ATTRIBUTION SENSITIVITY GUIDED PARAMETER ALLOCATION FOR ADVERSARIALLY ROBUST PARAMETER-EFFICIENT FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained foundation models that undergo standard pre-training can be efficiently fine-tuned for downstream tasks using parameter-efficient fine-tuning (PEFT) methods. However, these models remain highly vulnerable to adversarial perturbations. Although existing works provide solutions for adversarially robust PEFT, they often distribute PEFT parameters uniformly across layers, and less focus has been given to the distribution of the limited capacity of PEFT methods across model layers tailored for adversarial robustness. In this work, we introduce a novel method that guides the allocation of LoRA ranks and parameters across transformer blocks, providing better adversarial robustness compared to uniform allocation. Specifically, we re-examine a representation-level diagnostic, *attribution sharing*, for robust PEFT, which reveals mixed alignment between the gradient of the adversarial loss and attribution sharing. We then provide a theoretical analysis of why this mixed alignment can impede adversarial training optimization. This motivates our method, **Attribution Sensitivity**, a one-shot method that assigns more LoRA capacity to blocks where global attribution sharing is less sensitive to relative weight changes. Experiments on CIFAR-10 and ImageNet-R with standard-pretrained ViT-B/16 and DeiT-Tiny backbones across three rank budgets achieve the highest average robust accuracy, supporting attribution sensitivity as a useful signal for robustness-oriented LoRA rank allocation.

1 INTRODUCTION

Parameter-efficient fine-tuning (PEFT) has become a standard method to adapt pre-trained foundation models to downstream tasks with only a small number of trainable parameters (Lester et al., 2021; Jia et al., 2022; Hu et al., 2021; Houlsby et al., 2019; Pfeiffer et al., 2020; 2021; Cai et al., 2020; Zaken et al., 2022; Chen et al., 2021). For instance, Low-rank adaptation (LoRA) introduces trainable low-rank updates while keeping the pretrained backbone frozen and can still achieve strong clean accuracy (Hu et al., 2021). However, standard PEFT-adapted models remain vulnerable to adversarial perturbations, limiting their use in safety-critical settings.

Recent works have explored adversarially robust fine-tuning of pretrained models through modified training objectives and parameter-efficient adaptation mechanisms (Xu et al., 2024; Lv et al., 2026; Yuan et al., 2025). Some approaches select critical layers using feature vulnerability or sharpness to guide adversarial fine-tuning (Gopal et al., 2025b;a). Separately, AutoLoRA (Zhang et al., 2024) and AdaLoRA (Zhang et al., 2023) adapt layerwise ranks to improve downstream task performance, but their allocation strategies were not developed specifically for adversarial robustness. We study a complementary question at the intersection of these directions: *given a fixed budget, can the LoRA rank allocation among transformer layers be tailored for improved adversarial robustness?*

To investigate this question, we compare existing scoring criteria for distributing a fixed LoRA budget across transformer blocks (Lee et al., 2018; Gopal et al., 2025b;a; Paischer et al., 2026; Li et al., 2026). We then examine a representation-level diagnostic, *attribution sharing*, previously associated with adversarial robustness and robust overfitting (Wei et al., 2025). However, we find that its value does not consistently track robustness in our PEFT experiments. Local gradient analysis reveals a mix of helpful and harmful alignment with adversarial loss descent, which can impede adversarial

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

optimization. These observations motivate a different use of this diagnostic: to guide LoRA capacity allocation across transformer blocks that prioritize less sharing-sensitive regions.

Specifically, our local analysis characterizes when opposing components of per-example adversarial loss gradients along the sharing direction lead to loss interference between examples. Separately, we derive a bound on the first-order sharing change induced by a weight update. Under a common bound on update magnitude, lower sharing sensitivity yields a smaller upper bound on this change, regardless of whether an update increases or decreases sharing. This motivates allocating capacity according to sharing sensitivity rather than enforcing a common direction of sharing change across examples. Building on this analysis, we introduce **Attribution Sensitivity**, a one-shot method that scores transformer blocks by the sensitivity of global attribution sharing to relative weight changes and allocates a fixed LoRA rank budget inversely to these scores.

Empirically, we evaluate Attribution Sensitivity on CIFAR-10 and ImageNet-R with standard-pretrained ViT-B/16 and DeiT-Tiny backbones, and three uniform-equivalent rank budgets. Our method achieves the highest average robust accuracy among the compared allocation rules. Additional comparisons across datasets, backbones, and seeds show lower adversarial training and test cross-entropy, together with higher robust test accuracy. Thus, the results support attribution sensitivity as a useful LoRA rank allocation signal.

Our contributions are threefold:

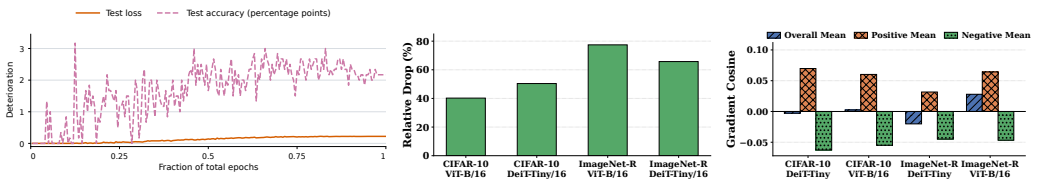
1. We examine attribution sharing under LoRA-based adversarial adaptation, identify heterogeneous alignment between loss-gradient and the global sharing gradient, and characterize when opposing sharing-direction components contribute to local loss interference.
2. We derive a first-order sharing-change bound under bounded relative weight perturbations and use its weight–gradient sensitivity to allocate more LoRA capacity to less-sensitive blocks.
3. Across two datasets, two pretrained backbones, and three rank budgets, our method attains the highest average robust accuracy among compared allocation rules.

2 Related Work

Robustness foundations and PEFT: Robust optimization (Madry et al., 2017) frames training as minimizing worst-case perturbation loss, while Lipschitz-margin training (Tsuzuku et al., 2018) relates score separation and input sensitivity to robustness certificates. Wei et al. (2025) analyze features shared across classes and introduce class-attribution similarity, motivating our sharing measure. For efficient adversarial adaptation, FullLoRA (Yuan et al., 2025) adds normalized low-rank branches throughout a frozen ViT; HyperAT (Lv et al., 2026) generates defense-specific LoRA weights with a hypernetwork and combines the resulting adapters. Our method instead determines where to allocate more LoRA capacity rather than allocating parameters uniformly across layers.

Adversarial layer and parameter selection: CLAT (Gopal et al., 2025b) chooses layers using changes in hidden-feature vulnerability. SAFER (Gopal et al., 2025a) selects layers by sharpness and fine-tunes them with sharpness-aware minimization. CAAT (Li et al., 2026) uses parameter-level criticality to combine masked weight updates with adapters in matrices containing many critical parameters. These methods select adaptation location through feature vulnerability, layer sharpness, or adversarial-loss sensitivity. Our method instead allocates a fixed LoRA budget using the inverse sensitivity of cross-class attribution sharing, favoring layers with lower disruption of shared class evidence.

Non-adversarial scoring and allocation: EVA (Paischer et al., 2026) uses activation singular directions for initialization and explained variance for rank assignment. GoRA (Ye et al., 2026) allocates ranks from weight–gradient scores before tuning and separately constructs a gradient-aware initialization; SNIP (Lee et al., 2018) scores individual connections for pruning. LoRA-GA (Wang et al., 2024) aligns initial low-rank gradients with full-tuning gradients. AdaLoRA (Zhang et al., 2023) revises a global rank budget during tuning by pruning learned singular triplets according to importance. AutoLoRA (Zhang et al., 2024) learns rank-selection variables through meta-learning. These approaches make distinct choices about scoring objectives and capacity allocation and are not optimized for adversarial robustness. However, our focus is one-shot allocation of a fixed LoRA budget



(a) Test Loss and Accuracy Deterioration (b) Attribution Sharing Drop (c) Gradient Cosine

Figure 1: Attribution-sharing diagnostics for robust PEFT with LoRA. (a) Extended adversarial training increases (decreases) test loss (test accuracy) only modestly. (b) Attribution sharing drops substantially with LoRA adaptation. (c) Gradient cosine similarity between adversarial loss gradient and the global attribution sharing gradients.

from sensitivity of cross-class attribution sharing to improve adversarial robustness. Appendix A.7 covers additional robustness and selection literature.

3 Attribution Sensitivity for LoRA Allocation

In this section, we present the empirical and theoretical motivation for our one-shot LoRA allocation method, which assigns more capacity to blocks where attribution sharing is less sensitive to relative weight changes. We first introduce the problem setup and sharing measure, then present empirical observations and local gradient analysis, followed by the block sensitivity score and fixed-budget rank-allocation rule.

Problem setup. We keep the pretrained backbone W_0 frozen and train LoRA parameters ϕ together with a linear classifier (Hu et al., 2021). For K classes, the logits can be written as $z(x) = Ch(x) + b$, where $h(x) \in \mathbb{R}^d$ is the feature vector with dimension d whereas $C \in \mathbb{R}^{K\times d}$ and $b \in \mathbb{R}^K$ are classifier weight and bias parameters respectively. In block ℓ, LoRA modifies each query/value projection $t \in \{Q, V\}$ as:

$$W_{\text{eff},\ell,t} = W_{0,\ell,t} + B_{\ell,t}A_{\ell,t}, \qquad \text{rank}(B_{\ell,t}A_{\ell,t}) \le r_\ell. \tag{1}$$

In our setup, Q and V projections share rank r_ℓ, and we fix the total budget $\sum_\ell r_\ell = R$. We train the LoRA parameters and classifier to minimize cross-entropy loss on adversarial samples. To obtain adversarial samples, PGD approximates the inner maximization over perturbations (Madry et al., 2017) where, given ϵ as the maximum perturbation allowed, the perturbation is then given by: $\|\delta\|_\infty \le \epsilon$ such that $x + \delta$ in the valid pixel domain is the adversarial sample. As will be detailed in the later sections, our goal is to choose the rank for each transformer block before this adversarial training, keeping the total LoRA parameters and total rank budget R fixed.

Global attribution sharing. Following Wei et al. (2025), we represent each class by its mean feature (prototype) on the adversarial training samples, weighted by the classifier parameter row for that class. Concretely, let W denote the backbone weights and $\mu_c(W)$ be the feature prototype for class c. The attribution vector for that class is then given by $a_c(W) = \mu_c(W) \odot C_c$, where $\odot$ is the element-wise product and C_c is the class c row vector of the classifier parameter C.

