

# SAFER: Sharpness Aware layer-selective Finetuning for Enhanced Robustness in vision transformers

Bhavna Gopal<sup>1</sup>, Huanrui Yang<sup>2</sup>, Mark Horton<sup>1</sup>, Yiran Chen<sup>1</sup>

<sup>1</sup>Duke University, Durham, NC, USA    <sup>2</sup>University of Arizona, Tucson, AZ, USA

{bhavna.gopal, mark.horton, yiran.chen}@duke.edu    huanruiyang@arizona.edu

## Abstract

*Vision transformers (ViTs) have become essential backbones in advanced computer vision applications and multi-modal foundation models. Despite their strengths, ViTs remain vulnerable to adversarial perturbations, comparable to or even exceeding the vulnerability of convolutional neural networks (CNNs). Furthermore, the large parameter count and complex architecture of ViTs make them particularly prone to adversarial overfitting, often compromising both clean and adversarial accuracy. This paper mitigates adversarial overfitting in ViTs through a novel, layer-selective fine-tuning approach: **SAFER**. Instead of optimizing the entire model, we identify and selectively fine-tune a small subset of layers most susceptible to overfitting, applying sharpness-aware minimization to these layers while freezing the rest of the model. Our method consistently enhances both clean and adversarial accuracy over baseline approaches. Typical improvements are around 5%, with some cases achieving gains as high as 20% across various ViT architectures and datasets.*

## 1. Introduction

Vision Transformers (ViTs) [7, 19, 28] have significantly advanced computer vision architecture design, achieving state-of-the-art performance across diverse tasks, including semantic segmentation [17], object detection [3], and image generation [23]. However, as ViTs see increased deployment in real-world applications, concerns regarding their robustness against adversarial attacks—small, carefully crafted modifications to input images that mislead the model [10, 26]—have become paramount. Initial optimism that ViTs might inherently offer greater robustness than convolutional neural networks (CNNs) was quickly tempered by stronger threat models, which revealed their vulnerability to adversarial attacks at levels comparable to, or even exceeding, those of CNNs [2]. To defend against adversarial examples, ViTs continue to rely heavily on adver-

sarial training (AT) [1, 6, 20], which incorporates adversarial examples into the training process to improve robustness. However, the large parameter counts and intricate architectures of ViTs exacerbate overfitting during adversarial training, limiting improvements in both adversarial robustness and clean data performance.

Recent methods, such as Attention Random Dropping (ARD) and Perturbation Random Masking (PRM) [21], seek to address these limitations and improve AT for transformers, though they often yield inconsistent results across various settings. A more systematic approach rooted in sharpness-aware minimization (SAM) [9] provides a theoretically grounded optimizer to mitigate overfitting during training. Although SAM has shown benefits for enhancing robustness without compromising clean performance (in both CNNs and ViTs [35]), work in this area remains relatively underexplored. Furthermore, integrating SAM into ViTs' complex adversarial training processes can still hinder convergence, ultimately diminishing performance and failing to address overfitting effectively.

Research advances in CNNs suggest that harnessing specific architectural properties within models can significantly improve robustness. Approaches such as RiFT [36], CLAT [11], and AutoLoRA [34] leverage metrics and heuristics to identify and selectively exploit architecture-specific features that are critical to model robustness. For instance, CLAT utilizes hidden feature-based robust criticality indices to pinpoint "critical" layers—those disproportionately contributing to adversarial vulnerability—and fine-tunes these layers to enhance robustness. Motivated by this insight and the challenges of performing adversarial training on the full ViT model, we hypothesize and demonstrate that ViTs, like CNNs, contain layers critical to learning the adversarial training objective effectively. Accordingly, this work aims to identify a select subset of ViT layers that can train and converge more effectively and smoothly than the full model, yet still contribute substantially to the model's overall robust generalization. By pinpointing and selectively fine-tuning these layers, we enable targeted ad-

adjustments that maximize robustness and improve model performance on both adversarial and clean data.

Unfortunately, transformers present unique challenges for critical layer identification, making CNN-based methods like RiFT and CLAT less effective, if not completely ineffective. First, transformers’ higher parameter counts exacerbate overfitting, necessitating more explicit regularization. Moreover, ViTs consist of diverse layer types—such as attentions, projections, and MLPs—that produce output distributions not directly comparable across layers, complicating the task of accurately identifying which layers are the most critical [29]. These challenges underscore the need for a transformer-specific approach capable of effectively harnessing critical layers and enhancing adversarial robustness.

To overcome these limitations and mitigate overfitting concerns, we introduce a novel metric for critical layer identification in transformers. As shown in Figure 1, we leverage insights from SAM [9] and introduce a sharpness-based metric that precisely identifies layers most prone to overfitting, enabling targeted regularization. Building on this foundation, we propose SAFER, an adaptation of the SAM framework that mitigates overfitting specifically within these critical layers while freezing the rest of the model, ultimately enhancing both adversarial robustness and clean accuracy.

Additionally, given the widespread adoption of Parameter Efficient Fine-Tuning (PEFT) techniques for transformers [12], we extend SAFER to PEFT methods such as LoRA [15] and DORA [18]. By incorporating our algorithm across these frameworks, we demonstrate that SAFER tuning consistently enhances the robustness of transformer models, making them well-suited for diverse applications. This integration underscores the adaptability and broad utility of our approach within the transformer landscape.

Our contributions are summarized as follows:

- We introduce a novel metric for transformers that accurately identifies layers prone to overfitting in the adversarial training process.
- We present SAFER, an advanced SAM-based fine-tuning algorithm designed to mitigate overfitting specifically within critical transformer layers, enhancing both adversarial robustness and clean accuracy.
- We demonstrate SAFER’s versatility through results across different training methods and its integration with PEFT frameworks, highlighting its robust performance across varied scenarios.

