defending specific types of attacks.

Table 12. Accuracy and ASR comparison between SAM and ST for text sequence classification.

Method	Natural Accuracy ($\uparrow$)	PSO ASR ($\downarrow$)	IGA ASR ($\downarrow$)
ST (AdamW)	84.4	22.7	84.1
SAM	**84.6**	**17.1**	**74.5**
AT (w/IGA)	84.0	22.1	71.7
AT (w/PSO)	83.8	16.9	86.8
AT (w/Textfooler)	83.6	21.1	72.9

These findings suggest that incorporating SAM can notably improve the adversarial robustness of text classification models without compromising natural performance, offering a promising avenue for future research and application to secure AI models against adversarial threats.

6. Conclusion

In this paper, we reveal the duality relationship between Sharpness-Aware Minimization (SAM) and Adversarial Training (AT) and show that using SAM alone can improve adversarial robustness. We first intuitively illustrate that both SAM and AT can learn robust features by demonstrating the duality between weight and sample perturbations, and provide theoretical justifications to support these insights. We further conduct extensive experiments under various settings to show that SAM can improve robustness without sacrificing natural performance, while AT inevitably hurts natural generalization. These results uncover the scalability and practicality of using SAM to improve robustness without compromising accuracy on clean data. Based on this, we propose that SAM can be used as a lightweight alternative to AT when accuracy is a priority and improved robustness is preferred.

Impact Statement

This work sheds light on how Sharpness-Aware Minimization (SAM), previously known for its clean accuracy improvements, can unexpectedly enhance the security of deep learning models by boosting their adversarial robustness. This finding presents a potential alternative to Adversarial Training (AT), a widely used but accuracy-sacrificing defense. By offering improved security without compromising clean accuracy, our work could significantly impact the deployment of reliable machine learning models in sensitive domains like healthcare, finance, and autonomous systems. However, responsible development and deployment practices are crucial, as robust models can also be misused. We emphasize the need for continued research into diverse defense mechanisms, fairness, interpretability, and theoretical underpinnings to ensure a secure and ethical future for machine learning.

Acknowledgement

This work was sponsored by the Beijing Natural Science Foundation's Undergraduate Initiating Research Program (Grant No. QY23041) and the National Natural Science Foundation of China (Grant No. 62172019, 623B2001, 82371112, 62394311).

References

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *ICML*, 2022. 1

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021. 1, 2

Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019. 3

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019. 2

Huanran Chen, Shitong Shao, Ziyi Wang, Zirui Shang, Jin Chen, Xiaofeng Ji, and Xinxiao Wu. Bootstrap generalization ability from loss landscape perspective. In *European Conference on Computer Vision*, pages 500–517. Springer, 2022. 1, 7

Huanran Chen, Yinpeng Dong, Zhengyi Wang, Xiao Yang, Chengqi Duan, Hang Su, and Jun Zhu. Robust classification via a single diffusion model. *arXiv preprint arXiv:2305.15241*, 2023a. 1

Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. In *The Twelfth International Conference on Learning Representations*, 2023b. 1

Huanran Chen, Yinpeng Dong, Shitong Shao, Zhongkai Hao, Xiao Yang, Hang Su, and Jun Zhu. Your diffusion model is secretly a certifiably robust classifier. *arXiv preprint arXiv:2402.02316*, 2024. 3

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 8

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 5

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019. 1, 3

F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020a. 6

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020b. 1

Francesco Croce and Matthias Hein. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In *ICML*, 2021. 6

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS*, 2021. 6

Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google's bard to adversarial image attacks? In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. 9

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent YF Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021. 1, 2

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Y. F. Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2022. 5

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017. 2

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 8

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.*, 2015. 8

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 1, 2, 5

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015. 6

Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV 2009*. 8

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016a. 8

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016b. 5

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 5, 6

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *NeurIPS*, 1994. 2

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 1997. 2

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019. 3, 4, 14

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. 9

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment, 2020. 9

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016. 2

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 6

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *ICML*, 2022. 2

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *ICML*, 2021. 1, 2, 5

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *CVPR*, 2022. 2, 7

Avery Ma, Yangchen Pan, and Amir massoud Farahmand. Understanding the robustness difference between stochastic gradient descent and adaptive gradient methods. *arXiv preprint arXiv:2308.06703*, 2023. 6

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 3

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *arXiv preprint arXiv:2210.05177*, 2022. 2

