Sparse Fine-Tuning for Parameter-Efficient Adversarial Training

Zhaoxin Wang
School of Artificial Intelligence
Xidian University
Xi'an, China

Zihang Ding
School of Artificial Intelligence
Xidian University
Xi'an, China

Handing Wang*
School of Artificial Intelligence
Xidian University
Xi'an, China

Abstract

Deep neural networks achieve strong performance across many tasks but remain vulnerable to adversarial perturbations. Adversarial training (AT) is one of the most effective defenses, yet it suffers from high memory cost on large models and the persistent clean-robust accuracy trade-off. Recent work introduces parameter-efficient fine-tuning, such as LoRA, into AT, but this constraint induces a notable robustness gap relative to full-parameter AT. In this work, we revisit parameter-efficient adversarial fine-tuning from a parameter space perspective. Through gradient analysis, we find that adversarial optimization is highly concentrated on a small subset of parameters, which we call Robustness-Critical Parameters (RCPs), suggesting that robustness is encoded in a sparse subspace. Building on this observation, we propose Sparse Adversarial Fine-Tuning (SAFT), which identifies RCPs via adversarial gradient saliency and performs AT by updating only these parameters while freezing the rest. Across architectures and datasets, SAFT outperforms LoRA-based methods in both clean and robust accuracy with fewer trainable parameters, and approaches or surpasses full parameter AT in robustness while updating only about 5% of parameters.

1 Introduction

Deep learning has achieved excellent performance across a wide range of tasks [26], such as image classification [16], semantic segmentation [14], and object detection [23]. However, deep learning models are vulnerable to adversarial attacks, in which imperceptible perturbations added to the input can lead to wrong predictions [35]. To improve the model's robustness against these attacks, one of the most effective defenses is adversarial training (AT) [30]. AT can be formulated as a bi-level optimization problem, where the inner maximization generates adversarial perturbations, typically using a multi-step procedure called projected gradient descent (PGD)[30], and the outer minimization optimizes the model parameters on the perturbed data.

Although AT significantly improves the robustness of the deep learning models, AT suffers from two longstanding issues. First, the bi-level optimization in AT requires multiple forward and backward passes per training step to generate adversarial perturbations, leading to substantially higher memory and time costs than standard training. Second, as adversarial examples and clean examples can be viewed as samples from two different distributions, AT induces a clean-robust accuracy trade-off that models trained to be robust often suffer a notable accuracy degradation on benign samples.

*Corresponding author. Email: hdwang@xidian.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To reduce the computational overhead of AT, one line accelerates the inner perturbation generation process [45, 34, 40, 49]. FastAT [43] replaces multi-step PGD with single-step FGSM, but it suffers from catastrophic overfitting, which subsequent studies seek to mitigate [1, 48, 22, 41]. While these methods reduce training time, they update all model parameters and leave the memory bottleneck unaddressed. To reduce memory cost on large transformer-based models, recent work introduces parameter-efficient fine-tuning (PEFT) into AT. LoFT [11], LNLoRA [44], and AdvLoRA [21] all build on LoRA [18] to limit trainable parameters to a low-rank subspace. However, LoRA-based PEFT-AT exhibits a persistent robustness gap relative to full-parameter AT [19, 5], suggesting that the low-rank constraint may be misaligned with the optimization directions needed for robustness.

Unlike these efficiency methods, another line of work targets the trade-off problems [46, 39, 42]. TRADES [46] reformulates the AT objective to balance clean and robust losses through a tunable regularizer. MART [38] reweights misclassified examples to better handle hard cases. Carmon et al. [4] leverage additional unlabeled data to push the trade-off frontier outward. While these methods improve the trade-off frontier, they all operate at the loss or data level and implicitly treat the trade-off as a property shared uniformly across all parameters. Whether the trade-off has structure in parameter space, and whether it can be mitigated by acting on that structure, remains largely unexplored.

In this work, we revisit AT from a parameter-space perspective and ask whether the two above issues share a common origin. Through gradient analysis, we observe that adversarial optimization is concentrated on a small subset of parameters. A small fraction of parameters, about 5%, carry the majority of the cumulative gradient mass, while over 90% remain near zero throughout training. These observations suggest that robustness is not uniformly distributed across the network but is encoded in a sparse subset of parameters, which we refer to as Robustness Critical Parameters (RCPs).

This sparsity has two important implications. From an efficiency perspective, full-parameter AT is over-parameterized for robustness, and updating only the RCPs should suffice to recover most of the robustness gain. From a trade-off perspective, if the adversarial signal lives in a sparse subspace of parameters, restricting updates to that subspace leaves all remaining pretrained weights unchanged. The two longstanding issues of AT may therefore be approached jointly, not by modifying the loss or expanding the data, but by acting on the parameter-space structure of robustness itself.

Building on this, we propose Sparse Adversarial Fine-Tuning (SAFT), a parameter-efficient adversarial fine-tuning framework that identifies RCPs via adversarial gradient saliency and performs adversarial fine-tuning by updating only these parameters while freezing the rest. The identification is performed once before fine-tuning and requires no additional learnable importance scores. SAFT is orthogonal to existing AT methods and can be easily integrated with different AT objectives. Our contribution can be summarised as:

- We identify a sparse parameter-space structure underlying adversarial robustness, where adversarial optimization is concentrated on a small subset of parameters, which we call RCPs. This concentration holds across datasets and on a convolutional backbone. The RCPs are highly consistent across different attack types, indicating that they reflect an intrinsic property of the model rather than an artifact of any specific attack.
- We propose SAFT, a sparse parameter-efficient adversarial fine-tuning framework that updates only the RCPs while freezing the rest of the model. By confining adversarial updates to the RCPs, SAFT leaves the remaining pretrained weights unchanged and directs robustness gains only to where they are needed. It reduces optimizer-state, persistent-gradient and peak training memory relative to full-parameter AT.
- Across architectures and datasets, SAFT outperforms LoRA-based adversarial fine-tuning and approaches or surpasses full-parameter adversarial fine-tuning in robust accuracy while updating only about 5% of parameters. The localization of robustness in the RCPs explains why SAFT improves the clean-robust trade-off frontier, offering a new parameter-space perspective. Control experiments show that RCPs are robustness-specific rather than generally important parameters.

2 Related Work

2.1 Efficient Adversarial Training

A major limitation of AT is the high computational cost incurred by multi-step adversarial example generation. Existing efficient AT methods mainly reduce the cost of the inner maximization.