SAFER demonstrates consistent improvements in both adversarial and clean performance, with typical gains around 5% and peaks of up to 20%, achieving state-of-the-art robustness across a variety of models and datasets.

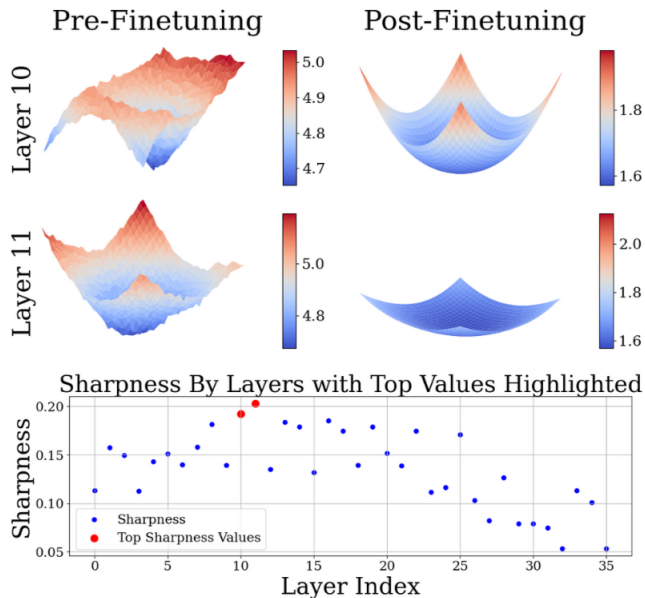


Figure 1. Sharpness value (Equ. (6)) measured over layers of an adversarially-trained DeiT-Tiny model (bottom), where the layers with top-2 values are selected for SAFER finetuning. The layers’ adversarial loss landscape before adversarial finetuning (left) and after SAFER finetuning (right) are visualized on the top.

## 2. Related Work

### 2.1. Adversarial training and overfitting

The concept of Adversarial Training (AT) emerged in response to the vulnerability of deep learning models to adversarial examples, first highlighted by Goodfellow et al. [10] in CNNs. By incorporating adversarial examples into the training process, Goodfellow et al. [10] demonstrated a significant improvement in model robustness. This idea was further developed into Projected Gradient Descent Adversarial Training (PGD-AT) [20], a minimax optimization approach that leverages multi-step PGD attacks on training data to enhance empirical robustness [1, 4, 6]. Later advancements, such as TRADES [33], introduced a loss function to balance accuracy and robustness, while other works further bolstered resilience using model ensembling and data augmentation [5, 30, 31]. Despite these improvements, techniques like PGD-AT and TRADES remain computationally demanding and prone to overfitting [25].

For transformers, adversarial training introduces even greater challenges. Transformers’ complex architectures and high parameter counts make them especially prone to overfitting, leading to difficulties with stability and convergence. Mo et al. [22] attempt to address these challenges by providing tailored guidelines and recipes to improve adversarial training for transformers, though the issue of overfitting remains persistent. The recent incorporation

of Sharpness-Aware Minimization (SAM) into adversarial training has shown promise, with Zhang et al. [35] demonstrating improved performance by smoothing the optimization landscape. However, even with SAM, transformers continue to struggle with catastrophic overfitting and convergence issues. Furthermore, these methods operate at a global level across the entire architecture, without leveraging transformer-specific architectural components.

## 2.2. Layer-selective adversarial training

Overfitting remains a central challenge in adversarial training. Subsequent efforts introduced methods specifically aimed at mitigating catastrophic overfitting, though these approaches were largely developed within the CNN space. RiFT [36] improved general performance by exploiting layer redundancies, though its reliance on heuristic-based redundancy measurements limited its adaptability. Zhang et al. [34] tackled overfitting by disentangling natural and adversarial objectives, yet model-wide adjustments constrained overall robustness. A more targeted solution, CLAT [11], employed theoretically grounded critical layer selection to fine-tune only a subset of layers, achieving notable gains in clean and adversarial robustness. Unlike global approaches, CLAT’s dynamic selection mechanism outperformed prior methods while remaining attack-agnostic, further suggesting that layer criticality could play an essential role in enhancing robustness. However, as our empirical results demonstrate (see Table 2), the criticality index selection in CLAT fails to capture essential layers in transformer architectures due to diverse layer types.

To overcome these limitations, we introduce SAFER—a novel SAM-based approach that uniquely identifies and selectively fine-tunes only the transformer layers most prone to overfitting. By targeting these critical layers, SAFER effectively addresses overfitting and delivers robust generalization, meeting the specific challenges of adversarial training in transformer architectures.

## 3. Methods

To address overfitting challenges in adversarial training for vision transformers, we first identify layers most susceptible to overfitting during training and then apply targeted layer-wise fine-tuning to reduce this tendency. In this section, we start by deriving metrics to identify overfitting layers, followed by a discussion of the training objective designed to mitigate overfitting in these layers.