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. Studious bob fight back against jailbreaking via prompt adversarial tuning. *arXiv preprint arXiv:2402.06255*, 2024. 9

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *EMNLP*, 2020. 9

Maximilian Mueller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. *arXiv preprint arXiv:2306.04226*, 2023. 2

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *NeurIPS*, 30, 2017. 2

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 2005. 9

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020. 5

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016. 1

Julien Piet, Maha Alrashed, Chawin Sitawarin, Sizhe Chen, Zeming Wei, Elizabeth Sun, Basel Alomair, and David Wagner. Jatmo: Prompt injection defense by task-specific finetuning. *arXiv preprint arXiv:2312.17673*, 2023. 9

Jary Pomponi, Simone Scardapane, and Aurelio Uncini. Pixle: a fast and effective black-box attack based on rearranging pixels. In *IJCNN*, 2022. 6

Ren and Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 8

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020. 3

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 8

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. 9

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *NeurIPS*, 2019. 3

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention. *arXiv preprint arXiv:2012.12877*, 2021. 5

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1, 2, 3, 4, 14

Xiaosen Wang, Hao Jin, Yichen Yang, and Kun He. Natural language adversarial defense through synonym encoding. *arXiv preprint arXiv:1909.06723*, 2021. 9

Yifei Wang, Liangchen Li, Jiansheng Yang, Zhouchen Lin, and Yisen Wang. Balance, imbalance, and rebalance: Understanding robust overfitting from a minimax game perspective. *NeurIPS*, 2023. 3

Yifei Wang, Yuyang Wu, Zeming Wei, Stefanie Jegelka, and Yisen Wang. A theoretical understanding of self-correction through in-context alignment. *arXiv preprint arXiv:2405.18634*, 2024. 9

Zeming Wei, Yiwen Guo, and Yisen Wang. Characterizing robust overfitting in adversarial training via cross-class features. *OpenReview preprint*, 2023a. 3

Zeming Wei, Yifei Wang, Yiwen Guo, and Yisen Wang. Cfa: Class-wise calibrated fair adversarial training. In *CVPR*, 2023b. 3, 5

Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023c. 9

Zeming Wei, Xiyue Zhang, Yihao Zhang, and Meng Sun. Weighted automata extraction and explanation of recurrent neural networks for natural language tasks. *Journal of Logical and Algebraic Methods in Programming*, 2024. 9

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 7

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 3, 8

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018. 6

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 1, 2

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 2023. 9

Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *ICML*, 2021. 3, 4, 14

Chaojian Yu, Bo Han, Mingming Gong, Li Shen, Shiming Ge, Bo Du, and Tongliang Liu. Robust weight perturbation for adversarial training. *arXiv preprint arXiv:2205.14826*, 2022a. 3

Chaojian Yu, Bo Han, Mingming Gong, Li Shen, Shiming Ge, Bo Du, and Tongliang Liu. Robust weight perturbation for adversarial training. *arXiv preprint arXiv:2205.14826*, 2022b. 3

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *ACL*, 2020. 9

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 3

Yihao Zhang and Zeming Wei. Boosting jailbreak attack with momentum. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024. 9

Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *CVPR*, 2021. 8

Tongtian Zhu, Fengxiang He, Kaixuan Chen, Mingli Song, and Dacheng Tao. Decentralized sgd and average-direction sam are asymptotically equivalent. *arXiv preprint arXiv:2306.02913*, 2023. 2

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. 9

A. Proofs

In this section, we provide all the proof for the theorems. To start with, we introduce a property of our model in the data distribution, which has been proved and used in a series of previous works (Tsipras et al., 2018; Ilyas et al., 2019; Xu et al., 2021):

Lemma (Tsipras et al., 2018) In the presented model and data distribution, given w_1, the optimal solution for the optimization objective will assign equal weight to all the non-robust features, *i.e.* $w_2 = w_3 = \cdots = w_{n+1}$.

The conclusion is proved in Lemma D.1 in Tsipras et al. (2018). Based on this lemma, since we only focus on the ratio $W_R = \frac{w_1}{w_2+w_3+\cdots+w_{n+1}}$, we can further assume $w_2 = w_3 = \cdots = w_{n+1} = 1$ without loss of generalization.