FastAT [43] replaces multi-step PGD with single-step FGSM [12], but suffers from catastrophic overfitting. Subsequent methods improve the stability of single-step AT through gradient regularization or modified perturbation updates, such as GradAlign [1] and FGSM-PCO [41]. Other methods reduce computation by reusing gradients or perturbations. FreeAT [34] combines adversarial example generation and model update within the same backward pass. YOPO [45] avoids full network backpropagation during inner maximization by using an approximation based on the first layer.

Recently, PEFT has been introduced to reduce the memory cost of adversarial adaptation of transformer-based large models. LoRA-based methods restrict trainable parameters to low-rank adaptation modules. Yuan et al. [44] apply LoRA to most modules and introduce LayerNorm before LoRA modules to improve robustness, while AdvLoRA [21] improves LoRA-based adversarial adaptation through reparameterization. DoRA [28] decomposes each pretrained weight into a magnitude and a direction and applies the low-rank update to the direction.

2.2 Accuracy-Robustness Trade-off

The trade-off between clean accuracy and adversarial robustness is hard to handle in AT. Improving robustness against adversarial perturbations often comes at the cost of lower accuracy on benign examples. Prior studies have provided several explanations for this phenomenon. Tsipras et al. [36] show that robustness and accuracy can be inherently conflicting under certain data distributions. Ilyas et al. [20] attribute this conflict to non-robust but predictive features, which are useful for clean classification but sensitive to small perturbations. These studies suggest that the trade-off is closely related to the features learned by the model. To mitigate this issue, TRADES [46] introduces a regularized objective that balances natural accuracy and boundary robustness. MART [38] improves robust generalization by reweighting misclassified examples. Semi-supervised AT and robust self-training leverage additional unlabeled data to push the trade-off frontier outward [4]. Recent methods further explore invariance regularization, input-level reweighting, attention regularization, and label-space reformulation to reduce the conflict between clean and adversarial learning [42, 47].

2.3 Parameter-Space Sparsity in Robustness

Parameter selection has been widely studied as a way to improve training efficiency while preserving model performance. Selective PEFT methods update only a subset of pretrained parameters instead of introducing additional adapter modules. Diff pruning [13] learns a sparse mask over parameter differences to identify task-relevant updates. PaFi [27] fine-tunes parameters with small absolute magnitudes in pretrained models. LT-SFT [2] selects parameters that change the most during the first epoch of fine-tuning. Parameter sparsity is also closely related to pruning and subnetwork discovery. The lottery ticket hypothesis [10] shows that dense neural networks contain sparse subnetworks that can be trained in isolation to match the performance of the original network. A large body of work on structured pruning further shows that many parameters or components can be removed while retaining accuracy [6, 50, 37]. These studies suggest that model capabilities are not uniformly distributed across all parameters, but can be concentrated in a subset of critical components.

3 Empirical Findings

Most existing AT methods update all model parameters during optimization. This practice assumes that adversarial robustness is distributed across the parameter space. However, our empirical analysis suggests a different picture. Robust adaptation is highly sparse, with adversarial update signals concentrated in a small subset of parameters. In this section, we present motivating evidence based on AT of a pretrained ViT-B model [9] on CIFAR-10 [24].

3.1 Adversarial Gradients Concentrate on Few Parameters

We analyze how adversarial optimization is distributed across model parameters. During AT, we accumulate the gradient saliency of each parameter under the adversarial loss and normalize the values for comparison. This measures how strongly each parameter participates in adversarial optimization throughout training.

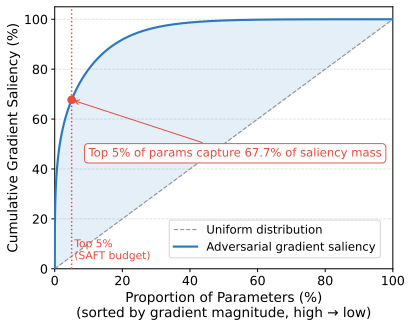

Figure 1: Top 5% of ViT-B parameters capture 67.7% of adversarial gradient.

As shown in Fig. 1, the cumulative distribution is highly skewed. The top 5% of parameters capture 67.7% of the total saliency mass, far above the 5% expected under a uniform distribution. The remaining 95% of parameters together account for only about a third of the cumulative saliency, with the bottom half contributing almost nothing. This finding suggests that adversarial fine-tuning does not update all parameters uniformly. Instead, the adversarial optimization signal is concentrated in a sparse subset of parameters, while the vast majority receive negligible adversarial gradients. From this perspective, full-parameter adversarial fine-tuning is over-parameterized for the purpose of robust adaptation.

3.2 High-Gradient Parameters Are Critical for the Robust Model

To examine whether high-gradient parameters are functionally important for the trained model, we perform a parameter intervention experiment. Starting from an adversarially fine-tuned ViT-B model on CIFAR-10, we identify the parameters with the largest accumulated adversarial gradients and mask them by setting their values to zero. We compare this intervention to masking a random subset, and re-evaluate both clean and robust accuracy without further training.

The results are shown in Table 1. Masking the top 5% high-gradient parameters causes robust accuracy to collapse from 43.60% to 3.70%, a relative loss of more than 90%, while masking a random 5% subset reduces it only to 39.48%. The destructive effect depends on which parameters are removed, not how many. Clean accuracy is also affected by masking high-gradient parameters, indicating that these parameters are critical to the overall function of the network rather than to robustness alone. However, the relative loss in robust accuracy is substantially larger than that in clean accuracy, showing that the robust behavior of the trained model relies on this small subset.

Table 1: Masking RCPs is more destructive than masking random parameters.

Setting	Clean	PGD-20
Baseline	85.04	43.60
5% Random	81.83	39.48
RCPs 5%	**29.82**	**3.70**
Random 10%	66.46	32.18
RCPs 10%	**24.35**	**4.75**

Together with the gradient distribution in Fig. 1, this intervention indicates that high-gradient parameters are both statistically salient and causally critical for the robust behavior of the adversarially trained model. We refer to these parameters as RCPs. The above findings motivate a sparse view of AT. If only a small subset of parameters receives strong adversarial update signals and is critical for robustness, updating the entire model may be unnecessary. Based on these observations, we propose SAFT, which first identifies robustness-critical parameters using adversarial gradient saliency and then updates only these parameters during adversarial fine-tuning. By concentrating adversarial optimization on RCPs while freezing the remaining parameters, SAFT targets the sparse subspace where adversarial optimization lives, and leaves the remaining pretrained weights unchanged.