### 3.1. Adversarial overfitting measurement

Given a neural network model, the extent of overfitting is determined by the weights it converges to, measurable through the sharpness of the loss landscape in the locality of the converged model. Previous work, Sharpness-Aware Minimization (SAM) [9], derived a theorem stating that the

generalization gap between the test loss  $L_D$  on a distribution  $\mathcal{D}$  and the model training loss  $L_S$  on a specific training set  $\mathcal{S}$  generated from  $\mathcal{D}$  is upper-bounded with high probability, for any  $\rho > 0$ , by

$$L_D(w) - L_S(w) \leq \max_{\|\epsilon\|_2 \leq \rho} L_S(w + \epsilon) - L_S(w) + h, \quad (1)$$

where  $w$  is the model weight and  $h$  is a strictly increasing function of  $\frac{\|w\|_2^2}{\rho^2}$ . The theory leads to an effective sharpness measurement

$$\gamma = \max_{\|\epsilon\|_2 \leq \rho} L_S(w + \epsilon) - L_S(w), \quad (2)$$

where a larger sharpness  $\gamma$  indicates greater susceptibility to overfitting, while a smaller  $\gamma$  reduces the generalization gap, as shown in SAM [9].

However, given the complexity of vision transformers and the optimization difficulty of the minimax adversarial training objective, directly applying SAM to the full model during adversarial training could impede effective convergence. To this end, we propose shifting away from training all layers together. Previous research not only identifies that a small subset of CNN layers are more prone to learning non-robust features [11, 36], but also provides insights into how to locate these layers. Inspired by this, we propose measuring each layer’s contribution to model overfitting to identify those layers that are most susceptible.

To establish a unified overfitting measure across layers of varying types and sizes, we isolate each layer’s contribution to overfitting by freezing the rest of the model. Specifically, we analyze the curvature of the loss landscape when fine-tuning each individual layer with adversarial training. For an adversarial training loss  $L_{adv}(w, x)$  defined as

$$L_{adv}(w, x) = \sup_{\|\delta\|_p \leq d} L(w, x + \delta), \quad (3)$$

where  $L$  is the clean training loss,  $x$  is the training data,  $d$  is the attack strength, and  $\delta$  is an adversarial perturbation achieved with some attack algorithm. We define the sharpness of the weight  $w_i$  in layer  $i$  based on Equ. (2) as

$$\gamma_i = \sum_{x \in \mathcal{B}} \max_{\|\epsilon\|_2 \leq \rho} L_{adv}(w_i + \epsilon, x) - L_{adv}(w_i, x), \quad (4)$$

where  $\mathcal{B}$  is a batch of training data.

To reduce the cost of explicitly solving the maximization problem in Equ. (4), we can further simplify the layer sharpness formulation by applying first-order Taylor expansion as

$$L_{adv}(w_i + \epsilon, x) \approx L_{adv}(w_i, x) + \epsilon^T \frac{\partial L_{adv}(w_i, x)}{\partial w_i}, \quad (5)$$

which yields a simplified sharpness measure as

$$\gamma_i \approx \sum_{x \in \mathcal{B}} \max_{\|\epsilon\|_2 \leq \rho} \epsilon^T \frac{\partial L_{adv}(w_i, x)}{\partial w_i} \propto \sum_{x \in \mathcal{B}} \left\| \frac{\partial L_{adv}(w_i, x)}{\partial w_i} \right\|_2. \quad (6)$$

We compute sharpness measures  $\gamma_i$  for all layers in the model using a single backward pass with the adversarial training loss as in Equ. (3). Note that all layers can be measured with the same batch of adversarial examples, requiring minimal computational overhead compared to the adversarial training process.

We rank each layer’s sharpness and select the top-K layers—those most prone to overfitting—for further fine-tuning, as described in the following section.

### 3.2. Layer-specific finetuning

To address adversarial overfitting in the selected top-K layers, we apply sharpness-aware minimization (SAM) to these layers only, keeping the remaining layers frozen. Freezing all other layers reduces optimization complexity and facilitates smoother model convergence. Based on the formulation in [9], we convert the adversarial training objective in Equ. (3) into

$$\min_{w_i} \sum_{x \in \mathcal{S}} \sup_{\|\delta\|_p \leq d} L(w_i + \epsilon, x + \delta), \quad (7)$$

where  $\epsilon = \rho \nabla_{w_i} L_{adv}(w, x) / \|\nabla_{w_i} L_{adv}(w, x)\|_2$  and  $\mathcal{S}$  is the training set. In adversarial training, note that perturbing the weight by  $\epsilon$  may result in the optimal adversarial example for weight  $w_i + \epsilon$  differing from that for weight  $w_i$ . To save computation time (from computing adversarial samples twice) and because the weight perturbation  $\epsilon$  is small, we use the adversarial sample computed on the unperturbed weight to compute the SAM loss in Equ. (7). Compared to standard adversarial training, this approach requires one additional backpropagation per optimization step to compute the weight perturbation  $\epsilon$  for SAM. However, this added overhead is acceptable given the multi-step optimization required to optimize the adversarial example during each adversarial training step.

Overall, the training method for SAFER is designed to systematically address overfitting in adversarial training by selectively focusing on layers most prone to it. Initially, we conduct standard adversarial training for some epochs, allowing all layers to learn meaningful features and progress toward a stable minimum. We then initiate an iterative process: identifying the top-K sharpest layers, fine-tuning these selected layers with the SAFER objective while freezing the remaining layers for several epochs, and periodically repeating sharpness measurements to update our selection of layers most susceptible to overfitting. This approach continuously targets the fine-tuning effort toward those layers most vulnerable to overfitting. The number of epochs for

the initial adversarial training, along with the interval between each round of sharpness measurement and layer selection, are SAFER hyperparameters, which we explore in ablation studies in Sec. 4.3.

## 4. Experiments

### 4.1. Experimental Settings

**Datasets and models** We conducted experiments on three widely recognized image classification datasets: CIFAR-10, CIFAR-100, and Imagenette. CIFAR-10 and CIFAR-100 consist of 60,000 color images at a resolution of 32×32 pixels, divided into 10 and 100 classes, respectively. For Imagenette, a 10-class subset of ImageNet-1K, we chose version v1 over the latest (v2) to avoid potential data leakage caused by class reshuffling in v2. Imagenette-v1 preserves a clear separation between training and validation sets, ensuring a more reliable evaluation [14, 16, 22].