An attribution vector describes which feature coordinates contribute to the logit of that class. Additionally, the cosine similarity between the attribution vector of classes measures the alignment of their evidence. Consequently, for a fixed collection $\mathcal{P}_0$ of class pairs, we define the global attribution sharing $S(W)$ as:

$$S(W) = \frac{1}{|\mathcal{P}_0|} \sum_{(c,j)\in\mathcal{P}_0} \frac{a_c(W)^\top a_j(W)}{\|a_c(W)\|_2 \|a_j(W)\|_2}. \tag{2}$$

3.1 Attribution sharing for robust PEFT with LoRA

In contrast to the relationship reported by Wei et al. (2025), our experiments suggest that class-attribution sharing behaves differently under LoRA-based robust PEFT. In the extended adversarial-

training experiment (Figure 1 a), robust test accuracy and loss exhibit only modest late-stage deterioration (detailed in Appendix A.11). In a separate experiment (Figure 1 b), increasing adaptation capacity from the zero-rank classifier head-only reference to uniform rank-8 LoRA produces a substantial decrease in attribution sharing across all datasets, backbones, and evaluation settings (detailed in Appendix A.12). Importantly, this decrease in sharing is not accompanied by a corresponding deterioration in adversarial robustness. These observations suggest that, under LoRA-based PEFT, the magnitude of attribution sharing alone is not a reliable proxy for adversarial robustness.

To better understand this behavior, we examine the interaction between the adversarial cross-entropy loss gradient g_i for sample i and the gradient of global attribution sharing $G_S = \nabla_W S$. The relationship is mixed, where $\cos(-g_i, G_S)$ takes both positive and negative values, and the proportions of aligned and conflicting examples vary across settings (Figure 1c and Table 2). For fixed adversarial inputs, a sufficiently small step along G_S may decrease cross-entropy for aligned examples and increase it for conflicting examples to first order. Thus, neither direction along the attribution sharing gradient decreases adversarial loss for all examples to first order. Together with the observed mismatch between attribution sharing and robustness, this cautions against using attribution sharing alone to guide adversarial optimization.

These findings do not contradict the observations of Wei et al. (2025), but caution against directly transferring a relationship observed under full adversarial training to adaptation with a frozen backbone and low-rank updates. Motivated by the observed heterogeneous alignment between adversarial-loss and attribution-sharing gradients, we next investigate whether these local interactions can guide the allocation of LoRA budget across layers to improve adversarial-loss minimization.

3.2 Heterogeneous gradient coupling and adversarial optimization

We examine under what conditions heterogeneous alignment with the attribution sharing gradient leads to interference between examples during adversarial optimization. Let $\mathcal{L}_i(W)$ denote cross-entropy on a fixed adversarial input, and let $F(W) = N^{-1}\sum_{i=1}^{N}\mathcal{L}_i(W)$, with $g_i = \nabla_W \mathcal{L}_i$ and $G_S = \nabla_W S$. We hold adversarial inputs, the classifier, and all non-Q/V parameters fixed, and view W as the vector of Q/V weights. We assume that the losses and S are differentiable with locally Lipschitz gradients and treat sharing as a diagnostic function even though it is not used as training objective.

Interference between examples. Let $\mathcal{T}$ be a fixed linear space of candidate local weight updates, with orthogonal projector P. Define $\widetilde{g}_i = Pg_i$ as the projection of the loss gradient onto the candidate update space $\mathcal{T}$. Likewise, when $PG_S \neq 0$, let $u = PG_S/\|PG_S\|_2$ denote the unit direction of the projected sharing gradient. Then we can decompose the projected gradient as:

$$\widetilde{g}_i = a_i u + b_i, \qquad a_i = \langle \widetilde{g}_i, u\rangle, \qquad \langle b_i, u\rangle = 0. \tag{3}$$

For nonzero projected loss gradients, the sign of $\cos(-\widetilde{g}_i, PG_S)$ is the sign of $-a_i$. Thus, we get mixed alignment when different examples prefer opposite directions of motion along u. For sufficiently small $\eta > 0$, an idealized projected descent step for example i changes the loss of example j as

$$\mathcal{L}_j(W - \eta\widetilde{g}_i) - \mathcal{L}_j(W) = -\eta\,(a_i a_j + \langle b_i, b_j\rangle) + O(\eta^2). \tag{4}$$

When $a_i a_j < 0$, the components of the two gradients along the sharing direction oppose each other, so this component increases $\mathcal{L}_j$ to first order. The total loss of example j increases when $\langle b_i, b_j\rangle < |a_i a_j|$. However, sufficiently strong alignment in directions orthogonal to u can offset this effect, so the complete gradients may not be in conflict. Across examples, such opposition can nevertheless reduce the coherent descent signal along the sharing direction and can impede optimization within the restricted update space.

From directional disagreement to sharing sensitivity. The analysis above shows that examples can prefer opposite directions of sharing change. How strongly their updates change sharing, however, also depends on their sensitivity within the update space. We investigate this as follows:

For the projected descent step $U_i = -\eta\widetilde{g}_i$, Taylor expansion gives: $S(W + U_i) - S(W) = \langle G_S, U_i\rangle + O(\|U_i\|_2^2)$. We denote the linear term by $\Delta S_i^{(1)} := \langle G_S, U_i\rangle$, which is the first-order

approximation to the sharing change. For updates of the same magnitude $\|U_i\|_2 = \eta\|\widetilde{g}_i\|_2 \leq \rho$ budgeted by ρ, a smaller projected sharing-gradient norm $\|PG_S\|_2$ gives a smaller upper bound on the magnitude of the first-order sharing change:

$$\begin{aligned} \Delta S_i^{(1)} &= -\eta\|PG_S\|_2 a_i, \\ |\Delta S_i^{(1)}| &\leq \eta\|PG_S\|_2\|\widetilde{g}_i\|_2 \leq \rho\|PG_S\|_2. \end{aligned} \tag{5}$$

Thus, when comparing updates under the same norm budget ρ, a smaller $\|PG_S\|_2$ gives a smaller upper bound on the first-order change in sharing, regardless of whether an example's update increases or decreases it.

Connection to layer selection. We can use the analysis above to motivate layer selection to allocate PEFT parameters. Layer selection can affect the bound in equation 5 by determining which weight coordinates can change. Specifically, let $\mathcal{B}$ be a selected set of layers and let $P_{\mathcal{B}}$ retain only their Q/V weight coordinates. Writing $G_{S,\ell}$ for the sharing gradient restricted to layer ℓ, we obtain

$$\|P_{\mathcal{B}}G_S\|_2 = \left(\sum_{\ell\in\mathcal{B}} \|G_{S,\ell}\|_2^2\right)^{1/2}. \tag{6}$$

For the same number of equal-sized layers, selecting layers with smaller sharing-gradient norms therefore lowers this bound. This motivates placing more adaptation capacity in less-sensitive layers. Adversarial loss minimization still determines the update directions, but adaptation is preferentially enabled where a given weight change has a smaller potential effect on sharing. The derivations are included in Appendix A.3. We next derive the per-layer score used to guide parameter allocation.

3.3 Local Sensitivity Guides Block Score and Rank Allocation

The above section motivates favoring blocks (layers) in which bounded weight changes have a smaller first-order effect on global sharing. The bound there compares absolute update magnitudes. To account for the scale of the pretrained weights, we now measure sensitivity to changes relative to weight magnitude.

Sensitivity to relative weight changes. Consider a small Q/V weight perturbation U satisfying $|U_k| \leq \alpha|W_k|$ for every weight entry k, where $\alpha > 0$ is a common relative-change limit. The first-order sharing change is $\langle G_S, U\rangle$, and the triangle inequality gives

$$\left|\langle G_S, U\rangle\right| \leq \sum_k |(G_S)_k U_k| \leq \alpha\|W \odot G_S\|_1. \tag{7}$$

Here, $\|\cdot\|_1$ sums the absolute values of all entries. Under the local smoothness assumptions in Appendix A.3, the actual sharing change differs from this first-order term by $O(\|U\|_2^2)$.

Thus, for the same relative-change limit α, a smaller $\|W \odot G_S\|_1$ gives a smaller upper bound on the magnitude of the first-order sharing change, whether sharing increases or decreases. This weight–gradient construction also underlies the connection sensitivity used by SNIP for task-loss-based pruning (Lee et al., 2018) and here, we apply it to global attribution sharing to guide allocation.

Block sensitivity score. Using equation 7 for block ℓ, we average the gradient and weight products within each Q/V matrix and average to obtain the block score κ_ℓ:

$$\kappa_\ell = \sum_{t\in\{Q,V\}} \frac{\|W_{\ell,t} \odot \nabla_{W_{\ell,t}} S\|_1}{|W_{\ell,t}|}. \tag{8}$$

Here, $|W_{\ell,t}|$ denotes the number of entries in the matrix. The score computation is detailed in Appendix A.4. A smaller κ_ℓ consequently identifies a block with a smaller first-order sharing-change bound. This motivates assigning more LoRA capacity to less-sensitive blocks, while leaving subsequent update directions to adversarial cross-entropy minimization. Below we present how these scores determine the LoRA rank allocation.

One-shot Rank Allocation and Training. Given total rank budget R, we assign continuous quotas inversely to block sensitivity and convert them to integer ranks:

$$q_\ell = R \frac{(\kappa_\ell + \tau)^{-1}}{\sum_j (\kappa_j + \tau)^{-1}}, \qquad r_\ell = \lfloor q_\ell \rfloor. \tag{9}$$

The positive numerical floor τ is defined in Equation 15 in the Appendix. Remaining rank units go to the largest fractional remainders, with ties resolved by block index, so $\sum_\ell r_\ell = R$ exactly. Q and V share a rank within each block. The same scores support all budgets, and equal total rank gives equal LoRA parameter counts within each backbone. Algorithm 1 gives the complete procedure; Figure 3 and Appendix A.8 report the resulting allocations.

4 RESULTS

Setup. We evaluate using CIFAR-10 (Krizhevsky et al., 2009) and ImageNet-R (Hendrycks et al., 2021) datasets using 224×224 size inputs. We freeze standard-pretrained ViT-B/16 and DeiT-Tiny backbones and train query/value (Q/V) LoRA adapters (Hu et al., 2021) together with a linear classifier head, including its bias. Q and V projections share the same rank within each of the 12 transformer blocks. The reported ranks $\bar{r} \in \{2, 4, 8\}$ denote uniform-equivalent budgets: $\sum_{\ell=1}^{12} r_\ell = 12\bar{r} \in \{24, 48, 96\}$. Thus, methods share the same total parameter count within each rank budget comparison.