A.1. Proof for Theorem 4.1

Proof. As x_1 has been chosen to be in ± 1, the perturbation over x_1 has no influence on it, and we can just ignore it. Therefore, to attack the classifier by a bias $\boldsymbol{\delta}$, to make the accuracy as small as possible, an intuitive idea is to set $\boldsymbol{\delta}$ which minimizes the expectation of $x_i (i = 2, \cdots, n+1)$, which made the standard accuracy smaller. In fact, the expected accuracy is monotonically increasing about each $\delta_i (i = 2, \cdots, n+1)$. Thus, choosing $\delta = (0, -\epsilon, \cdots, -\epsilon)$ can be the best adversarial attack vector for any $w > 0$. In this situation, this equals $x'_i (i = 2, \cdots, n+1) \sim \mathcal{N}(\eta - \epsilon, n)$.

Thus, R_A can be rewritten to be

$$
\begin{aligned}
R_A &= \mathbb{E}_{\mathbf{x}, y \sim D} \mathbb{P}(w_1 x_1 + \sum_{i=2}^{n+1} x_i - n\epsilon > 0) \\
&= p\Phi((w_1 + (\eta - \epsilon)n)/\sqrt{n}) + (1-p)\Phi((-w_1 + (\eta - \epsilon)n)/\sqrt{n}),
\end{aligned} \tag{17}
$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution.

Since W_R is obviously a monotonic increasing function of w_1, we only need to prove that RA in the above equation is a monotonic increasing function of w_1. To show this, we take a partial derivative of w_1 in RA:

$$
\frac{\partial RA}{\partial w_1} = p\exp(-(w_1 + (\eta - \epsilon)n)^2/2n)/\sqrt{2\pi n} - (1-p)\exp(-(w_1 - (\eta - \epsilon)n)^2/2n)/\sqrt{2\pi n}. \tag{18}
$$

Simplifying this, we get

$$
\frac{\partial R_A}{\partial w_1} = g(w_1) \cdot (1 - \frac{1-p}{p}\exp(2w_1(\eta - \epsilon))), \tag{19}
$$

where $g(w_1) = \frac{p}{\sqrt{2\pi n}}\exp(-(w_1 + (\eta - \epsilon)n)^2/2n)$ is positive . When $0 < W_R < W_R^{AT}$, or equivalantly $0 < w_1 < \frac{\ln p - \ln(1-p)}{2(\eta - \epsilon)}$, we can easily know that $(1 - \frac{1-p}{p}\exp(2w_1(\eta - \epsilon))) > 0$, which implies that $\frac{\partial R_A}{\partial w_1} > 0$, and further means that RA is a monotonic increasing function of W_R. ◻

A.2. Proof for Theorem 4.2

Proof. Due to symmetry, we only need to calculate the case of $y = 1$ without loss of generality. From the distribution, we can easily derive that $x_2 + \cdots + x_{n+1} \sim \mathcal{N}(\eta n, n)$.

Thus, since $w_2 = w_3 = \cdots w_{n+1} = 1$ are assumed to be fixed, we can know that the best parameter w_1 satisfies

$$
\begin{aligned}
w_1^* =& \underset{w_1}{\text{argmax}}\ \mathbb{E}_{\boldsymbol{x}.y \sim \mathcal{D}} \mathbf{1}_{f_{\boldsymbol{w}}(\boldsymbol{x}) = y} \\
=& \underset{w_1}{\text{argmax}}\ p\Pr(x_2 + \cdots + x_{n+1} > -w_1) + (1-p)\Pr(x_2 + \cdots + x_{n+1} > w_1) \\
=& \underset{w_1}{\text{argmax}}\ \frac{p}{\sqrt{2\pi n}} \int_{-w_1}^{\infty} e^{-(t-\eta n)^2/2n} \mathrm{d}t + \frac{1-p}{\sqrt{2\pi n}} \int_{w_1}^{\infty} e^{-(t-\eta n)^2/2n} \mathrm{d}t \\
:=& \underset{w_1}{\text{argmax}}\ u(w_1).
\end{aligned} \tag{20}
$$

Then, the best parameter w_1 can be derived by $\mathrm{d}u/\mathrm{d}w_1 = 0$. The derivative is

$$\frac{\mathrm{d}u}{\mathrm{d}w_1} = \frac{p}{\sqrt{2\pi n}} e^{-(w_1+\eta n)^2/2n} - \frac{1-p}{\sqrt{2\pi n}} e^{-(w_1-\eta n)^2/2n} = 0. \tag{21}$$