For our experiments, we deployed a suite of network architectures across varying sizes, including ViT [7], DeiT [28], ConViT [8], and Swin Transformers [19]. We closely follow the settings outlined in Mo et al. [22] aligning with best practices for evaluating adversarial robustness in transformers. Additionally, to ensure reliable robustness measurements, each experiment was conducted at least 10 times, with the lowest observed accuracies reported.

**Training and Evaluation** For all experiments, except those using exclusively standard PGD-AT or SAFER throughout training, models were first pretrained on clean data for fewer than 10 epochs, followed by adversarial training with PGD-AT for 50 epochs. In experiments where either standard PGD-AT or SAFER was applied across all epochs, models were trained without any clean data pre-training. Since SAFER can be layered over different adversarial training methods, results incorporating SAFER are denoted in our tables as “X + SAFER,” where “X” refers to the baseline method applied prior to SAFER finetuning.

During adversarial training, we generated adversarial examples using PGD with a random start [20], setting an attack budget of  $\epsilon = 0.03$  under the  $\ell_\infty$  norm, a step size of  $\alpha = 0.007$ , and using 20 attack steps. For PGD-AT, we employed either the SAM or SGD optimizer during training, with the optimizer indicated in brackets, such as PGD-AT (SGD). These same settings were maintained for PGD-based attack evaluations.

To further assess robustness, we conducted evaluations using AutoAttack, a comprehensive ensemble-based method that combines multiple attack types, including two PGD variants, the FAB attack, and Square Attack [6].

Unless otherwise noted, these training and evaluation settings remained consistent across all experiments.

All experiments were conducted on NVIDIA RTX

A5000 GPUs. Fine-tuning began with an initial learning rate of 0.015, following the decay schedule from Foret et al. [9], but with a modified decay factor, reduced from 5 to 2. Standard data augmentations, including random cropping with padding and random horizontal flipping, were applied.

**SAFER settings** Equ. (6) shows the computation for layer sharpness that guides our layer selection process. In customizing the SAFER methodology for different network sizes, we designate approximately 5% of layers as critical, based on hyperparameter optimization. To refine our approach adaptively, we dynamically adjust the selected layers for fine-tuning by recalculating sharpness every ten epochs.

## 4.2. Comparative Performance

### 4.2.1. Whitebox Robustness

Table 1 demonstrates SAFER’s effectiveness in enhancing both clean and adversarial accuracies across diverse transformer architectures and attack types in a white-box setting. The results underscore SAFER’s robust performance across datasets and training methods, including over SOTA techniques (for example, ARD + PRM [21] and Tian et al. [27]) from RobustBench. The consistency across different architectures and model sizes further illustrates SAFER’s scalability, even in larger models. Notably, models trained with SAFER retain robustness against Auto Attacks, despite having been trained solely on PGD attacks. This resilience suggests that SAFER further reduces the overfitting on the specific attack model is trained on.

As we utilize SAM techniques in SAFER optimization, we highlight the comparison between SAFER and naively applying SAM in the PGD-AT optimization process. As shown in Table 1, though SAM helps the clean accuracy and robustness over SGD in PGD-AT, the performance improvement is limited due to the complexity in convergence, such that SAM becomes less effective on larger models. In comparison, SAFER resolves the convergence complexity by only targeting the layers that are the most prone to overfitting, significantly enhancing the final performance.

Due to space limitations, only PGD-20 and AA are used for robustness evaluation in Table 1. We show the validity of our attack convergence and the consistency of SAFER’s robustness across different attacks in Appendix D.

### Benchmarking SAFER Against Existing Layer-selective Fine-Tuning Approaches

Furthermore, we highlight the limitations of existing layer-selective fine-tuning techniques, like CLAT [11] and RiFT [36], when applied to vision transformers. As shown in Tab. 2, while previous methods improve adversarial accuracy in CNN models such as Wide ResNet-50 and ResNet-18, they noticeably reduce adversarial accuracy in ViTs. This suggests that the feature-based layer selection metrics designed for CNNs are not ef-

fective for ViTs due to the diverse layer types and complicated architectures. In contrast, SAFER demonstrates superior performance, clearly outperforming these techniques.

### 4.2.2. Blackbox Robustness

In addition to white-box results, Tables 3 and 4 provide black-box robustness evaluations (Auto Attack and PGD, respectively), comparing models trained solely with PGD-AT versus those enhanced with SAFER, using the same attack settings as in white-box evaluations.

As a sanity check, the higher accuracies observed in black-box settings compared to white-box settings indicate that gradient masking is not present in models using SAFER, confirming the reliability of our white-box robustness evaluation.

In line with white-box results, models trained with SAFER also outperform those trained solely with PGD-AT in black-box settings, demonstrating greater resilience across both black-box and white-box settings, regardless of attack method or model architecture. These findings underscore SAFER as a robust, adaptable solution for adversarial training across diverse configurations and attack scenarios.

### 4.2.3. PEFT methods

With the recent advancements in large language models, PEFT methods such as LoRA [15] and DORA [18] have become frequently used approaches for updating weights when tuning large transformer models. To evaluate the generalizability of our method, we provide results for models trained using SAFER and compare them with models trained with PGD-AT, using both SGD and SAM optimizers. PEFT methods are applied for weight updates. The results, presented in Table 5, reveal a consistent performance trend similar to that of full fine-tuning, where SAFER consistently outperforms across different models, datasets, and PEFT techniques.