Training and evaluation. Each method uses classifier head warm-up using clean samples for five epochs followed by 20 epochs of adversarial training with cross-entropy loss. We use Adam optimizer with batch size 64, weight decay 0.01, and learning rates 10^{-3} for classifier head warm-up and 3×10^{-4} for adversarial training. All attacks operate in pixel space under L_∞ and $\epsilon = 3/255$. Training and PGD evaluation use random-start PGD-20 (Madry et al., 2017) with step size $2/255$. We additionally evaluate using the Auto-PGD-20 (Croce & Hein, 2020) attack. These settings remain fixed across methods, with only rank allocation varying across them. We evaluate the complete test splits and report mean accuracy (%) across runs. Source code is included *here*.

Allocation rules. Here we describe the baselines used for comparison. Uniform LoRA assigns equal rank $\bar{r}$ to every layer. Static allocation scores are computed once on the full training split using an adversarially adapted *rank-zero reference* where the backbone is frozen and only the classifier head is trained. The non-adversarial scoring methods are adapted accordingly to compute scores using adversarial samples. EVA (Paischer et al., 2026) selects successive activation-covariance eigenvalues from adversarial attention inputs. GoRA/SNIP (Ye et al., 2026; Lee et al., 2018) uses the sum of the mean absolute weight–gradient products for adversarial cross-entropy. CAAT-RPC (Li et al., 2026) accumulates squared mini-batch mean gradients, weighted by each batch's fraction of the training set. CLAT-criticality (Gopal et al., 2025b) uses adjacent-block ratios of feature weakness, measured as the mean clean-to-adversarial CLS-token L_2 deviation divided by feature dimension, where the first block uses its own weakness directly. Gradient norm is equivalent to the gradient-based layer-sharpness score in SAFER (Gopal et al., 2025a). Gradient-SVD allocates rank by successive sums of squared gradient singular values, which is motivated by LoRA-GA (Wang et al., 2024) where we focus on rank allocation and retain ordinary LoRA initialization. AdaLoRA (Zhang et al., 2023) adapts ranks during training by pruning singular triplets according to importance, retaining additional singular-value scalars. AutoLoRA (Zhang et al., 2024) selects ranks through 20 epochs of adversarial meta-learning before final LoRA training. Attribution Sensitivity (Ours) allocates inversely to weight–gradient sensitivity of the mean attribution cosine over a fixed set of positively correlated class pairs from the rank-zero reference.

Adversarial robustness comparison. Table 1 shows that Attribution Sensitivity (Ours) achieves the highest reported mean APGD accuracy and PGD/APGD average among the compared allocation rules in all 12 dataset/backbone/budget settings. Relative to uniform allocation, APGD accuracy improves in all settings, while PGD accuracy improves in 10 of the 12 settings. These improvements are obtained without increasing the LoRA parameter budget. Our method also exceeds the adversarially adapted AdaLoRA and AutoLoRA baselines under APGD in every setting, demonstrating that a fixed, one-shot allocation can outperform the evaluated adaptive-rank approaches.

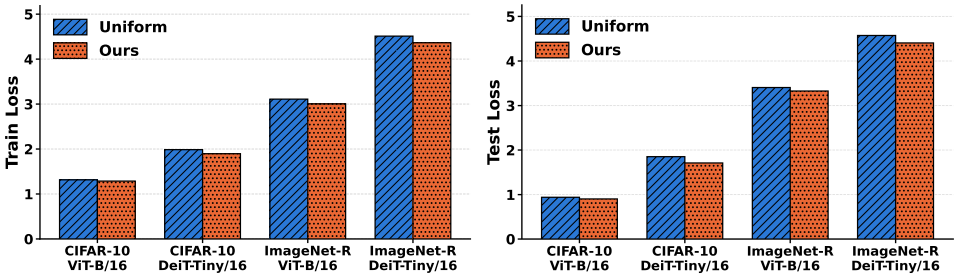
Table 1: Adversarial accuracy (%) of proposed method compared against baselines across different rank budget, datasets and backbones. Avg. is the arithmetic mean of PGD and APGD accuracies. Ranks $\bar{r}$ are uniform-equivalent budgets. Standard deviations are provided in Appendix A.6

	$\bar{r}=2$			$\bar{r}=4$			$\bar{r}=8$		
Method	PGD	APGD	Avg.	PGD	APGD	Avg.	PGD	APGD	Avg.
CIFAR-10 — ViT-B/16									
Uniform LoRA	56.91	33.02	44.97	66.31	38.28	52.30	63.30	63.28	63.29
AdaLoRA	47.50	46.71	47.11	52.39	51.47	51.93	59.02	58.63	58.82
AutoLoRA	48.89	48.84	48.86	56.31	56.27	56.29	72.30	44.09	58.19
EVA	44.66	44.55	44.61	53.99	53.94	53.97	61.46	61.45	61.46
GoRA/SNIP	44.62	44.54	44.58	54.83	54.81	54.82	61.85	61.83	61.84
CAAT-RPC	51.97	51.91	51.94	59.10	59.06	59.08	65.06	65.04	65.05
CLAT-criticality	34.26	33.57	33.92	38.63	37.98	38.31	44.07	43.36	43.72
Gradient norm	51.35	51.30	51.32	58.54	58.52	58.53	65.02	64.99	65.01
Gradient-SVD	50.62	50.60	50.61	57.20	57.18	57.19	62.04	61.99	62.02
Ours	53.00	52.96	**52.98**	59.79	59.78	**59.79**	65.14	65.11	**65.13**
CIFAR-10 — DeiT-Tiny									
Uniform LoRA	28.73	28.70	28.72	28.75	28.71	28.73	31.28	31.21	31.25
AdaLoRA	22.30	22.26	22.28	27.50	27.48	27.49	28.24	28.21	28.23
AutoLoRA	29.00	28.93	28.96	28.48	28.44	28.46	34.45	34.40	34.43
EVA-style	29.67	29.63	29.65	31.67	31.64	31.65	34.03	34.02	34.02
GoRA/SNIP	18.58	18.29	18.44	23.11	22.95	23.03	25.19	25.12	25.15
CAAT-RPC	18.37	18.15	18.26	20.59	20.28	20.44	21.51	21.43	21.47
CLAT-criticality	17.58	16.98	17.28	21.07	20.60	20.84	21.96	21.60	21.78
Gradient norm	19.22	18.80	19.01	20.85	20.73	20.79	26.04	25.95	25.99
Gradient-SVD	26.15	26.02	26.09	28.56	28.51	28.54	23.46	23.32	23.39
Ours	30.53	30.46	**30.49**	32.56	32.55	**32.56**	35.78	35.75	**35.77**
ImageNet-R — ViT-B/16									
Uniform LoRA	25.65	25.61	25.63	28.91	28.53	28.72	32.71	32.33	32.52
AdaLoRA	22.60	21.81	22.21	25.82	25.12	25.47	28.67	28.08	28.37
AutoLoRA	25.47	25.33	25.40	28.71	28.36	28.54	32.12	31.76	31.94
EVA-style	24.94	24.92	24.93	28.24	27.71	27.98	31.29	30.69	30.99
GoRA/SNIP	23.77	23.54	23.66	28.39	28.33	28.36	31.88	31.51	31.69
CAAT-RPC	25.89	25.51	25.70	29.08	28.64	28.86	32.56	32.13	32.35
CLAT-criticality	18.94	18.66	18.80	22.18	21.61	21.89	25.49	25.28	25.38
Gradient norm	26.47	26.20	26.33	29.84	29.42	29.63	33.59	33.14	33.37
Gradient-SVD	26.53	26.14	26.33	29.68	29.32	29.50	32.82	32.31	32.57
Ours	27.02	26.53	**26.78**	30.89	30.39	**30.64**	34.30	33.91	**34.11**
ImageNet-R — DeiT-Tiny									
Uniform LoRA	5.43	5.38	5.41	9.18	9.12	9.15	12.64	12.64	12.64
AdaLoRA	7.99	7.71	7.85	9.44	9.34	9.39	11.32	11.27	11.30
AutoLoRA	2.08	2.06	2.07	8.78	8.74	8.76	11.41	11.39	11.40
EVA-style	7.40	7.34	7.37	9.16	9.09	9.13	10.13	10.09	10.11
GoRA/SNIP	7.36	7.33	7.35	8.53	8.48	8.51	12.14	12.12	12.13
CAAT-RPC	6.03	5.96	5.99	10.52	10.48	10.50	12.96	12.93	12.94
CLAT-criticality	3.87	3.78	3.83	7.41	7.13	7.27	9.53	9.37	9.45
Gradient norm	7.62	7.59	7.60	10.14	10.12	10.13	13.19	13.18	13.18
Gradient-SVD	6.09	6.02	6.05	6.00	5.97	5.99	11.89	11.86	11.87
Ours	11.02	11.00	**11.01**	12.63	12.62	**12.63**	14.85	14.83	**14.84**

The comparisons further show that nonuniform allocation alone does not ensure better robustness. For example, CAAT-RPC and gradient-norm scoring improve APGD accuracy over uniform allocation on CIFAR-10 with ViT-B/16, but underperform uniform allocation with DeiT-Tiny at every

Table 2: Adversarial loss gradient conflict with global attribution sharing. For each sample, $\cos(-g_i, G_S)$ is computed over all Q/V layers. We report the frequency (%) of conflicting samples ($\cos(-g_i, G_S) < 0$) and the overall mean of $\cos(-g_i, G_S)$ ($\times 10^{-2}$). Averaged across seeds.