Solving this, we get the optimal value of w_1 is

$$w_1^* = \frac{\ln p - \ln(1-p)}{2\eta}. \tag{22}$$

Therefore, W_R^* under the optimal value of $\boldsymbol{w}$ is

$$W_R^* = \frac{\ln p - \ln(1-p)}{2n\eta}. \tag{23}$$

□

A.3. Proof for Theorem 4.3

Proof. As x_1 has been chosen to be in ± 1, the perturbation over x_1 has no influence on it, and we can just ignore it. Therefore, to attack the classifier by a bias $\boldsymbol{\delta}$, to make the accuracy as small as possible, an intuitive idea is to set $\boldsymbol{\delta}$ which minimizes the expectation of $x_i (i = 2, \cdots, n+1)$, which made the standard accuracy smaller. In fact, the expected accuracy is monotonically increasing about each $\delta_i (i = 2, \cdots, n+1)$. Thus, choosing $\delta = (0, -\epsilon, \cdots, -\epsilon)$ can be the best adversarial attack vector for any $w > 0$. In this situation, this equals $x_i'(i = 2, \cdots, n+1) \sim \mathcal{N}(\eta - \epsilon, n)$. Therefore, similar to equation (20), we can derive the train accuracy which is

$$v(w) = p\Phi((w + (\eta - \epsilon)n)/\sqrt{n}) + (1-p)\Phi((-w + (\eta - \epsilon)n)/\sqrt{n}). \tag{24}$$

Here $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution. Now we only need to solve equation $dv/dw = 0$. Through simple computation, this derives that

$$p \exp(-(w_1 + (\eta - \epsilon)n)^2/2n)/\sqrt{2\pi n} = (1-p)\exp(-(w_1 - (\eta - \epsilon)n)^2/2n)/\sqrt{2\pi n}. \tag{25}$$

Solving this equation, we finally get the optimal value for w_1 to be

$$w_1^{AT} = \frac{\ln p - \ln(1-p)}{2(\eta - \epsilon)}. \tag{26}$$

Therefore, W_R^{AT} under the optimal value of w_1 is

$$W_R^{AT} = \frac{\ln p - \ln(1-p)}{2n(\eta - \epsilon)}. \tag{27}$$

□

A.4. Proof for Theorem 4.4

Proof. Define the expected clean accuracy function

$$u(w) = \frac{p}{\sqrt{n}}\Phi((w + \eta n)/\sqrt{n}) + \frac{(1-p)}{\sqrt{n}}\Phi((-w + \eta n)/\sqrt{n}), \tag{28}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution and $w \in \mathbb{R}$. The derivative is

$$\begin{aligned} \frac{\mathrm{d}u(w)}{\mathrm{d}w} &= \frac{p}{\sqrt{2\pi n}} e^{-(w+\eta n)^2/2n} - \frac{1-p}{\sqrt{2\pi n}} e^{-(w-\eta n)^2/2n} \\ &= \frac{1}{\sqrt{2\pi n}} e^{-(w^2+\eta^2 n^2)/n} \big(p e^{-w\eta} - (1-p)e^{w\eta}\big). \end{aligned} \tag{29}$$

From the theorem 4.2 for standard training, we know $u(w)$ has only one global minimum $w_1^* = \frac{\ln p - \ln(1-p)}{2\eta}$ for which $\frac{\mathrm{d}u(w)}{\mathrm{d}w}|_{w=w_1^*} = 0$. Thus, from (29) we know that $\frac{du(w)}{w} < 0$ if $w > w_1^*$ and $\frac{du(w)}{w} > 0$ if $w < w_1^*$.