## 4.3. Ablation Studies

### 4.3.1. Pretraining epochs

As discussed in Sec. 3, we apply SAFER after an initial phase of adversarial training. Here, we examine how varying the number of pretraining epochs affects performance. Figure 2 presents training curves for different allocations of PGD pretraining and fine-tuning epochs within a 70-epoch training budget. In models trained solely with PGD-AT, even with SAM optimizer being applied, both clean and adversarial accuracy graphs reveal limitations. In the clean accuracy graph, PGD-AT accuracy plateaus and then declines slightly, signaling overfitting, as documented in previous research [24]. By contrast, SAFER continues to improve clean accuracy, effectively mitigating this overfitting issue. In the adversarial accuracy graph, PGD-AT performance falls significantly short of SAFER’s, highlighting a notable gap in robustness. Notably, incorporating SAFER

Table 1. Consolidated Performance Comparison of Various Models on CIFAR-10, CIFAR-100, and Imagenette with Clean, PGD-20, and Auto Attack (AA) Accuracy.

Model	Method	CIFAR-10			CIFAR-100			IMAGENETTE		
		CLEAN	PGD-20	AA	CLEAN	PGD-20	AA	CLEAN	PGD-20	AA
<b>DeiT-Ti [28]</b>	PGD-AT (SGD)	75.46	48.10	43.62	53.11	27.97	25.45	80.60	56.00	53.80
	PGD-AT (SAM)	77.12	54.45	45.12	56.88	32.83	32.00	83.91	65.42	65.20
	ARD + PRM	79.60	50.33	45.99	54.67	30.67	30.02	90.40	65.00	64.00
	TIAN ET AL. [27]	75.50	46.33	42.10	52.67	27.45	23.33	82.88	57.24	54.89
	[27] + SAFER	77.21	48.05	44.67	54.04	29.31	28.99	83.79	50.80	56.38
	<b>SAFER</b>	<b>82.36</b>	<b>68.50</b>	<b>50.12</b>	<b>62.37</b>	<b>40.15</b>	<b>35.65</b>	<b>92.45</b>	<b>68.36</b>	<b>67.88</b>
<b>DeiT-S</b>	PGD-AT (SGD)	81.43	51.88	47.10	55.36	29.12	27.88	92.20	64.60	63.40
	PGD-AT (SAM)	82.10	53.64	46.55	58.81	35.44	31.03	92.00	67.09	65.12
	ARD + PRM	83.04	52.52	48.34	58.45	30.13	28.15	91.00	66.60	65.80
	<b>SAFER</b>	<b>86.01</b>	<b>70.26</b>	<b>51.58</b>	<b>63.66</b>	<b>42.29</b>	<b>36.70</b>	<b>94.78</b>	<b>69.86</b>	<b>67.20</b>
<b>ViT-S [7]</b>	PGD-AT (SGD)	79.59	50.86	46.37	55.01	27.45	23.21	90.40	63.80	62.80
	PGD-AT (SAM)	80.11	52.10	48.11	56.45	29.30	25.52	90.00	65.11	62.13
	ARD + PRM	81.86	51.73	47.33	58.55	30.21	24.46	91.40	65.20	63.00
	<b>SAFER</b>	<b>83.40</b>	<b>68.89</b>	<b>50.12</b>	<b>60.24</b>	<b>32.56</b>	<b>25.50</b>	<b>94.22</b>	<b>67.01</b>	<b>64.57</b>
<b>ViT-B</b>	PGD-AT (SGD)	83.16	52.98	49.06	55.22	29.31	24.45	93.40	68.80	67.00
	PGD-AT (SAM)	84.45	53.63	51.84	57.18	30.07	26.42	94.58	69.47	68.12
	ARD + PRM	84.90	53.80	50.03	59.80	31.24	27.12	95.00	70.00	69.60
	<b>SAFER</b>	<b>86.12</b>	<b>71.95</b>	<b>53.51</b>	<b>61.55</b>	<b>33.19</b>	<b>29.89</b>	<b>97.65</b>	<b>72.43</b>	<b>70.34</b>
<b>ConViT-Ti [8]</b>	PGD-AT (SGD)	53.09	33.63	29.65	40.45	20.22	19.83	63.60	39.20	36.60
	ARD + PRM	80.28	47.47	45.42	55.64	26.67	26.60	90.40	65.00	64.40
	<b>SAFER</b>	<b>83.45</b>	<b>51.72</b>	<b>48.19</b>	<b>59.10</b>	<b>28.34</b>	<b>28.20</b>	<b>92.28</b>	<b>68.07</b>	<b>66.97</b>
<b>ConViT-S</b>	PGD-AT (SGD)	54.03	34.61	30.60	44.75	20.23	22.12	87.40	64.20	61.60
	PGD-AT (SAM)	55.92	38.65	45.02	46.12	21.10	23.04	88.05	65.03	65.58
	ARD + PRM	84.32	53.10	48.85	58.32	25.33	27.99	94.40	68.20	67.60
	<b>SAFER</b>	<b>87.34</b>	<b>56.55</b>	<b>50.00</b>	<b>61.01</b>	<b>30.45</b>	<b>29.06</b>	<b>96.21</b>	<b>71.43</b>	<b>69.31</b>
<b>ConViT-B</b>	PGD-AT (SGD)	61.54	38.77	34.21	45.51	27.68	25.55	92.20	68.20	68.00
	ARD + PRM	85.80	53.36	49.33	59.89	30.32	28.81	95.20	73.00	70.60
	<b>SAFER</b>	<b>88.91</b>	<b>56.21</b>	<b>51.51</b>	<b>62.23</b>	<b>31.42</b>	<b>30.02</b>	<b>96.22</b>	<b>77.02</b>	<b>72.24</b>
<b>Swin-Ti [19]</b>	PGD-AT (SGD)	79.34	47.95	45.98	56.46	28.42	22.10	94.80	72.80	71.80
	ARD + PRM	82.63	48.87	45.31	58.12	30.32	24.35	96.20	74.40	71.20
	<b>SAFER</b>	<b>84.15</b>	<b>50.99</b>	<b>49.01</b>	<b>60.11</b>	<b>31.45</b>	<b>25.51</b>	<b>97.33</b>	<b>75.45</b>	<b>72.69</b>
<b>Swin-S</b>	PGD-AT (SGD)	79.34	48.53	44.88	57.89	29.39	23.94	95.40	74.00	73.80
	PGD-AT (SAM)	82.95	51.60	47.25	58.40	30.05	25.00	95.85	74.65	76.10
	ARD + PRM	84.46	50.02	46.17	59.12	30.23	25.21	96.00	75.00	74.80
	<b>SAFER</b>	<b>86.52</b>	<b>52.00</b>	<b>50.10</b>	<b>61.78</b>	<b>31.97</b>	<b>26.40</b>	<b>97.10</b>	<b>78.76</b>	<b>78.00</b>
<b>Swin-B</b>	PGD-AT (SGD)	83.36	50.19	46.89	57.88	29.11	24.64	96.40	75.80	74.60
	ARD + PRM	84.16	51.47	47.50	59.94	30.34	26.00	97.20	77.40	76.20
	TIAN ET AL. [27]	84.50	52.42	50.10	58.67	29.34	25.98	89.88	75.45	71.18
	[27] + SAFER	85.55	53.01	51.11	59.78	31.00	27.22	91.48	76.97	73.20
	<b>SAFER</b>	<b>86.78</b>	<b>53.65</b>	<b>52.00</b>	<b>61.20</b>	<b>32.91</b>	<b>27.13</b>	<b>98.45</b>	<b>80.13</b>	<b>77.68</b>