		Conflict frequency (%)			Mean cosine ($\times 10^{-2}$)		
Dataset	Backbone	Rank-zero	Uniform	Ours	Rank-zero	Uniform	Ours
CIFAR-10	DeiT-Tiny	54.8	37.5	36.2	−0.28	3.05	3.93
CIFAR-10	ViT-B/16	50.0	36.7	34.8	0.25	3.25	3.82
ImageNet-R	DeiT-Tiny	67.9	44.8	44.2	−2.02	2.82	2.78
ImageNet-R	ViT-B/16	33.2	40.2	39.2	2.80	1.12	1.22



(a) Train Loss (b) Test Loss

Figure 2: Comparison of adversarial optimization using train (a) and test (b) loss metrics of final checkpoint between Uniform allocation and our method.

evaluated budget. In contrast, Attribution Sensitivity improves APGD accuracy over uniform allocation across both backbones and datasets. Allocation can also compensate for a smaller adaptation budget: on CIFAR-10 with DeiT-Tiny, our method at $\bar{r} = 4$ achieves 32.56% PGD and 32.55% APGD accuracy, exceeding uniform allocation at $\bar{r} = 8$ (31.28% and 31.21%, respectively) with half as many LoRA parameters. These findings support the motivation in our theoretical analysis as well. The placement of adaptation capacity matters in addition to its total amount, and sharing sensitivity provides a useful criterion for that placement. The following section examines the accompanying optimization and sharing behavior.

4.1 Ablation Studies

Loss–Sharing gradient coupling. Table 2 measures alignment between per-example adversarial loss descent and the global sharing gradient. At the rank-zero reference, 33.2–67.9% of examples have $\cos(-g_i, G_S) < 0$, indicating that their loss-descent directions decrease sharing to first order. The coexistence of aligned and conflicting examples supports the heterogeneous coupling analyzed in Section 3.2 where examples can favor opposite directions of sharing change. Our allocation lowers the conflict frequency relative to uniform allocation in all four settings and increases the mean cosine in three. Here we used the uniform-equivalent rank-8 budget: $\bar{r} = 8$. Appendix A.10 reports computation details the layer-wise results.

Optimization metrics. We compare our attribution sensitivity allocation method with uniform allocation in 12 settings of dataset, backbone and seed combinations. We aim to check whether our allocation method shows evidence of better optimization and sharing values. Figure 2 provides the general evidence using train and test losses between the two methods, whereas Table 3 summarizes all metrics used. Appendix A.9 provides experiment setting details and additional results.

Our allocation method achieves lower adversarial training cross-entropy and lower PGD test cross-entropy in all 12 comparisons settings. Test PGD and APGD accuracies are also higher in all settings. These consistent loss and accuracy outcomes support improved adversarial optimization and performance. Similarly, per-image global attribution sharing gradient-conflict frequency is lower in 9 out of 12 comparisons. In terms of attribution sharing measure (Table 22 in Appendix), the shar-

ing is higher for our method than uniform in 8 out of 12 comparisons under each attack, providing positive evidence for our sharing sensitivity avoidance objective.

Table 3: Optimization and sharing at matched capacity. Counts indicate comparisons favoring Ours over uniform among the 12 dataset/backbone/seed settings.

Measure	Favorable direction	Comparisons
Adversarial training cross-entropy	Lower	12/12
Adversarial test cross-entropy	Lower	12/12
Adversarial test accuracy	Higher	12/12
Per-image gradient-conflict frequency	Lower	9/12
Attribution sharing	Higher	8/12

Rank allocation across blocks. Figure 3 shows how the methods in Table 1 distribute capacity across the 12 transformer blocks. Each block's rank applies to both Q and V projections; the ranks sum to $12\bar{r}$ and remain fixed during adaptation. PGD and APGD evaluate the same allocation. We average the rank allocations over three seeds for visualization. Appendix A.8 lists exact rank values. Our allocation consistently favors earlier blocks, where blocks 1–4 receive 45.83%–68.06% of the budget across the 12 settings, compared with 33.33% under uniform allocation. The methods also differ in concentration: the mean share of the largest block across seeds ranges from 20.83% to 35.42% for our method, while CLAT-criticality reaches 47.92%.

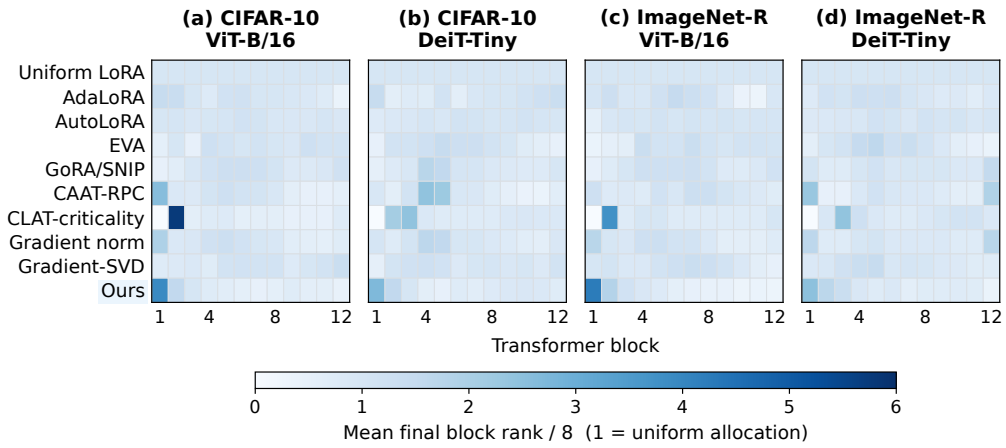


Figure 3: Final rank allocation across datasets and standard-pretrained backbones at uniform-equivalent rank $\bar{r} = 8$ (budget 96). Colors show the mean final block rank across three seeds divided by 8, using the same linear scale in every panel. A value of 1 matches uniform allocation, and larger (darker) values indicate more capacity in that block.

5 Conclusion

We studied how to distribute limited LoRA capacity across transformer blocks for adversarially robust PEFT. We provide theoretical and empirical results that motivate using sharing sensitivity for allocation rather than directly maximizing attribution sharing. We introduce **Attribution Sensitivity**, a one-shot method that assigns more LoRA capacity to blocks where sharing is less sensitive to relative weight changes, while preserving the total rank budget and retaining adversarial cross-entropy training. Across two datasets, two pretrained backbones, and three rank budgets, our method achieves the highest reported average robust accuracy among the compared allocation rules. These findings highlight the importance of robustness-oriented capacity allocation and support attribution sensitivity as an effective signal for improving robustness without increasing the LoRA budget.

AI USE STATEMENT

In this work, we used generative AI tools for the following: aiding or polishing writing, correcting grammar, and fixing errors in Python and LaTeX code. We have not used generative AI tools for generating ideas, methodologies, and datasets. We have reviewed all AI-assisted works. We take responsibility for the final content of this work produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide detailed experiment setup, algorithm, and score calculation in the Appendix. We also provide an anonymous link containing the source code in the main paper and the Appendix.

REFERENCES

Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33:11285–11297, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9620–9629. IEEE, 2021.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Masih Eskandar, Tooba Imtiaz, Zifeng Wang, and Jennifer Dy. Adapt to robustify prompt tuning vision transformers. *arXiv preprint arXiv:2403.13196*, 2024.

Bhavna Gopal, Huanrui Yang, Mark Horton, and Yiran Chen. Safer: Sharpness aware layer-selective finetuning for enhanced robustness in vision transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3999–4008. IEEE, 2025a.

Bhavna Gopal, Huanrui Yang, Jingyang Zhang, Mark Horton, and Yiran Chen. Boosting adversarial robustness with clat: Criticality leveraged adversarial training. In *Forty-second International Conference on Machine Learning*, 2025b.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329. IEEE, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

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

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Wenyun Li, Zheng Zhang, Dongmei Jiang, Yaowei Wang, and Xiangyuan Lan. Efficient adversarial training via criticality-aware fine-tuning. *arXiv preprint arXiv:2604.12780*, 2026.

Kangtao Lv, Wenyan Fan, Huangsen Cao, Kainan Tu, Yihuai Xu, Zhimeng Zhang, Yang Li, Xin Ding, and Yongwei Wang. Hyper adversarial tuning for boosting adversarial robustness of pretrained large vision transformers. *Pattern Recognition*, 171:112158, 2026.

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

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. *Advances in Neural Information Processing Systems*, 38:46215–46267, 2026.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 46–54, 2020.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pp. 487–503, 2021.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024.

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

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024.

Zeming Wei, Yiwen Guo, and Yisen Wang. Identifying and understanding cross-class features in adversarial training. *arXiv preprint arXiv:2506.05032*, 2025.

Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: An automated robust fine-tuning framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Peng Ye, Yuchen Ren, et al. Gora: Gradient-driven adaptive low rank adaptation. *Advances in Neural Information Processing Systems*, 38:114074–114102, 2026.

Zheng Yuan, Jie Zhang, Shiguang Shan, and Xilin Chen. Fulllora: Efficiently boosting the robustness of pretrained vision transformers. *IEEE Transactions on Image Processing*, 2025.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5048–5060, 2024.

Kaijie Zhu, Xixu Hu, Jindong Wang, Xing Xie, and Ge Yang. Improving generalization of adversarial training via robust critical fine-tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4401–4411. IEEE, 2023.

A APPENDIX

A.1 TABLE OF CONTENTS FOR APPENDIX

Section		Page
A.2	Table of Symbols	14
A.3	Theoretical Details and Derivations	14
A.4	Attribution Sensitivity based score computation	17
A.5	Algorithm for attribution-sensitivity allocation	18
A.6	Allocation Results Across Seeds	20
A.7	Additional Related Work	21
A.8	Exact rank allocations	21
A.9	Optimization and sharing evidence	28
A.10	Layer-wise conflict with global attribution sharing	32
A.11	Extended adversarial training: implementation details	33
A.12	Attribution sharing across ranks	33
A.13	Impact Statement and Limitations	34

A.2 TABLE OF SYMBOLS

Symbol	Description
$x,\ \delta$	Input; adversarial perturbation
ϵ	Perturbation bound
$h(x),\ z(x)$	Features; logits
$d,\ K$	Feature dimension; class count
$C,\ C_c,\ b$	Classifier matrix; class row; bias
$c,\ i,\ k$	Class, example, weight-entry indices
$W_0,\ W$	Pretrained weights; backbone weights
$\ell,\ t \in \{Q, V\}$	Block index; query/value projection
$W_{\ell,t},\ W_{\mathrm{eff},\ell,t}$	Projection weights; effective weights
$A_{\ell,t},\ B_{\ell,t},\ \phi$	LoRA factors; all LoRA parameters
$r_\ell,\ \bar{r}$	Block rank; uniform-equivalent rank
R	Total rank budget
$\mu_c(W),\ a_c(W)$	Class mean feature; attribution vector
$\mathcal{P}_0$	Fixed class-pair set
$S(W)$	Global attribution sharing
$G_S,\ G_{S,\ell}$	Sharing gradient; layer restriction
$\mathcal{L}_i(W),\ F(W)$	Adversarial example loss; mean loss
N	Number of examples
$g_i,\ \widetilde{g}_i$	Loss gradient; projected gradient
$\mathcal{T},\ P$	Update space; orthogonal projector
u	Unit projected sharing gradient
$a_i,\ b_i$	Gradient coefficient along u; orthogonal component
$U,\ U_i$	Weight change; example descent step
$\eta,\ \rho$	Step size; update-norm bound
$\Delta S_i^{(1)}$	First-order sharing change
$\mathcal{B},\ P_{\mathcal{B}}$	Selected layers; coordinate projector
α	Relative weight-change limit
κ_ℓ	Block sensitivity score
τ	Positive numerical floor
q_ℓ	Continuous rank quota
$\lvert W_{\ell,t}\rvert$	Number of matrix entries
$\odot$	Element-wise product
$\langle\cdot,\cdot\rangle$	Euclidean inner product
$\lVert\cdot\rVert_1$	Sum of absolute entries
$\lVert\cdot\rVert_2,\ \lVert\cdot\rVert_\infty$	Euclidean norm; largest absolute entry
$\lfloor\cdot\rfloor$	Integer floor

A.3 THEORETICAL DETAILS AND DERIVATIONS

Setup and assumptions. We fix a checkpoint W, the adversarial inputs, the classifier, and all non-Q/V parameters. We regard $W \in \mathbb{R}^D$ as the vector of Q/V weights, where D is their total number of entries. The class-pair set defining S is also held fixed. All gradients below are evaluated at W.