4 Method

Motivated by the empirical findings in Section 3, we propose SAFT. Instead of updating all parameters during adversarial fine-tuning, SAFT first identifies RCPs using adversarial gradient saliency, and then updates only these parameters while freezing the rest of the model. This design preserves the original model architecture, introduces no adapter modules, and directly confines adversarial adaptation to the parameter subset most relevant to robustness.

4.1 Formulation

Let f_θ denote a pretrained model with parameters θ, and let $\mathcal{D}$ denote the downstream training distribution. Standard AT optimizes

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|_\infty \le \epsilon} \ell\big(f_\theta(x+\delta),\, y\big) \right], \tag{1}$$

where δ is the adversarial perturbation and ϵ is the perturbation budget. Existing AT methods update the full parameter vector θ during the outer minimization. SAFT instead introduces a binary update mask $m \in \{0,1\}^{|\theta|}$, where $m_i = 1$ indicates that θ_i is trainable and $m_i = 0$ indicates that it is frozen. Given an AT loss $\mathcal{L}_{adv}$, the masked update is

$$\theta^{t+1} = \theta^t - \eta \left(m \odot \nabla_\theta \mathcal{L}_{adv}(\theta^t) \right), \tag{2}$$

where η is the learning rate and $\odot$ denotes element-wise multiplication. The central design question is how to construct m so that the selected parameters are those most responsible for robust adaptation. SAFT constructs this mask using adversarial gradient saliency.

4.2 Robustness-Critical Parameter Identification

We identify RCPs through a short calibration phase before sparse fine-tuning. Starting from the pretrained model f_θ, we run a small number of AT steps on a calibration set $\mathcal{D}_{cal}$ and accumulate the adversarial gradient magnitude of each parameter throughout this phase. For parameter θ_i, the accumulated saliency score is

$$s_i = \mathbb{E}_{(x,y)\sim\mathcal{D}_{cal}} \left[\left| \nabla_{\theta_i} \ell\big(f_\theta(x+\delta), y\big) \right| \right], \tag{3}$$

where δ is the adversarial perturbation generated against the current model by the same attack used for AT,

$$\delta = \arg\max_{\|\delta'\|_\infty \le \epsilon} \ell\big(f_\theta(x+\delta'), y\big). \tag{4}$$

The expectation is taken over the calibration data. The saliency score s_i measures how strongly parameter θ_i participates in adversarial optimization across the calibration phase. Parameters with larger scores receive stronger adversarial update signals and are therefore more likely to be critical for robust adaptation. We select the top ρ fraction of parameters according to s_i as RCPs:

$$\mathcal{R} = \mathrm{Top}_\rho\big(\{s_i\}_{i=1}^{|\theta|}\big), \quad m_i = \begin{cases} 1, & \theta_i \in \mathcal{R}, \\ 0, & \theta_i \notin \mathcal{R}. \end{cases} \tag{5}$$

The mask is computed once during the calibration phase and fixed throughout AT. In all experiments, the calibration phase is a single epoch over the training set. When applying SAFT to transformer-based models, we restrict the search space to the main transformation modules, following common practice in parameter-efficient fine-tuning. Specifically, we search for RCPs in the projection matrices of self-attention modules (W_Q, W_K, W_V, W_O) and in the two linear layers of the feed-forward networks (W_{up}, W_{down}). This focuses the update budget on modules that dominate representation transformation.

4.3 Sparse Adversarial Fine-Tuning

After constructing the RCPs mask, SAFT performs AT with masked gradients according to the update rule introduced in Section 4.1. Only RCPs are updated, while all other parameters remain at their pretrained values. SAFT modifies only the update space, not the adversarial objective itself, so it can be combined with different AT losses such as PGD-AT [30], TRADES [46], and MART [38].

SAFT differs from both pruning and LoRA-based PEFT. Unlike pruning, SAFT does not remove parameters or alter the inference architecture. The full pretrained model is retained, and the mask only controls training-time updates. Unlike LoRA-based methods, SAFT does not introduce low-rank adapters or constrain updates to an external low-rank subspace. Instead, it directly updates a sparse subset of the original model parameters selected by adversarial gradient saliency, providing a sparse counterpart to LoRA-based PEFT AT.

This sparse update strategy has two benefits. First, it reduces training memory, since persistent gradients and optimizer states need to be maintained only for the selected ρ fraction of parameters. The backward pass still computes the full gradient in Eq. (2), so SAFT does not reduce the FLOPs of AT but only the selected coordinates keep persistent gradients and optimizer states. Second, by confining adversarial adaptation to RCPs and keeping all other parameters at their pretrained values, SAFT minimizes the unnecessary perturbation that full-parameter AT introduces across the network. The remaining parameters of the model stay close to its pretrained state, which achieves a better trade-off between training cost and robust adaptation.

5 Experimental Results

5.1 Experimental Setup

Datasets and Models. We conduct experiments across datasets. CIFAR-10 and CIFAR-100 [24] each contain 60,000 images of size 32×32, with 50,000 for training and 10,000 for testing, distributed across 10 and 100 classes respectively. TinyImageNet [25] contains 200 classes with 500 training images and 50 for testing per class. We evaluate SAFT on three pretrained backbones, including ViT-B, ViT-S [9], and Swin-B [29], and additionally ResNet-50 [15] in Appendix B. ViT-B and ViT-S both contain 12 Transformer blocks, with hidden widths of 768 and 384, while Swin-B adopts a hierarchical architecture with a shifted-window attention mechanism. All models are pretrained on ImageNet-1K [33]. We keep the default architectures and adjust the image and patch sizes to match the input resolution. For CIFAR-10 and CIFAR-100, the image size is set to 32 and the patch size to 4. For TinyImageNet, the image size is set to 64 and the patch size to 8.

Evaluation Metrics. We report classification accuracy on clean inputs and adversarial examples. Four attacks, PGD-20 [30], CW-20 [3], PGD-100 [30], and AutoAttack [7] are used to generate perturbations, where the number following PGD or CW refers to the number of iterations.

Comparison Methods. We compare SAFT against full-parameter adversarial fine-tuning and two LoRA-based PEFT-AT methods, LoRA [18] and LNLoRA [44]. To verify the effectiveness of adversarial gradient saliency for parameter selection, we compare against random parameter selection, weight-magnitude selection and clean-data gradient saliency under the same parameter budget.