Table 2. PGD-20 Adversarial Accuracies on CIFAR-10 for Different Training/Finetuning Methods with Differences from SAFER

Model	PGD-AT (SGD)	RIFT	CLAT	SAFER	SAFER - RIFT $\Delta$	SAFER - CLAT $\Delta$
<b>WRN-34-10 [32]</b>	57.40	55.01	57.11	<b>59.02</b>	+4.01	+1.91
<b>RN-18 [13]</b>	53.63	54.65	55.37	<b>56.98</b>	+2.33	+1.61
<b>DeiT-Ti</b>	48.10	44.32	46.29	<b>68.50</b>	+24.18	+22.21
<b>ViT-S</b>	50.86	45.93	46.12	<b>68.89</b>	+22.96	+22.77

at any stage of training results in higher clean accuracy and robustness at convergence.

We also include results for applying SAFER from

scratch (0 epochs of pretraining) with some interesting observations. Although the model trained exclusively with SAFER eventually achieves competitive performance with

Table 3. Comparative Analysis of Black-box Auto Attack Accuracy on CIFAR-10 and Imagenette. Each row is the attacker, and each column is the victim.

Network	Method	CIFAR-10 Adv. Acc. (%)				IMAGENETTE Adv. Acc. (%)			
		DeiT-Ti	DeiT-S	ViT-S	Swin-B	DeiT-Ti	DeiT-S	ViT-S	Swin-B
<b>DeiT-Ti</b>	PGD-AT (SGD)	-	50.22	50.15	53.22	-	70.45	69.21	80.21
	SAFER	-	54.45	53.86	57.89	-	74.89	72.13	84.93
<b>DeiT-S</b>	PGD-AT (SGD)	50.71	-	49.83	51.22	67.67	-	70.45	83.48
	SAFER	52.28	-	55.01	55.43	71.50	-	73.68	86.72
<b>ViT-S</b>	PGD-AT (SGD)	50.13	50.67	-	50.41	68.89	71.28	-	83.91
	SAFER	53.91	55.16	-	54.82	70.23	75.97	-	87.43
<b>Swin-B</b>	PGD-AT	52.01	52.45	54.03	-	68.23	76.00	74.88	-
	SAFER	55.91	57.01	56.65	-	72.45	79.21	76.53	-

Table 4. Comparative Analysis of Black-box PGD-20 Accuracy on CIFAR-10 and Imagenette. Each row is the attacker, and each column is the victim.

Network	Method	CIFAR-10 Adv. Acc. (%)				IMAGENETTE Adv. Acc. (%)			
		DeiT-Ti	DeiT-S	ViT-S	Swin-B	DeiT-Ti	DeiT-S	ViT-S	Swin-B
<b>DeiT-Ti</b>	PGD-AT (SAM)	-	57.50	60.30	54.10	-	69.70	70.01	81.32
	SAFER	-	74.60	68.40	58.20	-	72.53	72.24	84.45
<b>DeiT-S</b>	PGD-AT (SAM)	54.50	-	59.00	53.50	70.20	-	71.45	80.73
	SAFER	71.80	-	64.23	57.80	70.88	-	73.89	85.01
<b>ViT-S</b>	PGD-AT (SAM)	55.50	58.00	-	55.80	68.45	71.48	-	79.45
	SAFER	73.98	76.50	-	60.20	71.87	73.96	-	83.20
<b>Swin-B</b>	PGD-AT (SAM)	58.10	59.80	64.30	-	72.76	74.59	74.88	-
	SAFER	76.21	78.90	68.50	-	74.53	75.82	76.92	-

Table 5. Effect of SAFER on PEFT Models (LoRA/DoRA) for CIFAR-10 and Imagenette with PGD-20 Adversarial Accuracy.