Let $\mathcal{T} \subseteq \mathbb{R}^D$ be a fixed linear space of candidate weight updates, and let P be its orthogonal projector. Thus,

$$P^\top = P, \qquad P^2 = P.$$

For example i, define

$$g_i = \nabla_W \mathcal{L}_i(W), \qquad \widetilde{g}_i = P g_i, \qquad G_S = \nabla_W S(W).$$

We assume that the losses and S are differentiable with locally Lipschitz gradients in a neighborhood of W. All updates are sufficiently small that the corresponding line segments remain in this neighborhood.

When $PG_S \neq 0$, define

$$u = \frac{PG_S}{\|PG_S\|_2}, \qquad a_i = \langle \widetilde{g}_i, u \rangle, \qquad b_i = \widetilde{g}_i - a_i u.$$

By construction,

$$\|u\|_2 = 1, \qquad \langle b_i, u \rangle = 0, \qquad \widetilde{g}_i = a_i u + b_i.$$

Derivation for interference between examples. Consider the projected descent step $W' = W - \eta \widetilde{g}_i$, with $\eta > 0$. A first-order expansion of the loss of example j gives

$$\mathcal{L}_j(W - \eta \widetilde{g}_i) - \mathcal{L}_j(W) = -\eta \langle g_j, \widetilde{g}_i \rangle + R_{ij}(\eta).$$

If $\nabla_W \mathcal{L}_j$ is locally Lipschitz with constant β_j, the remainder satisfies

$$|R_{ij}(\eta)| \leq \frac{\beta_j \eta^2}{2} \|\widetilde{g}_i\|_2^2.$$

Because $\widetilde{g}_i \in \mathcal{T}$, we have $P\widetilde{g}_i = \widetilde{g}_i$. Using $P^\top = P$,

$$\begin{aligned} \langle g_j, \widetilde{g}_i \rangle &= \langle g_j, P\widetilde{g}_i \rangle \\ &= \langle Pg_j, \widetilde{g}_i \rangle \\ &= \langle \widetilde{g}_j, \widetilde{g}_i \rangle. \end{aligned}$$

Substituting the sharing-direction decompositions yields

$$\begin{aligned} \langle \widetilde{g}_i, \widetilde{g}_j \rangle &= \langle a_i u + b_i, a_j u + b_j \rangle \\ &= a_i a_j \langle u, u \rangle + a_i \langle u, b_j \rangle + a_j \langle b_i, u \rangle + \langle b_i, b_j \rangle \\ &= a_i a_j + \langle b_i, b_j \rangle. \end{aligned}$$

The two cross terms vanish because b_i and b_j are orthogonal to u. Therefore,

$$\mathcal{L}_j(W - \eta \widetilde{g}_i) - \mathcal{L}_j(W) = -\eta \big(a_i a_j + \langle b_i, b_j \rangle \big) + O(\eta^2)$$

Here the $O(\eta^2)$ notation holds with the checkpoint, projector, and examples fixed.

If $a_i a_j < 0$, the sharing-direction contribution $-\eta a_i a_j$ is positive. The total first-order loss change is positive precisely when

$$a_i a_j + \langle b_i, b_j \rangle < 0,$$

or, equivalently,

$$\langle b_i, b_j \rangle < |a_i a_j|.$$

Under this strict inequality, the actual loss change is also positive for sufficiently small η. Opposition along u alone is therefore insufficient to establish total loss interference: alignment in the orthogonal components can offset it.

Derivation for first-order sharing sensitivity. For the same update $U_i = -\eta \widetilde{g}_i$, a first-order expansion of sharing gives

$$S(W + U_i) - S(W) = \langle G_S, U_i \rangle + R_S(U_i),$$

where, if $\nabla_W S$ is locally Lipschitz with constant β_S,

$$|R_S(U_i)| \leq \frac{\beta_S}{2} \|U_i\|_2^2.$$

We define the first-order sharing change as the linear term,

$$\Delta S_i^{(1)} := \langle G_S, U_i \rangle.$$

Thus, $\Delta S_i^{(1)}$ approximates the actual sharing change; the difference is the quadratic remainder.

Since $U_i \in \mathcal{T}$, we have $PU_i = U_i$. Consequently,

$$\begin{aligned}\Delta S_i^{(1)} &= \langle G_S, U_i \rangle \\ &= \langle PG_S, U_i \rangle \\ &= -\eta \langle PG_S, \widetilde{g}_i \rangle \\ &= -\eta \|PG_S\|_2 \langle u, \widetilde{g}_i \rangle \\ &= -\eta \|PG_S\|_2 a_i.\end{aligned}$$

The second equality uses $P^\top = P$ and $PU_i = U_i$; the fourth uses $PG_S = \|PG_S\|_2 u$.

By the Cauchy–Schwarz inequality and $\|u\|_2 = 1$,

$$|a_i| = |\langle \widetilde{g}_i, u \rangle| \le \|\widetilde{g}_i\|_2.$$

For updates satisfying the common norm bound $\|U_i\|_2 = \eta \|\widetilde{g}_i\|_2 \le \rho$, we obtain

$$\begin{aligned}|\Delta S_i^{(1)}| &= \eta \|PG_S\|_2 |a_i| \\ &\le \eta \|PG_S\|_2 \|\widetilde{g}_i\|_2 \\ &= \|PG_S\|_2 \|U_i\|_2 \\ &\le \rho \|PG_S\|_2\end{aligned}$$

The bound controls the magnitude of the first-order sharing change, regardless of its sign. The radius ρ is a common bound used to compare weight-update magnitudes; it is not implied by a fixed learning rate or LoRA rank budget. If $PG_S = 0$, the direction u is undefined, but $\langle G_S, U_i \rangle = \langle PG_S, U_i \rangle = 0$ for every $U_i \in \mathcal{T}$, so the bound remains valid.

Derivation for layer-selection identity. Partition the Q/V weight coordinates into L disjoint layerwise blocks. Let $G_{S,\ell} \in \mathbb{R}^{n_\ell}$ contain the derivatives of the same global sharing function S with respect to the n_ℓ Q/V weights in layer ℓ. Then

$$G_S = \begin{bmatrix} G_{S,1} \\ \vdots \\ G_{S,L} \end{bmatrix}, \qquad \sum_{\ell=1}^{L} n_\ell = D.$$

For a selected layer set $\mathcal{B}$, define the coordinate-selection projector $P_{\mathcal{B}}$ by

$$(P_{\mathcal{B}} v)_\ell = \begin{cases} v_\ell, & \ell \in \mathcal{B}, \\ 0, & \ell \notin \mathcal{B}, \end{cases}$$

where v_ℓ denotes the block of v corresponding to layer ℓ. Applying the definition of the Euclidean norm gives

$$\begin{aligned}\|P_{\mathcal{B}} G_S\|_2^2 &= \sum_{\ell=1}^{L} \|(P_{\mathcal{B}} G_S)_\ell\|_2^2 \\ &= \sum_{\ell \in \mathcal{B}} \sum_{k=1}^{n_\ell} (G_{S,\ell})_k^2 \\ &= \sum_{\ell \in \mathcal{B}} \|G_{S,\ell}\|_2^2.\end{aligned}$$

Taking the square root:

$$\|P_{\mathcal{B}} G_S\|_2 = \left(\sum_{\ell \in \mathcal{B}} \|G_{S,\ell}\|_2^2 \right)^{1/2}.$$

This identity follows from the disjoint parameter coordinates of the layerwise blocks. It does not assume independent layer effects or an additive decomposition of S across layers.

Equivalently, when retaining the matrix representation of the Q/V gradients,

$$\|G_{S,\ell}\|_2^2 = \|\nabla_{W_{\ell,Q}} S\|_F^2 + \|\nabla_{W_{\ell,V}} S\|_F^2,$$

because vectorization preserves the Frobenius norm.

Implication and scope. Combining, any update supported on $\mathcal{B}$ with $\|U_i\|_2 \le \rho$ satisfies

$$|\Delta S_i^{(1)}| \le \rho \left(\sum_{\ell \in \mathcal{B}} \|G_{S,\ell}\|_2^2 \right)^{1/2}.$$

At a fixed checkpoint and common update-norm budget, selecting layers with smaller sharing-gradient norms therefore lowers this first-order upper bound. Equal-sized layers allow comparisons with the same number of selected weight coordinates, but are not required for the norm identity itself.

Here $P_{\mathcal{B}}$ retains all Q/V coordinates in the selected layers; it is not necessarily the projector onto their more restricted local LoRA update space. The result motivates layer selection through potential sharing change, without guaranteeing smaller realized sharing drift or reduced pairwise loss interference.

A.4 Attribution Sensitivity based score computation

Attribution Sensitivity based score computation (Ours) in the main text, assigns a fixed LoRA budget using the sensitivity of cross-class attribution sharing. We describe the exact score and integer allocation used in the experiments.