Training Details. We use SGD [32] with momentum 0.9 and weight decay 10^{-4}. Models are trained for 40 epochs with an initial learning rate of 0.04, decayed by a factor of 10 at epochs 35 and 38. For adversarial perturbations, we use the ℓ_∞ norm with $\epsilon = 8/255$ and step size $\alpha = 2/255$. For LoRA, LNLoRA and DoRA, the rank is set to $r = 32$. For SAFT, we select the top 5% of parameters as RCPs. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

5.2 Main Results across Architectures

We compare SAFT against full-parameter adversarial fine-tuning and two LoRA-based PEFT-AT methods on three pretrained backbones. Results on CIFAR-10 are reported in Table 2. SAFT consistently outperforms LoRA-based PEFT-AT across all architectures. On every backbone and under every attack, SAFT achieves higher clean and robust accuracy than both LoRA and LNLoRA, while using fewer trainable parameters. The improvement is substantial. On ViT-B, SAFT improves over LoRA by 3.04% in clean accuracy and 2.83% on PGD-20, while reducing trainable parameters from 4.7M to 3.1M. On ViT-S, the gain is even larger, with SAFT improving robust accuracy by 2.99% on PGD-20 and 3.50% on AutoAttack. On Swin-B, SAFT improves over LoRA by 3.67% on PGD-20 and 3.35% on AutoAttack, with comparable clean accuracy.

SAFT approaches or surpasses full-parameter AT in robust accuracy while updating fewer than 5% of parameters. On ViT-B, SAFT surpasses Full AT under all four attacks, despite training only 3.6% of the parameters. On ViT-S and Swin-B, SAFT approaches Full AT in robustness, with a remaining gap of about 1% across attacks. Clean accuracy of SAFT is lower than Full AT by 2.5%, likely reflecting that the frozen parameters provide less flexibility for fitting clean examples.

Sparse update can outperform dense update in robustness. The observation that SAFT surpasses Full AT on ViT-B is particularly noteworthy. We attribute this to an implicit regularization effect. By restricting adversarial updates to a small sparse subspace, SAFT reduces the effective capacity of the optimizer and mitigates robust overfitting. This finding suggests that full parameter updates are not necessarily the upper bound for adversarial fine-tuning. SAFT also applies to CNNs. On ResNet-50 and CIFAR-10, it reaches 49.32% PGD-20 accuracy compared to the accuracy of 48.95% for Full AT (Appendix B).

5.3 Main Results across Datasets

We further evaluate SAFT on CIFAR-100 and TinyImageNet to test whether the conclusions generalize to more complex datasets. Results are reported in Table 3, with CIFAR-10 in Table 2. SAFT's

Table 2: Comparison across architectures on CIFAR-10. Trainable parameters refer to updated parameters during AT. Below each backbone: its total parameter count, to which LoRA-based methods add adapters.

Model	Method	Trainable Params	Clean	PGD-20	CW-20	PGD-100	AutoAttack
ViT-B (85.2M)	Pretrained	–	96.26	0.00	0.00	0.00	0.00
	Full Parameter	85.2M (100%)	85.04	43.60	44.57	42.96	41.16
	LoRA	4.7M (5.5%)	79.51	41.58	41.27	41.31	38.63
	LNLoRA	4.8M (5.6%)	79.59	41.79	41.50	41.40	39.07
	SAFT	**3.1M** (3.6%)	**82.55**	**44.41**	**44.87**	**44.04**	**41.61**
ViT-S (21.4M)	Pretrained	–	94.49	0.00	0.00	0.00	0.00
	Full Parameter	21.4M (100%)	83.09	42.60	43.96	42.19	40.53
	LoRA	1.1M (5.1%)	75.57	38.50	37.73	38.10	34.97
	LNLoRA	1.1M (5.1%)	75.09	38.47	37.62	38.43	35.36
	SAFT	**0.8M** (3.7%)	**78.13**	**41.49**	**41.43**	**41.11**	**38.47**
Swin-B (85.9M)	Pretrained	–	87.26	0.00	0.00	0.00	0.00
	Full Parameter	85.9M (100%)	72.00	39.03	38.24	38.80	35.92
	LoRA	5.9M (6.9%)	67.29	33.38	32.07	33.15	29.85
	LNLoRA	5.9M (6.9%)	67.30	33.42	32.09	33.16	29.84
	SAFT	**4.1M** (4.8%)	**67.37**	**37.05**	**35.75**	**36.53**	**33.20**

advantage over LoRA-based PEFT-AT grows with dataset complexity. On CIFAR-10, SAFT improves over LoRA by about 3% on PGD-20 and AutoAttack. On CIFAR-100, the improvement is around 3%. On TinyImageNet, the gap becomes dramatic. LoRA and LNLoRA collapse to low robust accuracy, while SAFT maintains 18.01% on PGD-20 and 14.34% on AutoAttack, an improvement of more than 11% and 9% respectively. The collapse of LoRA on TinyImageNet suggests that as the downstream task becomes more distant from the pretraining distribution, the low-rank optimization subspace becomes increasingly inadequate for capturing the directions needed for robustness. Sparse update, by contrast, draws from the full parameter space and remains effective even under significant distribution shift.

Table 3: Comparison across datasets on ViT-B (CIFAR-10 results are in Table 2). SAFT consistently outperforms LoRA-based PEFT-AT in robust accuracy across CIFAR-10, CIFAR-100, and TinyImageNet. The robustness gap of LoRA-based methods grows with dataset complexity, while SAFT remains close to full-parameter AT.