In the SAM algorithm where we set $\lambda = 0$ and with ϵ given, we know that

$$w_1^{SAM} = \operatorname*{argmax}_{w} \min_{\delta\in[-\epsilon,\epsilon]} u(w+\delta). \tag{30}$$

It is easy for us to know that state that

$$\min_{\delta\in[-\epsilon,\epsilon]} u(w_1^{SAM}+\delta) = \min\big\{u(w_1^{SAM}-\epsilon), u(w_1^{SAM}+\epsilon)\big\}. \tag{31}$$

If $u(w_1^{SAM}-\epsilon) > u(w_1^{SAM}+\epsilon)$, then $w_1^{SAM}+\epsilon > w_1^*$. Since $\frac{\mathrm{d}u(w)}{\mathrm{d}w}$ is continuous and locally bounded, there exists $\delta_0 > 0$ such that $u(w_1^{SAM}-\epsilon-\delta_0) > u(w_1^{SAM}+\epsilon)$ and $u(w_1^{SAM}+\epsilon-\delta_0) > u(w_1^{SAM}+\epsilon)$ Thus, we have

$$\min\big\{u(w_1^{SAM}-\epsilon-\delta_0), u(w_1^{SAM}+\epsilon-\delta_0)\big\} > \min\big\{u(w_1^{SAM}-\epsilon), u(w_1^{SAM}+\epsilon)\big\}. \tag{32}$$

Therefore

$$\min_{\delta\in[-\epsilon,\epsilon]} u\big((w_1^{SAM}-\delta_0)+\delta\big) > \min_{\delta\in[-\epsilon,\epsilon]} u(w_1^{SAM}+\delta), \tag{33}$$

which means that w_1^{SAM} is not the optimal value we want. Similarly we can disprove that $u(w_1^{SAM}-\epsilon) < u(w_1^{SAM}+\epsilon)$. Thus, $u(w_1^{SAM}-\epsilon) = u(w_1^{SAM}+\epsilon)$.

From this, we know that

$$\int_{w_1^{SAM}-\epsilon}^{w_1^*} \frac{\mathrm{d}u(w)}{\mathrm{d}w}\mathrm{d}w = -\int_{w_1^*}^{w_1^{SAM}+\epsilon} \frac{\mathrm{d}u(w)}{\mathrm{d}w}\mathrm{d}w. \tag{34}$$

Using (29), we have

$$\begin{aligned}
&\int_{w_1^{SAM}-\epsilon}^{w_1^*} e^{-w^2}\big(pe^{-w\eta}-(1-p)e^{w\eta}\big)\mathrm{d}w \\
=&-\int_{w_1^*}^{w_1^{SAM}+\epsilon} e^{-w^2}\big(pe^{-w\eta}-(1-p)e^{w\eta}\big)\mathrm{d}w.
\end{aligned} \tag{35}$$

If $w_1^{SAM} \le w_1^*$, define $h = w_1^{SAM}+\epsilon-w_1^*$. Thus, $h \le \epsilon \le -w_1^{SAM}+\epsilon+w_1^*$. In (35), we have

$$\begin{aligned}
&\int_{-h}^{0} e^{-(w_1^*+w)^2}\big(pe^{-(w_1^*+w)\eta}-(1-p)e^{(w_1^*+w)\eta}\big)\mathrm{d}w \\
\le&\int_{w_1^{SAM}-\epsilon}^{w_1^*} e^{-w^2}\big(pe^{-w\eta}-(1-p)e^{w\eta}\big)\mathrm{d}w \\
=&-\int_{w_1^*}^{w_1^{SAM}+\epsilon} e^{-w^2}\big(pe^{-w\eta}-(1-p)e^{w\eta}\big)\mathrm{d}w \\
=&-\int_{0}^{h} e^{-(w_1^*+w)^2}\big(pe^{-(w_1^*+w)\eta}-(1-p)e^{(w_1^*+w)\eta}\big)\mathrm{d}w \\
=&\int_{-h}^{0} e^{-(w_1^*-w)^2}\big(-pe^{-(w_1^*-w)\eta}+(1-p)e^{(w_1^*-w)\eta}\big)\mathrm{d}w.
\end{aligned} \tag{36}$$

Since $w_1^* > 0$, we can know that $e^{-(w_1^*+w)^2} > e^{-(w_1^*-w)^2} > 0$ for $w \in [-h, 0)$.