Scoring reference and data. For each dataset, backbone, and seed, we first warm the linear classifier on clean training data for five epochs. A rank-zero branch then trains only this head for 20 adversarial epochs, yielding $\theta^{\mathrm{ref}} = (W_0, \phi = 0, C^{\mathrm{ref}}, b^{\mathrm{ref}})$. The pretrained backbone W_0 remains unchanged. Scoring uses the complete training split $\mathcal{D}_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^N$ in evaluation mode, with the same deterministic training augmentation and batch order in both passes. For the reported experiments, we generate $\widetilde{x}_i$ with untargeted, random-start PGD-20 on cross-entropy at θ^{ref}, using $\epsilon = 3/255$ and step size $2/255$ in pixel space. Attacked inputs are detached before differentiating the sharing score. All prototypes, pair selection, and allocation gradients use training data only.

Pass 1: class attributions and fixed positive pairs. Let $n_c = \#\{i : y_i = c\}$ and let h_W denote the feature extractor without LoRA. With the attacked inputs and classifier held fixed, define

$$\begin{aligned} \mu_c(W) &= \frac{1}{n_c} \sum_{i:y_i=c} h_W(\widetilde{x}_i), \qquad a_c(W) = \mu_c(W) \odot C^{\mathrm{ref}}_{c,:}, \\ \rho_{cj}(W) &= \frac{a_c(W)^\top a_j(W)}{\|a_c(W)\|_2 \|a_j(W)\|_2}. \end{aligned} \tag{10}$$

The first pass accumulates class feature sums and counts over all training examples, including incorrectly classified examples, using their true labels. The classifier bias does not enter a_c. We fix the ordered pair set and its sharing objective as

$$\mathcal{P}_0 = \{(c,j) : c \ne j,\ \rho_{cj}(W_0) > 0\}, \qquad S_{\mathcal{P}_0}(W) = \frac{1}{|\mathcal{P}_0|} \sum_{(c,j)\in\mathcal{P}_0} \rho_{cj}(W). \tag{11}$$

At the reference, this is the positive CAS sum divided by $|\mathcal{P}_0|$. The pair set is held fixed during differentiation; there is no positive-part clipping inside this objective. This is the allocation objective, whereas the reported S_+ in equation 19 selects positive pairs separately for each evaluated model. Scoring requires every class to be represented, nonzero attribution norms, and a nonempty $\mathcal{P}_0$; the implementation rejects an undefined score.

Pass 2: exact gradient of the fixed-input objective. Compute the prototype coefficients $v_c = \partial S_{\mathcal{P}_0} / \partial \mu_c|_{W_0}$ by automatic differentiation, holding C^{ref} fixed. Explicitly, with $a_c^0 = a_c(W_0)$ and $\rho_{cj}^0 = \rho_{cj}(W_0)$,

$$v_c = \frac{2}{|\mathcal{P}_0|} C^{\mathrm{ref}}_{c,:} \odot \sum_{j:(c,j)\in\mathcal{P}_0} \left(\frac{a_j^0}{\|a_c^0\|_2 \|a_j^0\|_2} - \rho_{cj}^0 \frac{a_c^0}{\|a_c^0\|_2^2} \right). \tag{12}$$

The factor two accounts for both orders of each pair. In the second pass, we regenerate the same attacked inputs using the same augmentation, batch order, and attack seeds. For each batch $\mathcal{B}$, use the detached coefficients v_c to form

$$J_{\mathcal{B}}(W) = \sum_{i \in \mathcal{B}} \frac{v_{y_i}^\top h_W(\widetilde{x}_i)}{n_{y_i}}, \qquad G_{\ell,t}^S = \sum_{\mathcal{B}} \nabla_{W_{\ell,t}} J_{\mathcal{B}}(W)\Big|_{W_0}, \quad t \in \{Q, V\}. \tag{13}$$

By the chain rule, $G_{\ell,t}^S = \nabla_{W_{\ell,t}} S_{\mathcal{P}_0}(W)\big|_{W_0}$ for these fixed inputs. The $1/n_{y_i}$ factors recover the full class means, including for imbalanced classes; no additional batch or dataset averaging is applied to $J_{\mathcal{B}}$. This avoids retaining a computation graph for the whole training set. Gradients are taken with respect to the original Q/V weight matrices, with no optimizer update during scoring. Feature sums, prototype coefficients, and accumulated gradients use FP64.

Layer sensitivity and inverse allocation score. For transformer block ℓ, let $m_{\ell,t}$ be the number of entries in its projection matrix $W_{0,\ell,t}$. Its sensitivity and allocation weight are

$$\kappa_\ell = \sum_{t \in \{Q,V\}} \frac{1}{m_{\ell,t}} \sum_{u,v} \left| (W_{0,\ell,t})_{uv} (G_{\ell,t}^S)_{uv} \right|, \qquad u_\ell = \frac{1}{\kappa_\ell + \tau}. \tag{14}$$

The absolute value is applied after accumulating the full-training-set gradient. Thus, κ_ℓ is the sum of the Q and V mean absolute weight–gradient products. These products measure first-order response to relative changes in individual weights. Lower-sensitivity blocks receive larger allocation weights. We use

$$\tau = \max\{10^{-12} s_+, 10^{-30}\}, \qquad s_+ = \mathrm{median}_+(\kappa_1, \ldots, \kappa_L). \tag{15}$$

Here median_+ takes the upper middle element of the sorted strictly positive sensitivities, and returns 1 if none are positive. This specifies the numerical floor even when some layers have zero sensitivity. Absolute sensitivity does not specify the sign of a particular update's sharing change, so inverse allocation is a local preservation heuristic.

Integer ranks and subsequent training. For $L = 12$ blocks and uniform-equivalent rank $\bar{r} \in \{2, 4, 8\}$, let $R = L\bar{r} \in \{24, 48, 96\}$. We form

$$p_\ell = \frac{u_\ell}{\sum_{j=1}^L u_j}, \qquad q_\ell = R p_\ell, \qquad r_\ell = \lfloor q_\ell \rfloor. \tag{16}$$

The remaining $R - \sum_\ell r_\ell$ units go to blocks in descending order of $q_\ell - \lfloor q_\ell \rfloor$, breaking ties by ascending block index. The configured per-block cap is R, so these quotas already obey the cap. The resulting ranks satisfy $0 \le r_\ell \le R$ and $\sum_\ell r_\ell = R$ exactly, with the same rank for Q and V in each block. For the constant-width backbones, this gives $4dR$ LoRA parameters, equal to uniform rank $\bar{r}$; the common classifier adds $K(d+1)$ parameters. Zero-rank blocks have no adapters.

The same sensitivity vector supplies all three budgets. Each allocated model starts from W_0 and the corresponding clean-warmed head, with Kaiming-uniform LoRA A, zero B, and unit scaling. We fix the rank profile and adversarially train LoRA and the classifier for 20 epochs. Sharing determines allocation only; subsequent training minimizes adversarial cross-entropy.

A.5 Algorithm for attribution-sensitivity allocation

Algorithm 1 shares the two scoring passes across all rank budgets. Source code is uploaded at this link:

`https://anonymous.4open.science/r/DF36/README.md`.

Algorithm 1 One-shot LoRA allocation by attribution sensitivity

Input: Training data; frozen backbone W_0; classifier (C, b) after five clean epochs; PGD with $\epsilon = 3/255$ and step size $2/255$.
Output: Ranks r_ℓ and trained parameters (ϕ, C, b) for each budget $R \in \{24, 48, 96\}$.

Adversarially train a copy of (C, b) for 20 epochs without LoRA.
Pass 1: Compute $\mu_c(W_0)$ and $a_c(W_0)$ from PGD attacks on the full training split.
Fix $\mathcal{P}_0$ to ordered, distinct-class pairs with positive attribution cosine; form S by Eq. 2.
Compute and detach the prototype derivatives of S (Eq. 12).
Pass 2: Replay the same attacked batches; accumulate $G_S = \nabla_W S$ at W_0 (Eq. 13).
Compute κ_ℓ at W_0 by Eq. 8 and τ by Eq. 15.
For each $R \in \{24, 48, 96\}$ **do**
    $q_\ell = R\dfrac{(\kappa_\ell + \tau)^{-1}}{\sum_j (\kappa_j + \tau)^{-1}}, \quad r_\ell = \lfloor q_\ell \rfloor$.
    Assign the remaining $R - \sum_\ell r_\ell$ units by descending $q_\ell - \lfloor q_\ell \rfloor$; break ties by ascending block index.
    Restart from W_0 and the input (C, b); initialize Q/V LoRA with shared ranks r_ℓ.
    Fix W_0 and r_ℓ; train only (ϕ, C, b).
End for

Both passes use the scoring classifier in evaluation mode, with identical augmentation and attack seeds. The classifier, $\mathcal{P}_0$, and detached attacked inputs stay fixed during differentiation; only Q/V gradients enter the score. All data used are from the training split.

A.6 Allocation Results Across Seeds

Table 4 gives the same comparisons as Table 1, with sample standard deviation across seeds 1993, 2027, and 3407, using $s^2 = \frac{1}{2}\sum_{i=1}^{3}(a_i - \bar{a})^2$. For Avg., a_i is the arithmetic mean of PGD and APGD accuracy for seed i, so its SD uses the three paired averages.

Table 4: Three-seed mean accuracy (%) above ± sample SD at $\epsilon = 3/255$. PGD, APGD, and Avg. follow Table 1; only the best displayed Avg. means are bold.