Dataset	Method	Clean	PGD-20	CW-20	PGD-100	AutoAttack
CIFAR-100	Pretrained	86.10	0.00	0.00	0.00	0.00
	Full Parameter	64.14	24.42	25.14	24.09	22.58
	LoRA	60.78	22.25	22.21	21.81	19.76
	LNLoRA	61.05	21.96	22.01	21.52	19.80
	SAFT	**62.60**	**24.84**	**24.80**	**24.45**	**21.80**
TinyImageNet	Pretrained	80.94	0.00	0.00	0.00	0.00
	Full Parameter	58.82	18.64	18.36	18.20	14.92
	LoRA	55.53	6.71	7.27	5.97	4.89
	LNLoRA	56.03	7.92	8.41	7.27	5.73
	SAFT	54.40	**18.01**	**16.86**	**17.79**	**14.34**

SAFT remains close to full-parameter AT across all three datasets. On CIFAR-100, SAFT slightly outperforms Full AT on PGD-20 and PGD-100, with a small gap on AutoAttack. On TinyImageNet, SAFT closely matches Full AT in robust accuracy, with a remaining gap of 0.4–1.5%. Clean accuracy of SAFT is somewhat lower than Full AT and, on TinyImageNet, also lower than LoRA-based methods, reflecting the constraint imposed by freezing over 95% of the parameters under significant

distribution shift. Even so, SAFT preserves most of Full AT's robustness across datasets of varying complexity while training only a small fraction of parameters.

5.4 Robustness Critical Parameter under Different Attacks

A natural concern is whether the parameters identified by SAFT depend on the specific attack used to compute adversarial gradients. We test this by identifying RCPs separately under four different attacks (PGD-20, PGD-100, CW-20, and AutoAttack) on ViT-B / CIFAR-10, and compute the pairwise overlap between the resulting RCPs sets.

Results are reported in Table 4. The overlap is consistently high across all attack pairs, ranging from 82.82% to 92.42%. The two PGD variants (PGD-20 and PGD-100) yield the highest overlap of 92.42%, indicating that the number of attack iterations has only a small effect on which parameters are identified as critical. Across structurally different attacks (e.g., CW-20 vs AutoAttack), the overlap remains above 82%, far higher than the 5% level expected if RCPs were arbitrary 5% subsets.

Hence, adversarial gradient saliency does not overfit to a specific perturbation method, and the identified RCPs reflect a structural property of the model rather than an artifact of the attack used during selection. Section 5.5 tests whether they are also robustness-specific.

Table 4: Overlap ratio (%) of RCPs under different attacks on ViT-B with CIFAR-10.

	PGD-20	PGD-100	CW-20	AA
PGD-20	–	92.42	85.87	83.88
PGD-100	92.42	–	86.90	87.96
CW-20	85.87	86.90	–	82.82
AA	83.88	87.96	82.82	–

Table 5: Training with 5% of parameters with different criteria on ViT-B with CIFAR-10.

Selection criterion	Clean	PGD-20
Random	52.32	11.92
Weight magnitude	38.32	22.18
Clean-data saliency	54.33	25.36
Adv. saliency (SAFT)	82.55	44.41

5.5 Specificity of Robustness Critical Parameters

RCPs might simply be generally important parameters. We test this on ViT-B and CIFAR-10 with three controls. **Set identity.** The top 5% of parameters by pretrained weight magnitude share only 5.27% of their entries with the RCPs, and the top 5% by clean-data gradient saliency share 30.94%. **Training under each criterion.** We train separate models that update only the 5% selected by each criterion, with identical budget, schedule and seed (Table 5). Adversarial saliency beats clean-data saliency by 28.2 points in clean accuracy and 19.1 points on PGD-20 although the two selections overlap by a third, so the gain comes from the adversarial-specific part of the selection. **Zeroing.** Zeroing the RCPs of the trained SAFT model reduces PGD-20 accuracy by 86.0% in relative terms, compared with 57.8% for the weight-magnitude subset and 10.2% for a random subset (Appendix C). Together with the attack invariance in Table 4, these controls indicate that RCPs are specific to adversarial adaptation.

5.6 Generalization Across AT Objectives

To test whether SAFT generalizes beyond the standard PGD-AT objective, we apply it to two additional AT objectives, TRADES [46] and MART [38], both designed to address the clean-robust trade-off through loss formulations. Results on ViT-B and CIFAR-10 are reported in Table 6. SAFT outperforms LoRA-based methods under three AT objectives. The improvement under MART is large, where SAFT improves over LoRA by 4.02% in clean accuracy and 2.81% on PGD-20.

SAFT can surpass full-parameter AT under standard objectives. Under TRADES, SAFT also surpasses full-parameter AT on PGD-20 and PGD-100, and improves clean accuracy from 86.24% to 87.00%. Under MART, full-parameter AT achieves the highest robust accuracy overall, with SAFT trailing by 3.5–4.7% across attacks. We attribute this to MART's strong reliance on per-example reweighting, which benefits from updating the full parameter set. Even so, SAFT remains the strongest PEFT-AT method under MART, improving over LoRA by 2.81% on PGD-20 and 2.50% on AutoAttack.

Table 6: SAFT under different AT objectives on ViT-B and CIFAR-10. SAFT consistently outperforms LoRA-based PEFT-AT under all three AT objectives.

Method	AT Objective	Clean	PGD-20	CW-20	PGD-100	AutoAttack
Full Parameter	TRADES	86.24	42.88	43.61	42.55	41.38
LoRA	TRADES	86.26	40.22	39.96	39.79	38.00
LNLoRA	TRADES	86.04	39.47	39.16	39.02	37.38
SAFT	TRADES	**87.00**	**43.07**	**43.06**	**42.74**	**41.04**
Full Parameter	MART	75.91	48.73	46.12	48.57	44.73
LoRA	MART	66.66	42.31	39.37	42.30	38.00
LNLoRA	MART	66.47	42.17	38.67	42.04	37.31
SAFT	MART	**70.68**	**45.12**	**41.39**	**45.07**	**40.50**

Comparison with LoRA variants. Table 7 further compares SAFT with DoRA [28], a stronger LoRA variant, on ViT-B and CIFAR-10 under PGD-AT. DoRA improves over LoRA by 1.60% on PGD-20 and 1.66% on AutoAttack but still trails SAFT despite a larger trainable budget. Since its directional update remains rank-r constrained, this indicates that the limitation of LoRA-based PEFT-AT lies in the low-rank constraint itself.

Table 7: Comparison with LoRA variants on ViT-B and CIFAR-10.

Method	Trainable	Clean	PGD-20	AA
LoRA	4.7M	79.51	41.58	38.63
LNLoRA	4.8M	79.59	41.79	39.07
DoRA	4.8M	80.16	43.18	40.29
SAFT	**3.1M**	**82.55**	**44.41**	**41.61**

5.7 Effect of Robustness Critical Parameter Percentage

To understand how robustness depends on the size of the selected RCPs subset, we vary the percentage from 1% to 7% on ViT-B / CIFAR-10 and compare with full-parameter AT (100%). Results are shown in Table 8. With only 1% of parameters selected, SAFT achieves 43.71% on PGD-20 and 40.82% on AutoAttack, already close to full-parameter AT. It indicates that the robustness-critical subspace is highly concentrated, and that even an extremely sparse update is sufficient to reach the robustness ceiling of full AT.