For the function $r(v) := \big(pe^{-v\eta}-(1-p)e^{v\eta}\big)$ is monotonically decreasing and has one zero point w_1^*, thus $pe^{-(w_1^*+w)\eta} - (1-p)e^{(w_1^*+w)\eta} > 0$ and $-pe^{-(w_1^*-w)\eta}+(1-p)e^{(w_1^*-w)\eta} > 0$ for $w \in [-h, 0]$. And

$$\begin{aligned}
&pe^{-(w_1^*+w)\eta}-(1-p)e^{(w_1^*+w)\eta}-\big(-pe^{-(w_1^*-w)\eta}+(1-p)e^{(w_1^*-w)\eta}\big) \\
=&\big(pe^{-w_1^*\eta}-(1-p)e^{w_1^*\eta}\big)\big(e^{w\eta}+e^{-w\eta}\big) \\
=&0.
\end{aligned} \tag{37}$$

Therefore, $\forall w \in [-h, 0)$,

$$e^{-(w_1^*+w)^2}\big(pe^{-(w_1^*+w)\eta} - (1-p)e^{(w_1^*+w)\eta}\big) > e^{-(w_1^*-w)^2}\big(-pe^{-(w_1^*-w)\eta} + (1-p)e^{(w_1^*-w)\eta}\big). \tag{38}$$

Combining this with (36), we reach a contradiction.

Thus, $w_1^{SAM} > w_1^*$. Applying this to the definition of W_R^{SAM} and W_R^*, we can easily obtain the result that

$$W_R^{SAM} > W_R^*. \tag{39}$$

□

A.5. Proof for Theorem 4.5

Proof. We proceed our proof from (35). Since we have proven that $w_1^{SAM} > w_1^*$, we suppose that $h = w_1^* - w_1^{SAM} + \epsilon < \epsilon$. Therefore, we can derive from (35) and the definition of h that

$$\begin{aligned}0 = &\int_{-h}^{0} e^{-(w_1^*+w)^2}(pe^{-(w_1^*+w)\eta} - (1-p)e^{(w_1^*+w)\eta})\mathrm{d}w \\ &- \int_{-h}^{0} e^{-(w_1^*-w)^2}(-pe^{-(w_1^*-w)\eta} + (1-p)e^{(w_1^*-w)\eta})\mathrm{d}w \\ &- \int_{h}^{2\epsilon-h} e^{-(w_1^*+w)^2}(-pe^{-(w_1^*+w)\eta} + (1-p)e^{(w_1^*+w)\eta})\mathrm{d}w.\end{aligned} \tag{40}$$

Since we only focus on h, we consider omitting the $o(h)$ terms in the calculation. To be more specific, $o(w^2)$ term in the integral symbol $\int_{-h}^{0}$ can be omitted, and $o(w)$ or $o(h)$ term for w and h in the integral symbol $\int_{h}^{2\epsilon-2h}$ can also be omitted.[1]

Combined with the proof in (37) and the definition of w_1^*, and abandoning the high order terms, we can calculate the right-hand side as follows.

$$\begin{aligned}RHS = &\int_{-h}^{0} \big(e^{-(w_1^*+w)^2} - e^{-(w_1^*-w)^2}\big)\cdot\big(pe^{-(w_1^*+w)\eta} - (1-p)e^{(w_1^*+w)\eta}\big)\mathrm{d}w \\ &+ \int_{h}^{2\epsilon-h} e^{-(w_1^*+w)^2}\big(-pe^{-(w_1^*+w)\eta} + (1-p)e^{(w_1^*+w)\eta}\big)\mathrm{d}w \\ \approx &\int_{-h}^{0} e^{-(w_1^*)^2}\big(1 - 2w_1^*w - 1 - 2w_1^*w + o(w)\big)\cdot\big(pe^{-w_1^*\eta}(1-w\eta) - (1-p)e^{w_1^*\eta}(1+w\eta)\big)\mathrm{d}w \\ &- \int_{0}^{2\epsilon-2h} e^{-(w_1^*)^2}(1-2w_1^*h)\cdot\big(-pe^{-w_1^*\eta}(1-h\eta) + (1-p)e^{w_1^*\eta}(1+h\eta)\big)\mathrm{d}w \\ \approx &4e^{-(w_1^*)^2}w_1^*\int_{-h}^{0} w^2\eta\big(-pe^{-w_1^*\eta} - (1-p)e^{w_1^*\eta}\big)\mathrm{d}w - e^{-(w_1^*)^2}\int_{0}^{2\epsilon-2h}\eta h\big(pe^{-w_1^*\eta} + (1-p)e^{w_1^*\eta}\big)\mathrm{d}w \\ \approx &\frac{4}{3}e^{-(w_1^*)^2}w_1^*\big(pe^{-w_1^*\eta} + (1-p)e^{w_1^*\eta}\big)h^3 - 2e^{-(w_1^*)^2}(\epsilon-h)\eta h\big(pe^{-w_1^*\eta} + (1-p)e^{w_1^*\eta}\big) \\ \approx &\frac{2}{3}e^{-(w_1^*)^2}\big(pe^{-w_1^*\eta} + (1-p)e^{w_1^*\eta}\big)\eta\big(2w_1^*h^2 - 3(\epsilon-h)\big).\end{aligned} \tag{41}$$