Network	Method	CIFAR-10		IMAGENETTE	
		Clean	Adv. Acc.	Clean	Adv. Acc.
<b>DeiT-Ti</b>	LORA PGD-AT (SGD)	57.01	40.23	58.30	41.10
	LORA PGD-AT (SAM)	71.45	51.45	73.20	52.30
	LORA SAFER	<b>78.12</b>	<b>63.50</b>	<b>80.50</b>	<b>64.20</b>
	DORA PGD-AT (SGD)	59.65	42.86	60.10	43.00
	DORA PGD-AT (SAM)	73.30	53.82	74.00	54.10
	DORA SAFER	<b>80.18</b>	<b>65.55</b>	<b>81.10</b>	<b>65.80</b>
<b>ViT-B</b>	LORA PGD-AT (SGD)	75.50	60.70	77.00	62.10
	LORA PGD-AT (SAM)	78.80	65.10	81.30	66.80
	LORA SAFER	<b>81.40</b>	<b>70.20</b>	<b>84.90</b>	<b>69.10</b>
	DORA PGD-AT (SGD)	76.80	61.50	78.50	63.00
	DORA PGD-AT (SAM)	80.90	66.10	82.20	68.00
	DORA SAFER	<b>81.99</b>	<b>71.30</b>	<b>85.30</b>	<b>71.20</b>
<b>ViT-L</b>	PGD-AT (SGD)	85.30	55.41	55.21	30.15
	PGD-AT (SAM)	87.46	68.58	62.39	39.28
	SAFER	<b>89.91</b>	<b>70.42</b>	<b>65.33</b>	<b>41.37</b>
	LORA PGD-AT (SGD)	79.16	48.84	47.06	26.53
	LORA PGD-AT (SAM)	85.41	67.22	62.09	39.23
	LORA SAFER	<b>86.76</b>	<b>68.42</b>	<b>63.18</b>	<b>40.31</b>
	DORA PGD-AT (SGD)	80.78	50.23	50.30	28.54
	DORA PGD-AT (SAM)	87.13	68.34	63.52	39.81
	DORA SAFER	<b>88.63</b>	<b>69.92</b>	<b>64.83</b>	<b>41.24</b>

PGD-AT, its convergence is significantly slower. This suggests that full model training is beneficial in the early stages for rapid convergence, while layer-selective training helps mitigate overfitting and boosts model performance in later stages.

### 4.3.2. Layer Selection

SAFER’s effectiveness hinges on selecting the layers most prone to overfitting for fine-tuning, while keeping the remaining layers frozen. To verify the importance of this selection, we compare SAFER with an alternative approach where randomly chosen layers are dynamically fine-tuned instead of those identified by our method. Results of this comparison are presented in Table 6, demonstrating that targeting the layers identified by SAFER significantly improves both adversarial robustness and clean accuracy. Furthermore, as shown in Appendix A, certain layers are consistently selected within the same model architecture, even across diverse datasets. This consistency suggests that some layers possess inherent properties that predispose them to overfit, reinforcing the importance of accurately identifying and targeting these layers for SAFER to be effective.

Additional ablation studies in the Appendix B and C examine two key factors in SAFER’s performance: frequency of dynamic layer selection and the number of layers cho-

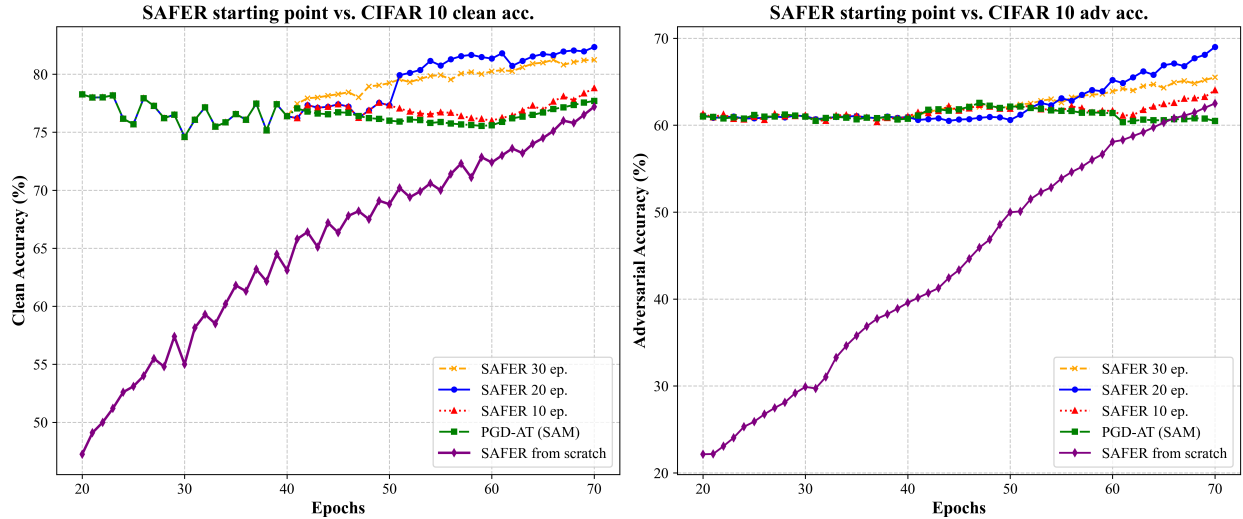


Figure 2. SAFER performance at different starting points on CIFAR-10 with DeiT-Ti: clean (left) vs. adversarial (right) accuracies.

Table 6. PGD-20 and Auto Attack (AA) adversarial accuracies for models with sharpness-selected vs. randomly selected layers for fine-tuning in SAFER.