Table 8: Effect of the RCPs percentage on ViT-B and CIFAR-10. Even with only 1% of parameters selected as RCPs, SAFT recovers most of the robustness of full-parameter AT.

RCPs %	Clean	PGD-20	CW-20	PGD-100	AA
1%	81.51	43.71	43.58	43.35	40.82
3%	81.56	43.88	44.01	43.60	41.20
5%	82.55	44.41	44.87	44.04	41.61
7%	82.66	**44.71**	**44.87**	**44.29**	**41.68**
100% (Full AT)	**85.04**	43.60	44.57	42.96	41.16

Increasing the budget from 1% to 5% yields modest additional gains (PGD-20 from 43.71% to 44.41%, AutoAttack from 40.82% to 41.61%), and clean accuracy improves by about 1%. Beyond 5%, robustness gains become marginal (PGD-20 changes from 44.41% to 44.71% between 5% and 7%, with CW-20 unchanged at 44.87%). This fast saturation reinforces the sparsity finding in Fig. 1.

5.8 Training Cost and Further Analysis

Training cost. Table 9 reports the training cost on ViT-B and CIFAR-10 under an identical configuration. SAFT lowers peak memory by 0.97 GiB relative to Full AT and by 0.55 GiB relative to LoRA. Its time per step matches that of Full AT, since all gradients are still computed, and is 32% lower than that of LoRA. The 40-epoch times are extrapolated from the time per step. The parameter-side saving, at least $4kN(1-\rho)$ bytes for N parameters and k optimizer states per parameter, grows linearly with

Table 9: Measured training cost on ViT-B and CIFAR-10 on a single RTX 4090.

Method	Peak (GiB)	Step (s)	40 ep. (h)
Full AT	6.551	1.118	4.856
LoRA (r=32)	6.129	1.629	7.078
SAFT	5.582	1.114	4.840

Table 10: Overlap (%) between the RCPs recomputed at a given epoch and the calibration mask (ViT-B and CIFAR-10).

Epoch	1	10	20	30	40
Full AT	95.65	50.97	51.45	50.08	50.03
SAFT	95.65	55.73	56.42	56.75	57.58

model size: under Adam it reaches 2.18 and 4.54 GiB for ViT-L and ViT-H (Appendix A). Activations dominate peak memory in our setting, so the share of the parameter side grows for larger backbones and smaller batches.

Mask stability. Recomputing adversarial saliency during training in Table 10 and Appendix D shows that the RCP set drifts. Its overlap with the calibration mask falls from 95.65% after epoch 1 to 55.73% at epoch 10 and then stays stable, an order of magnitude above the roughly 5% expected for random subsets. The static mask is thus a workable rather than an optimal choice.

Distribution shift. On CIFAR-10-C, CIFAR-100-C, Tiny-ImageNet-C [17], CIFAR-10.1 [31] and CINIC-10 [8], SAFT achieves higher clean and PGD-20 accuracy than LoRA under every shift, attains the highest PGD-20 accuracy among all adapted models, and never suffers a larger relative accuracy drop than LoRA, as shown in Table 11 and Appendix E.

Table 11: No-attack / PGD-20 accuracy (%) on shifted data. Corruptions use severity 5. CINIC-10 IN uses a class-balanced 10,000-image subsample.

Benchmark	Pretrained	Full AT	LoRA	SAFT
CIFAR-10.1	–	72.80 / 25.05	66.50 / 27.10	70.00 / 28.20
CINIC-10 IN	–	60.62 / 22.54	56.77 / 23.32	59.05 / 24.30
CIFAR-10-C	77.01 / 0.00	69.58 / 25.21	65.50 / 26.79	68.29 / 27.86
CIFAR-100-C	55.03 / 0.01	43.81 / 11.37	41.80 / 11.43	43.77 / 12.49
Tiny-ImageNet-C	35.20 / 0.00	19.50 / 2.57	16.99 / 0.88	17.73 / 2.79

6 Conclusion

In this work, we revisit adversarial fine-tuning from a parameter-space perspective. Through gradient analysis, we find that adversarial optimization is concentrated in a sparse subset of parameters, which we call RCPs. A masking analysis further shows that this same subset is causally critical for the robust behavior of the trained model. Building on this finding, we propose SAFT, which identifies RCPs via adversarial gradient saliency and performs adversarial fine-tuning by updating only these parameters. Across three architectures, three datasets, and three AT objectives, SAFT consistently outperforms LoRA-based PEFT-AT and approaches or surpasses full-parameter AT in robustness while updating fewer than 5% of parameters. It reduces optimizer-state, persistent-gradient and peak training memory. These results suggest that the optimization subspace relevant to adversarial robustness is sparse rather than low-rank, offering a parameter-space perspective complementary to existing loss-level approaches to AT.

Limitations. We acknowledge three main limitations. First, despite updating fewer than 5% of parameters, SAFT does not bring wall-clock speedup or FLOP reduction over full-parameter AT. The current implementation uses unstructured sparsity, which reduces optimizer state and persistent gradient memory but does not accelerate forward or backward passes on standard hardware. Structured sparse selection at the head or row level is a promising direction to obtain real training speedup. Second, while SAFT improves the trade-off between trainable parameters and robustness, the clean-robust trade-off itself is not eliminated. Clean accuracy of SAFT remains below full-parameter AT in most settings. Third, the RCP mask is static, although about half of it drifts during training (Section 5.8); we have not evaluated periodic re-selection.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62376202). We declare no competing interests.

References

[1] M. Andriushchenko and N. Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

[2] A. Ansell, E. Ponti, A. Korhonen, and I. Vulić. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, 2022.

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[4] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

[5] Y. Chang, C. Guo, Y. Chang, and Y. Wu. Lora-mgpo: Mitigating double descent in low-rank adaptation via momentum-guided perturbation optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 648–659, 2025.

[6] H. Cheng, M. Zhang, and J. Q. Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024.

[7] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[8] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505*, 2018.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[10] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[11] J. Fu, J. Fang, J. Sun, S. Zhuang, L. Geng, and Y. Liu. Loft: Lora-based efficient and robust fine-tuning framework for adversarial training. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[13] D. Guo, A. M. Rush, and Y. Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pages 4884–4896, 2021.