Since $RHS = 0$, by solving the last equality in (41) we get that

$$\frac{\epsilon - h}{h} = \frac{2}{3}w_1^*h = o(1). \tag{42}$$

Thus, the calculation and the abandoning of high-order terms in the calculation above are valid. Since $h \to \epsilon$, we have

$$\epsilon - h = \frac{2}{3}w_1^*h^2 \approx \frac{2}{3}w_1^*\epsilon^2. \tag{43}$$

[1]The validity of abandoning these high order terms can be seen from the result of the calculation, which shows that $\frac{h-\epsilon}{h} \to 0$.

Therefore, we obtain that $w_1^{SAM} \approx w_1^* + \frac{2}{3} w_1^* \epsilon^2$.

Applying the definition of W_R^{SAM} and W_R^* to this result, we finally reach the conclusion that

$$W_R^{SAM} \approx W_R^* + \frac{2}{3} W_R^* \epsilon^2. \tag{44}$$

□

A.6. Proof for Theorem 5

Proof. When both methods derives the same robust feature weight W_R (with different perturbation strength ϵ_{SAM} and ϵ_{AT}), denoting the standard training optimal parameter $W_R^* = (\ln p - \ln(1-p))/2n\eta$, we have

$$W_R = W_R^{SAM} \approx W_R^*(1 + \frac{2}{3}\epsilon_{SAM}^2), \tag{45}$$

which is the result of (44), and

$$W_R = W_R^{AT} = \frac{\eta}{\eta - \epsilon_{AT}} w_R^*, \tag{46}$$

which is the result of (27) and (23). Thus, combining two equations, we have

$$\frac{\eta}{\eta - \epsilon_{AT}} \approx 1 + \frac{2}{3}\epsilon_{SAM}^2. \tag{47}$$

Solving this as an equation of η, we get relationship

$$\frac{2}{3}\epsilon_{SAM}^2 \epsilon_{AT} + \epsilon_{AT} \approx \frac{2}{3}\eta \epsilon_{SAM}^2. \tag{48}$$

By dividing both sides with $\epsilon_{AT}\epsilon_{SAM}^2$, the relation in the theorem can be simply derived. This ends the proof. □

B. Detailed Experimental Results

Overall experimental results for general robust accuracy are shown in Table 13.

Table 13. General robust accuracy on **CIFAR-10C** dataset.

Method	Natural Accuracy	Brightness	Fog	Frost	Gaussian Blur	Impulse Noise	Jpeg	Shot Noise	Snow	Speckle Noise
SGD	94.5	63.7	26.3	30.8	19.2	40.6	32.4	26.5	41.9	32.8
Adam	93.9	54.8	21.9	25.4	15.9	35.7	28.1	21.5	38.0	25.1
SAM ($\rho = 0.1$)	95.4	63.1	26.2	29.2	20.6	37.0	31.7	23.0	42.7	28.3
SAM ($\rho = 0.2$)	**95.5**	61.5	20.2	27.5	16.6	40.2	27.2	26.1	38.1	30.2
SAM ($\rho = 0.3$)	95.4	61.0	24.8	28.6	19.1	40.3	28.2	23.7	40.2	27.5
SAM ($\rho = 0.4$)	94.7	59.5	22.4	31.8	17.0	37.7	30.0	29.7	41.7	35.3
ASAM ($\rho = 0.3$)	95.4	**68.9**	**28.7**	**34.0**	18.2	**41.9**	**34.0**	25.9	**45.9**	32.7
ESAM ($\rho = 0.3$)	95.2	64.3	28.1	32.9	**20.9**	39.3	32.8	**30.0**	44.2	**34.1**
AT (ℓ_∞-$\epsilon = \frac{8}{255}$)	84.5	16.1	12.5	11.6	15.5	18.2	16.7	17.3	16.2	16.9
AT (ℓ_2-$\epsilon = \frac{128}{255}$)	89.2	25.8	14.6	18.7	21.5	25.3	25.4	26.3	25.2	25.4