	$\bar{r} = 2$			$\bar{r} = 4$			$\bar{r} = 8$		
Method	PGD	APGD	Avg.	PGD	APGD	Avg.	PGD	APGD	Avg.
CIFAR-10 — ViT-B/16									
Uniform LoRA	56.91 ± 15.39	33.02 ± 25.94	44.97 ± 5.32	66.31 ± 17.54	38.28 ± 31.00	52.30 ± 6.73	63.30 ± 0.14	63.28 ± 0.14	63.29 ± 0.14
EVA-style allocation	44.66 ± 1.71	44.55 ± 1.77	44.61 ± 1.74	53.99 ± 0.38	53.94 ± 0.39	53.97 ± 0.38	61.46 ± 0.13	61.45 ± 0.11	61.46 ± 0.12
GoRA/SNIP scoring	44.62 ± 1.95	44.54 ± 1.97	44.58 ± 1.96	54.83 ± 0.75	54.81 ± 0.76	54.82 ± 0.75	61.85 ± 0.48	61.83 ± 0.50	61.84 ± 0.49
CAAT-RPC scoring	51.97 ± 0.58	51.91 ± 0.59	51.94 ± 0.58	59.10 ± 0.88	59.06 ± 0.87	59.08 ± 0.87	65.06 ± 0.22	65.04 ± 0.21	65.05 ± 0.22
CLAT-criticality scoring	34.26 ± 0.22	33.57 ± 0.32	33.92 ± 0.27	38.63 ± 0.29	37.98 ± 0.25	38.31 ± 0.27	44.07 ± 0.32	43.36 ± 0.33	43.72 ± 0.32
Gradient norm	51.35 ± 0.59	51.30 ± 0.63	51.32 ± 0.61	58.54 ± 0.32	58.52 ± 0.33	58.53 ± 0.33	65.02 ± 0.11	64.99 ± 0.10	65.01 ± 0.10
Gradient-SVD	50.62 ± 0.35	50.60 ± 0.37	50.61 ± 0.36	57.20 ± 0.46	57.18 ± 0.48	57.19 ± 0.47	62.04 ± 0.28	61.99 ± 0.29	62.02 ± 0.28
Ours	53.00 ± 0.59	52.96 ± 0.59	**52.98** ± 0.59	59.79 ± 0.19	59.78 ± 0.19	**59.79** ± 0.19	65.14 ± 0.18	65.11 ± 0.17	**65.13** ± 0.17
CIFAR-10 — DeiT-Tiny									
Uniform LoRA	28.73 ± 0.56	28.70 ± 0.57	28.72 ± 0.57	28.75 ± 1.69	28.71 ± 1.70	28.73 ± 1.69	31.28 ± 0.83	31.21 ± 0.86	31.25 ± 0.85
EVA-style allocation	29.67 ± 0.49	29.63 ± 0.47	29.65 ± 0.48	31.67 ± 1.53	31.64 ± 1.51	31.65 ± 1.52	34.03 ± 0.67	34.02 ± 0.66	34.02 ± 0.67
GoRA/SNIP scoring	18.58 ± 0.57	18.29 ± 0.42	18.44 ± 0.49	23.11 ± 4.27	22.95 ± 4.24	23.03 ± 4.25	25.19 ± 5.44	25.12 ± 5.40	25.15 ± 5.42
CAAT-RPC scoring	18.37 ± 0.21	18.15 ± 0.25	18.26 ± 0.23	20.59 ± 1.96	20.28 ± 2.07	20.44 ± 2.01	21.51 ± 0.76	21.43 ± 0.74	21.47 ± 0.75
CLAT-criticality scoring	17.58 ± 0.86	16.98 ± 0.79	17.28 ± 0.82	21.07 ± 1.72	20.60 ± 1.57	20.84 ± 1.64	21.96 ± 2.71	21.60 ± 2.59	21.78 ± 2.65
Gradient norm	19.22 ± 0.57	18.80 ± 0.29	19.01 ± 0.43	20.85 ± 0.51	20.73 ± 0.49	20.79 ± 0.50	26.04 ± 4.56	25.95 ± 4.57	25.99 ± 4.57
Gradient-SVD	26.15 ± 4.60	26.02 ± 4.63	26.09 ± 4.62	28.56 ± 3.05	28.51 ± 3.04	28.54 ± 3.04	23.46 ± 4.95	23.32 ± 5.05	23.39 ± 5.00
Ours	30.53 ± 1.05	30.46 ± 1.14	**30.49** ± 1.10	32.56 ± 2.58	32.55 ± 2.57	**32.56** ± 2.57	35.78 ± 1.02	35.75 ± 1.01	**35.77** ± 1.01
ImageNet-R — ViT-B/16									
Uniform LoRA	25.65 ± 0.24	25.61 ± 0.23	25.63 ± 0.23	28.91 ± 0.28	28.53 ± 0.04	28.72 ± 0.14	32.71 ± 0.24	32.33 ± 0.46	32.52 ± 0.34
EVA-style allocation	24.94 ± 0.25	24.92 ± 0.25	24.93 ± 0.25	28.24 ± 0.24	27.71 ± 0.24	27.98 ± 0.24	31.29 ± 0.16	30.69 ± 0.09	30.99 ± 0.10
GoRA/SNIP scoring	23.77 ± 0.20	23.54 ± 0.46	23.66 ± 0.31	28.39 ± 0.17	28.33 ± 0.16	28.36 ± 0.17	31.88 ± 0.19	31.51 ± 0.09	31.69 ± 0.06
CAAT-RPC scoring	25.89 ± 0.19	25.51 ± 0.49	25.70 ± 0.34	29.08 ± 0.16	28.64 ± 0.39	28.86 ± 0.26	32.56 ± 0.39	32.13 ± 0.44	32.35 ± 0.42
CLAT-criticality scoring	18.94 ± 0.32	18.66 ± 0.24	18.80 ± 0.28	22.18 ± 0.09	21.61 ± 0.14	21.89 ± 0.02	25.49 ± 0.15	25.28 ± 0.18	25.38 ± 0.17
Gradient norm	26.47 ± 0.16	26.20 ± 0.27	26.33 ± 0.08	29.84 ± 0.28	29.42 ± 0.12	29.63 ± 0.10	33.59 ± 0.07	33.14 ± 0.04	33.37 ± 0.05
Gradient-SVD	26.53 ± 0.03	26.14 ± 0.30	26.33 ± 0.15	29.68 ± 0.35	29.32 ± 0.04	29.50 ± 0.19	32.82 ± 0.47	32.31 ± 0.44	32.57 ± 0.46
Ours	27.02 ± 0.40	26.53 ± 0.06	**26.78** ± 0.22	30.89 ± 0.28	30.39 ± 0.19	**30.64** ± 0.23	34.30 ± 0.21	33.91 ± 0.22	**34.11** ± 0.21
ImageNet-R — DeiT-Tiny									
Uniform LoRA	5.43 ± 1.83	5.38 ± 1.83	5.41 ± 1.83	9.18 ± 0.55	9.12 ± 0.55	9.15 ± 0.55	12.64 ± 0.35	12.64 ± 0.37	12.64 ± 0.36
EVA-style allocation	7.40 ± 1.85	7.34 ± 1.83	7.37 ± 1.84	9.16 ± 0.70	9.09 ± 0.69	9.13 ± 0.70	10.13 ± 1.31	10.09 ± 1.29	10.11 ± 1.30
GoRA/SNIP scoring	7.36 ± 1.69	7.33 ± 1.69	7.35 ± 1.69	8.53 ± 3.20	8.48 ± 3.22	8.51 ± 3.21	12.14 ± 0.56	12.12 ± 0.55	12.13 ± 0.55
CAAT-RPC scoring	6.03 ± 3.84	5.96 ± 3.79	5.99 ± 3.81	10.52 ± 0.80	10.48 ± 0.82	10.50 ± 0.81	12.96 ± 0.33	12.93 ± 0.34	12.94 ± 0.33
CLAT-criticality scoring	3.87 ± 4.07	3.78 ± 4.00	3.83 ± 4.04	7.41 ± 0.24	7.13 ± 0.22	7.27 ± 0.23	9.53 ± 0.32	9.37 ± 0.26	9.45 ± 0.29
Gradient norm	7.62 ± 0.80	7.59 ± 0.80	7.60 ± 0.80	10.14 ± 0.95	10.12 ± 0.98	10.13 ± 0.97	13.19 ± 0.15	13.18 ± 0.15	13.18 ± 0.15
Gradient-SVD	6.09 ± 1.04	6.02 ± 1.03	6.05 ± 1.04	6.00 ± 3.44	5.97 ± 3.45	5.99 ± 3.44	11.89 ± 0.57	11.86 ± 0.56	11.87 ± 0.57
Ours	11.02 ± 0.51	11.00 ± 0.50	**11.01** ± 0.51	12.63 ± 0.42	12.62 ± 0.41	**12.63** ± 0.41	14.85 ± 0.31	14.83 ± 0.30	**14.84** ± 0.31

A.7 Additional Related Work

Robustness foundations and PEFT. Hein & Andriushchenko (2017) combine class-score margins with bounds on differences between input gradients to certify a neighborhood in which a prediction cannot change. Their cross-Lipschitz regularizer encourages stability by controlling these gradient differences. Such certificates concern input perturbations, whereas our local sensitivity comparison perturbs backbone parameters at a fixed checkpoint. Tsipras et al. (2018) construct a distribution for which improving adversarial robustness requires sacrificing standard accuracy. This example explains why the two objectives need not agree, without making the trade-off universal across tasks. Ilyas et al. (2019) study predictive signals that support generalization yet remain vulnerable to small input changes. By constructing datasets that isolate such signals, they connect learned representations to adversarial vulnerability and transferability. Their feature-based account motivates examining what information survives adaptation, while leaving open how to distribute a low-rank budget.

ADAPT (Eskandar et al., 2024) studies adversarial training of visual prompts. It generates attacks through the prompted network and trains with a clean term plus an adversarial term; its design also allows patch-embedding and classifier updates. The authors show that attacks which omit the learned prompts can give misleading robustness estimates. This concerns the interaction between the adapted architecture and its evaluation, rather than a rule for allocating LoRA ranks. AutoLoRa (Xu et al., 2024) routes the natural objective through a low-rank branch while optimizing the feature extractor with the adversarial objective, and introduces schedules for loss weights and learning rates. It therefore uses low-rank branches to manage competing training objectives while updating backbone weights. This differs from our frozen-backbone adaptation family, where only Q/V projections and the classifier are trainable.

Adversarial layer and parameter selection. RiFT (Zhu et al., 2023) starts from an adversarially trained model and measures how much worst-case weight perturbations within a module can damage robustness. It chooses a module with low robust criticality, fine-tunes that module on clean data, and interpolates the resulting weights with the original model. The choice of a less critical module serves its goal of recovering generalization while retaining existing robustness. RiFT thus supplies a related preservation perspective, but its initialization, clean fine-tuning objective, and selected unit differ from allocating adapter ranks across a standard-pretrained transformer. In particular, a module's tolerance to weight perturbations and the sensitivity of cross-class attribution sharing are different quantities.

Non-adversarial scoring and allocation. Wanda (Sun et al., 2024) evaluates a weight by multiplying its magnitude by the norm of the corresponding input-activation column. Its selection target is sparsity in existing weights, whereas our budget determines the dimensions of newly trainable low-rank updates.

A.8 Exact rank allocations

The tables below give the integer rank of each transformer block for every method and seed in Figure 3. Block indices run from input to output. Each entry applies to both Q and V projections; zero disables their LoRA updates in that block. Every row sums to the stated budget. All models use standard pretraining and PGD-20 adversarial adaptation at $\epsilon = 3/255$. The same profile is used for both PGD and APGD evaluation.