[14] S. Hao, Y. Zhou, and Y. Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] Z. He. Deep learning in image classification: A survey report. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pages 174–177. IEEE, 2020.

[17] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[19] H. Huang and R. Balestriero. Allora: Adaptive learning rate mitigates lora fatal flaws. *arXiv preprint arXiv:2410.09692*, 2024.

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[21] Y. Ji, Y. Liu, Z. Zhang, Z. Zhang, Y. Zhao, X. Hao, G. Zhou, X. Zhang, and X. Zheng. Enhancing adversarial robustness of vision-language models through low-rank adaptation. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pages 550–559, 2025.

[22] X. Jia, Y. Zhang, X. Wei, B. Wu, K. Ma, J. Wang, and X. Cao. Improving fast adversarial training with prior-guided knowledge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):6367–6383, 2024.

[23] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu. A survey of deep learning-based object detection. *IEEE access*, 7:128837–128868, 2019.

[24] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[25] Y. Le and X. Yang. Tiny ImageNet visual recognition challenge. Stanford CS231N course report, 2015.

[26] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[27] B. Liao, Y. Meng, and C. Monz. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4260, 2023.

[28] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen. DoRA: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[31] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning*, pages 5389–5400, 2019.

[32] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[34] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

[35] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[36] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[37] S. Vadera and S. Ameen. Methods for pruning deep neural networks. *Ieee Access*, 10:63280–63300, 2022.

[38] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

[39] Z. Wang, H. Wang, C. Tian, and Y. Jin. Adversarial training of deep neural networks guided by texture and structural information. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4958–4967, 2023.

[40] Z. Wang, H. Wang, C. Tian, and Y. Jin. Efficient adversarial training with multi-fidelity optimization for robust neural network. *Neurocomputing*, 585:127627, 2024.

[41] Z. Wang, H. Wang, C. Tian, and Y. Jin. Preventing catastrophic overfitting in fast adversarial training: A bi-level optimization perspective. In *European Conference on Computer Vision*, pages 144–160. Springer, 2024.

[42] F. K. Waseda, C.-C. Chang, and I. Echizen. Rethinking invariance regularization in adversarial training to improve robustness-accuracy trade-off. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

[44] Z. Yuan, J. Zhang, S. Shan, and X. Chen. Fulllora: Efficiently boosting the robustness of pretrained vision transformers. *IEEE Transactions on Image Processing*, 2025.

[45] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in neural information processing systems*, 32, 2019.

[46] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[47] J. Zhang, F. Liu, D. Zhou, J. Zhang, and T. Liu. Improving accuracy-robustness trade-off via pixel reweighted adversarial training. In *Forty-first International Conference on Machine Learning*, 2024.

[48] Y. Zhang, G. Zhang, P. Khanduri, M. Hong, S. Chang, and S. Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pages 26693–26712. PMLR, 2022.

[49] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1181–1190, 2020.

[50] K. Zhu, F. Hu, Y. Ding, W. Zhou, and R. Wang. A comprehensive review of network pruning based on pruning granularity and pruning time perspectives. *Neurocomputing*, 626:129382, 2025.

A Memory Scaling of the Parameter Side

For N parameters stored in fp32 and an optimizer with k state tensors per parameter, full-parameter AT holds $4N$ bytes of weights, $4N$ of gradients and $4kN$ of optimizer state. SAFT holds the same weights and at most $4N$ of transient gradients, but optimizer state only for the trainable fraction ρ, i.e., at most $8N + 4k\rho N$ bytes. The saving of at least $4kN(1-\rho)$ depends only on model size and optimizer order. Table 12 extrapolates it to larger backbones for Adam ($k = 2$), using the trainable fraction of ViT-B ($\rho \approx 3.6\%$). For SGD with momentum ($k = 1$), which we use in our experiments, the saving is halved.

The parameter count grows as $d^2 L$, whereas activation memory grows as $B\,T\,d\,L$ for batch size B, sequence length T, width d and depth L. Our measurement in Table 9 (a large batch and only $T = 65$ tokens at 32×32 resolution with patch size 4) is therefore the least favourable case for SAFT. For larger backbones, smaller batches and gradient checkpointing, the parameter side becomes the dominant term.

Table 12: Analytic parameter-side memory (weights, gradients and Adam states, fp32) of Full AT and SAFT, extrapolated with the trainable fraction of ViT-B.

Backbone	Params	Full AT (GiB)	SAFT (GiB)	Saved (GiB)
ViT-B	85.2M	1.27	0.66	0.61
ViT-L	304M	4.53	2.35	2.18
ViT-H	632M	9.42	4.88	4.54
ViT-g/14	1.1B	16.39	8.49	7.90
ViT-G/14	1.8B	26.82	13.90	12.92
SwinV2-G	3.0B	44.70	23.16	21.54
InternViT-6B	5.9B	87.92	45.56	42.36

B Other Datasets and a Convolutional Backbone

We repeat the gradient analysis of Section 3 on further datasets and on ResNet-50 [15]. Table 13 reports the share of the cumulative adversarial saliency mass captured by the top 5% of parameters. The concentration is consistent across datasets for ViT-B and weaker but still far above the uniform level of 5% for the convolutional backbone. On ResNet-50 / CIFAR-10, SAFT reaches 49.32% PGD-20 accuracy, compared with 48.95% for Full AT.

Table 13: Share of cumulative adversarial gradient saliency captured by the top 5% of parameters.

Backbone	Dataset	Top-5% share (%)
ViT-B	CIFAR-10	67.7
ViT-B	CIFAR-100	61.88
ViT-B	TinyImageNet	63.59
ResNet-50	CIFAR-10	43.15

C Zeroing Interventions on the SAFT Model

Table 1 zeroes parameters of the Full AT model. Table 14 extends the intervention to the trained SAFT model on ViT-B and CIFAR-10 with two further criteria, each zeroing 5% of the parameters. Zeroing the RCPs causes by far the largest damage, whereas the weight-magnitude subset causes smaller and differently structured damage, even though large weights are generally important.

D Drift of the Robustness-Critical Parameters during Training

We recompute adversarial gradient saliency at several epochs and compare each resulting top-ρ set with the mask fixed at calibration, for the SAFT run and, as a reference, a Full AT run (Table 10). The set reorganizes once as the model leaves its pretrained state and is stable thereafter. At the end of

Table 14: Zeroing 5% of the parameters of the trained SAFT model. Relative drops are computed with respect to the unmasked SAFT model (82.55% clean, 44.41% PGD-20).