Network	Sharpness-Selected Layers			Randomly Selected Layers		
	Clean Acc.	PGD-20	AA	Clean Acc.	PGD-20	AA
DeiT-Ti	82.36	68.50	50.12	79.15	65.46	45.38
ViT-S	83.40	68.89	50.12	80.19	64.35	46.19
Swin-B	86.78	53.65	52.00	81.73	49.72	44.38

sen for fine-tuning. Results indicate that dynamically re-evaluating layers for fine-tuning is crucial, as new overfitting layers can emerge once previously selected layers shift away from overfitting. Additionally, the number of layers selected is crucial, as choosing too many layers complicates the optimization process and can result in poorer performance. The hyperparameters used in our main experiments are informed by these observations.

#### 4.4. Overhead analysis

Lastly, we show that the time required to determine layer sharpness is negligible within the overall adversarial training process. As shown in Tab. 7, layer sharpness ranking can be reliably computed with a batch size as small as 50, with top-ranking layers consistent across larger batches. We conducted over 1,000 runs per network, randomly selecting data for sharpness estimation, and found remarkable consistency in the computed layer rankings. This stability allows us to use only **0.001%** of the training data for sharpness estimation every 10 epochs, adding just **0.2%** extra time to the standard adversarial training process, as shown in Tab. 7. For reference, one epoch of DeiT-Ti PGD-AT with SGD on CIFAR-10 takes approximately 290 seconds. Although an additional backpropagation step is required for SAM compared to SGD, the overhead of using SAM-based SAFER remains minimal. A SAFER fine-tuning epoch takes 298

Table 7. DeiT-Ti fine-tuning layers selected by SAFER with varying data amounts and corresponding computation times.

BATCH SIZE	CIFAR-10		IMAGENETTE	
	CRITICAL LAYERS	TIME (S)	CRITICAL LAYERS	TIME (S)
50	11, 10, 13, 8, 16	7.20	11, 10, 13, 8, 16	9.13
100	11, 10, 13, 8, 16	8.50	11, 10, 13, 9, 16	10.20
200	11, 10, 13, 8, 14	13.00	11, 10, 13, 8, 16	15.45
300	11, 10, 13, 8, 16	16.15	11, 10, 13, 9, 16	17.60
500	11, 10, 16, 13, 14	17.34	11, 10, 13, 8, 16	19.50

seconds, adding only  $\sim 3\%$  more time to SGD. This is because computing adversarial examples, which requires multiple backpropagations, takes up the majority of the time.

## 5. Conclusions

This work introduces SAFER, a layer-selective fine-tuning framework for Vision Transformers (ViTs) that addresses adversarial overfitting by selectively refining layers identified as most prone to adversarial overfitting using sharpness-aware minimization (SAM). Our results show that fine-tuning a limited subset of layers achieves notable improvements in both clean and adversarial accuracy across various architectures and baseline adversarial training methods. Additionally, we demonstrate that SAFER integrates effectively with Parameter-Efficient Fine-Tuning (PEFT) approaches, underscoring its versatility in transformer-based models. We limit the scope of this work to enhancing empirical robustness in ViTs. Open questions remain regarding why certain layers in ViTs become prone to overfitting, how they might be identified with greater precision, and whether architectural or training modifications could further improve robustness. A deeper theoretical exploration of these questions is left for future work.

## References

- [1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018. 1, 2
- [2] Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than cnns?, 2021. 1
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1
- [4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. 2
- [5] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness, 2022. 2
- [6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020. 1, 2, 4
- [7] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 4, 6
- [8] Stéphane d’Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning*, pages 2286–2296. PMLR, 2021. 4, 6
- [9] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021. 1, 2, 3, 4, 5
- [10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 1, 2
- [11] Bhavna Gopal, Huanrui Yang, Jingyang Zhang, Mark Horton, and Yiran Chen. Criticality leveraged adversarial training (clat) for boosted performance via parameter efficiency, 2024. 1, 3, 5
- [12] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. 2
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6
- [14] Jeremy Howard. Imagewang. 4
- [15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 2, 5
- [16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 4
- [17] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 1
- [18] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024. 2, 5
- [19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 4, 6
- [20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 1, 2, 4
- [21] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture, 2022. 1, 5
- [22] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. *Advances in Neural Information Processing Systems*, 35:18599–18611, 2022. 2, 4
- [23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1
- [24] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020. 5
- [25] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free!, 2019. 2
- [26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. 1
- [27] Rui Tian, Zuxuan Wu, Qi Dai, Han Hu, and Yu-Gang Jiang. Deeper insights into the robustness of vits towards common corruptions. *arXiv preprint arXiv:2204.12143*, 2022. 5, 6
- [28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 1, 4, 6
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 2
- [30] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V. Le. Adversarial examples improve image recognition, 2020. 2
- [31] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33: 5505–5515, 2020. 2
- [32] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. 6
- [33] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy, 2019. 2

- [34] Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning, 2024. [1](#), [3](#)
- [35] Yihao Zhang, Hangzhou He, Jingyu Zhu, Huanran Chen, Yifei Wang, and Zeming Wei. On the duality between sharpness-aware minimization and adversarial training. *arXiv preprint arXiv:2402.15152*, 2024. [1](#), [3](#)
- [36] Kaijie Zhu, Jindong Wang, Xixu Hu, Xing Xie, and Ge Yang. Improving generalization of adversarial training via robust critical fine-tuning, 2023. [1](#), [3](#), [5](#)

## Acknowledgements

This work was made possible through the support of NSF Grant No. 2112562 and ARO Grant No. W911NF-23-2-0224.