Zeroed subset (5%)	Clean	PGD-20	Rel. drop, clean	Rel. drop, PGD-20
Random	76.41	39.86	7.4%	10.2%
Weight magnitude	62.69	18.75	24.1%	57.8%
RCPs	30.60	6.23	62.9%	86.0%

training it still overlaps the calibration mask by 57.58%, an order of magnitude above the roughly 5% expected for two random subsets of this size. Table 8 further shows that reducing the budget from 5% to 1% costs only 0.70 points on PGD-20 and 0.79 on AutoAttack. This suggests that robustness is carried by a core substantially smaller than the mask, and that a static mask suffices as long as it contains this core. We did not evaluate dynamic re-selection.

E Distribution Shift

Synthetic shifts. We report the mean corruption accuracy (mCA) on CIFAR-10-C, CIFAR-100-C and Tiny-ImageNet-C [17], averaged over the 15 standard corruption types at all 5 severity levels, and the relative degradation $\Delta\% = (\text{Clean} - \text{mCA})/\text{Clean} \times 100$, where Clean is the in-distribution accuracy of each model from Tables 2 and 3 (lower is better). **Natural shifts.** We evaluate on CIFAR-10.1 [31], an independently re-collected CIFAR-10 test set, and on the ImageNet-sourced portion of the CINIC-10 [8] test split. **Robustness under shift.** We additionally evaluate PGD-20 under both shift families. For corruptions we use severity 5 with a stratified sample of 1,000 images per corruption type. For CINIC-10 we use a class-balanced 10,000-image subsample of the same ImageNet-sourced pool, which is why its no-attack accuracies in Table 11 differ slightly from Table 16. SAFT has a smaller relative degradation than LoRA on every benchmark and than Full AT on three of the five, and the highest PGD-20 accuracy under every shift.

Table 15: Mean corruption accuracy (mCA, %) and relative degradation ($\Delta\%$, lower is better) on ViT-B.

	CIFAR-10-C		CIFAR-100-C		Tiny-ImageNet-C	
Model	mCA	$\Delta\%$	mCA	$\Delta\%$	mCA	$\Delta\%$
Pretrained	86.81	9.81	68.51	20.44	51.81	35.99
Full AT	76.16	10.45	52.85	17.61	28.01	52.38
LoRA	70.73	11.04	50.19	17.43	25.12	54.76
SAFT	73.98	10.38	51.99	16.95	25.73	52.70

Table 16: Accuracy (%) and relative degradation ($\Delta\%$) under natural shift on ViT-B. CINIC-10 IN: ImageNet-sourced portion of the CINIC-10 test split.

		CIFAR-10.1		CINIC-10 IN	
Model	Clean	Acc.	$\Delta\%$	Acc.	$\Delta\%$
Full AT	85.04	72.80	14.39	60.11	29.31
LoRA	79.51	66.50	16.36	56.50	28.94
SAFT	82.55	70.00	15.20	58.79	28.78

F Broader Impact

SAFT contributes to making adversarial training more accessible by reducing the optimizer-state and gradient memory cost of robust adaptation on large pretrained models, which lowers the barrier for researchers and practitioners with limited compute resources to deploy robust models in safety-sensitive applications. As a defense method, our work has no foreseeable negative societal impact beyond the general dual-use nature of adversarial robustness research.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state that adversarial optimization is concentrated on a sparse subset of parameters, that SAFT updates only these Robustness-Critical Parameters, and that it improves robustness with fewer than 5% trainable parameters. These claims are supported by the empirical gradient analysis in Section 3, the method in Section 4, and the experiments in Section 5.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6, including that the current implementation uses unstructured sparsity and therefore reduces optimizer-state and gradient memory but does not provide wall-clock speedup on standard hardware. We also note that SAFT improves but does not eliminate the clean-robust accuracy trade-off.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The mathematical formulations in Section 4 define the adversarial fine-tuning objective, the masked update rule, and the RCPs selection criterion.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 specifies the datasets, model architectures, evaluation attacks, baselines, optimizer, training schedule, perturbation budget, attack step size, LoRA rank, SAFT selection ratio and compute device used in the experiments. The method in Section 4 also describes how RCPs are identified and how the sparse update mask is applied during adversarial fine-tuning.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use publicly available datasets and pretrained models. Code and scripts to reproduce the main results are available at `https://github.com/HandingWangXDGroup/SAFT`.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the dataset splits, evaluated architectures, image and patch sizes, evaluation attacks, comparison methods, optimizer, learning-rate schedule, weight decay, perturbation budget, attack step size, LoRA rank, and SAFT parameter budget. These details are sufficient to understand the reported results and compare SAFT with full-parameter AT and LoRA-based PEFT-AT baselines.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The current experiments report results from runs with fixed random seeds and deterministic experimental settings, and the evaluated datasets use fixed standard test sets. We do not report error bars or statistical significance tests because our main comparisons are conducted under the same training and evaluation protocol across methods.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 states that all experiments are conducted on a single NVIDIA RTX 4090 GPU, and Table 9 reports the peak GPU memory, time per step and 40-epoch training time of Full AT, LoRA and SAFT.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research studies adversarial robustness and parameter-efficient adversarial training using standard public image-classification datasets and pretrained models. We do not collect private data, involve human subjects, or release systems intended for harmful use.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Appendix F. The positive impact is that SAFT may make adversarial training more accessible by reducing trainable parameters.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce or release high-risk pretrained models, image generators, scraped datasets, or other assets with substantial misuse potential. The experiments are conducted on standard image-classification datasets and focus on robustness evaluation and adversarial fine-tuning.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original works for the datasets, model architectures, adversarial attacks, and baseline methods used in the experiments.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, pretrained model, or other standalone research asset. The proposed contribution is an adversarial fine-tuning method and empirical analysis based on existing public datasets and models.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, human annotation, user studies, or research with human subjects. Therefore, participant instructions, screenshots, and compensation details are not applicable.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or crowdsourcing experiments. Therefore, IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method, experiments, and analyses do not use LLMs as an important, original, or non-standard component. Any use of general writing or editing tools, if applicable, does not affect the methodology, scientific rigor, or originality of the research.