Table 5: Rank allocations: CIFAR-10, ViT-B/16, $\bar{r} = 2$ (budget 24).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	2	2	2	2	2	2	2	2	2	2	2	2
	2027	2	2	2	2	2	2	2	2	2	2	2	2
	3407	2	2	2	2	2	2	2	2	2	2	2	2
EVA-style allocation	1993	1	2	1	1	2	3	3	3	2	2	2	2
	2027	1	2	1	1	2	3	3	3	2	2	2	2
	3407	1	2	1	1	2	3	3	3	2	2	2	2
GoRA/SNIP scoring	1993	1	1	2	2	3	2	2	2	2	2	2	3
	2027	1	1	2	2	3	2	2	2	2	2	2	3
	3407	1	1	2	2	3	3	2	2	2	2	2	2
CAAT-RPC scoring	1993	5	2	2	2	3	2	2	2	1	1	1	1
	2027	5	2	2	2	3	2	2	2	1	1	1	1
	3407	5	2	2	2	3	2	2	2	1	1	1	1
CLAT-criticality scoring	1993	0	11	1	2	2	1	1	1	1	1	1	2
	2027	0	11	1	2	2	1	1	1	1	1	1	2
	3407	0	11	1	2	2	1	1	1	1	1	1	2
Gradient norm	1993	4	2	2	2	3	2	2	2	2	1	1	1
	2027	4	2	2	2	3	2	2	2	1	1	1	2
	3407	4	2	2	3	3	2	2	2	1	1	1	1
Gradient-SVD	1993	3	3	3	2	2	2	2	2	2	1	1	1
	2027	3	3	3	2	2	2	2	2	1	1	1	2
	3407	3	3	3	2	2	2	2	2	2	1	1	1
Ours	1993	9	3	2	2	1	1	1	1	1	1	1	1
	2027	8	3	2	2	1	1	1	1	2	1	1	1
	3407	7	3	2	2	1	1	1	1	2	2	1	1

Table 6: Rank allocations: CIFAR-10, ViT-B/16, $\bar{r} = 4$ (budget 48).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	4	4	4	4	4	4	4	4	4	4	4	4
	2027	4	4	4	4	4	4	4	4	4	4	4	4
	3407	4	4	4	4	4	4	4	4	4	4	4	4
EVA-style allocation	1993	2	2	2	5	6	5	7	5	4	3	3	4
	2027	2	2	2	4	6	5	7	5	5	3	3	4
	3407	2	2	2	4	6	5	7	5	5	3	3	4
GoRA/SNIP scoring	1993	2	3	4	5	5	5	5	4	3	3	4	5
	2027	2	3	4	5	5	5	5	4	3	3	4	5
	3407	2	3	4	5	5	5	5	4	3	3	4	5
CAAT-RPC scoring	1993	10	4	3	4	5	5	4	4	3	2	2	2
	2027	10	3	3	4	5	5	4	4	3	2	2	3
	3407	11	4	3	4	5	5	4	4	2	2	2	2
CLAT-criticality scoring	1993	0	23	3	3	3	2	2	3	2	2	2	3
	2027	0	23	3	3	3	2	2	3	2	2	2	3
	3407	0	23	3	3	3	2	2	3	2	2	2	3
Gradient norm	1993	8	3	3	4	5	5	4	4	3	3	3	3
	2027	8	3	4	5	5	5	4	4	3	2	2	3
	3407	8	4	4	5	5	5	4	4	3	2	2	2
Gradient-SVD	1993	4	5	4	4	4	4	5	4	2	3	3	6
	2027	4	5	4	4	4	4	4	5	2	3	3	6
	3407	5	5	4	4	4	4	5	4	2	2	3	6
Ours	1993	19	7	4	3	2	2	2	2	2	2	2	1
	2027	15	7	4	3	3	3	2	2	3	2	2	2
	3407	14	6	4	3	3	2	2	3	3	3	3	2

Table 7: Rank allocations: CIFAR-10, ViT-B/16, $\bar{r} = 8$ (budget 96).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	8	8	8	8	8	8	8	8	8	8	8	8
	2027	8	8	8	8	8	8	8	8	8	8	8	8
	3407	8	8	8	8	8	8	8	8	8	8	8	8
EVA-style allocation	1993	5	8	4	10	9	8	9	8	7	10	9	9
	2027	4	8	4	10	9	8	9	8	8	10	9	9
	3407	5	8	4	10	9	8	9	8	7	10	9	9
GoRA/SNIP scoring	1993	4	5	7	9	10	10	10	9	7	7	8	10
	2027	4	5	8	9	10	10	10	9	7	7	7	10
	3407	4	6	8	10	11	11	10	8	6	6	7	9
CAAT-RPC scoring	1993	21	7	6	8	10	9	9	8	5	4	4	5
	2027	19	7	7	9	10	9	9	8	5	4	4	5
	3407	22	7	7	9	10	9	8	7	5	4	4	4
CLAT-criticality scoring	1993	0	46	5	6	7	4	5	5	4	4	4	6
	2027	0	46	5	6	6	5	5	5	4	4	4	6
	3407	0	46	5	6	6	5	5	5	4	4	4	6
Gradient norm	1993	15	7	7	9	10	9	8	8	6	6	5	6
	2027	15	7	7	10	11	9	8	8	5	5	5	6
	3407	17	7	8	10	11	10	8	7	5	4	4	5
Gradient-SVD	1993	6	6	7	6	9	10	9	9	6	8	9	11
	2027	6	7	7	6	9	9	9	9	7	8	8	11
	3407	6	7	7	5	9	9	9	9	6	8	10	11
Ours	1993	37	13	8	6	5	4	4	4	4	4	4	3
	2027	30	13	8	6	6	5	4	5	6	5	5	3
	3407	27	12	8	6	5	5	4	5	7	6	6	5

Table 8: Rank allocations: CIFAR-10, DeiT-Tiny/16, $\bar{r} = 2$ (budget 24).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	2	2	2	2	2	2	2	2	2	2	2	2
	2027	2	2	2	2	2	2	2	2	2	2	2	2
	3407	2	2	2	2	2	2	2	2	2	2	2	2
EVA-style allocation	1993	2	1	4	1	2	2	2	2	3	3	1	1
	2027	2	2	4	1	2	2	3	2	2	2	1	1
	3407	2	2	4	2	2	2	2	2	2	2	1	1
GoRA/SNIP scoring	1993	1	2	2	4	3	2	2	2	2	1	1	2
	2027	1	1	2	3	3	2	2	2	2	2	2	2
	3407	1	2	2	4	3	2	2	2	2	1	1	2
CAAT-RPC scoring	1993	2	1	2	5	5	2	2	1	1	1	1	1
	2027	2	1	2	4	4	2	2	2	1	1	1	2
	3407	2	1	2	5	4	2	2	2	1	1	1	1
CLAT-criticality scoring	1993	0	4	5	2	2	1	1	1	2	2	2	2
	2027	0	4	5	2	2	1	1	1	2	2	2	2
	3407	0	4	5	2	2	1	1	1	2	2	2	2
Gradient norm	1993	2	2	2	3	3	2	2	2	2	1	1	2
	2027	2	2	2	3	3	2	2	2	2	1	1	2
	3407	2	2	2	3	3	2	2	2	2	1	1	2
Gradient-SVD	1993	1	2	2	3	3	2	3	2	2	2	1	1
	2027	1	2	2	2	3	2	2	2	2	2	2	2
	3407	1	2	2	3	3	2	3	2	2	1	1	2
Ours	1993	5	3	2	1	1	2	1	2	2	2	2	1
	2027	5	3	2	1	1	2	1	2	2	2	2	1
	3407	5	3	2	1	1	2	1	2	2	2	2	1

Table 9: Rank allocations: CIFAR-10, DeiT-Tiny/16, $\bar{r} = 4$ (budget 48).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	4	4	4	4	4	4	4	4	4	4	4	4
	2027	4	4	4	4	4	4	4	4	4	4	4	4
	3407	4	4	4	4	4	4	4	4	4	4	4	4
EVA-style allocation	1993	4	5	5	4	5	5	5	3	3	3	3	3
	2027	4	5	5	4	5	5	5	3	4	3	3	2
	3407	4	5	5	4	6	4	4	3	4	3	3	3
GoRA/SNIP scoring	1993	2	3	4	7	7	4	4	4	3	3	3	4
	2027	2	3	4	7	6	4	4	4	4	3	3	4
	3407	2	3	4	7	6	4	4	4	4	3	3	4
CAAT-RPC scoring	1993	4	3	5	10	9	3	3	3	2	2	1	3
	2027	4	2	4	9	9	4	4	3	2	2	2	3
	3407	3	3	5	10	9	4	3	3	2	2	1	3
CLAT-criticality scoring	1993	0	8	10	3	3	3	3	3	4	4	4	3
	2027	0	8	10	3	3	3	3	3	4	4	4	3
	3407	0	8	10	3	3	3	3	3	4	4	4	3
Gradient norm	1993	4	3	5	7	7	4	4	3	3	3	2	3
	2027	4	3	4	6	6	4	4	4	3	3	3	4
	3407	4	3	4	7	6	4	4	4	3	3	3	3
Gradient-SVD	1993	3	4	4	4	4	3	4	5	5	4	4	4
	2027	2	4	4	5	4	3	4	6	6	4	3	3
	3407	2	5	4	4	4	3	4	5	5	4	4	4
Ours	1993	11	6	4	2	2	3	3	3	3	4	4	3
	2027	11	6	4	2	2	3	3	3	3	4	4	3
	3407	11	6	4	2	2	3	3	3	3	4	4	3

Table 10: Rank allocations: CIFAR-10, DeiT-Tiny/16, $\bar{r} = 8$ (budget 96).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	8	8	8	8	8	8	8	8	8	8	8	8
	2027	8	8	8	8	8	8	8	8	8	8	8	8
	3407	8	8	8	8	8	8	8	8	8	8	8	8
EVA-style allocation	1993	4	9	9	10	12	10	10	9	8	6	5	4
	2027	4	9	9	10	11	10	11	9	8	6	5	4
	3407	4	9	9	10	12	11	10	9	8	5	5	4
GoRA/SNIP scoring	1993	5	7	8	14	13	8	8	7	6	6	6	8
	2027	4	6	7	13	12	9	9	7	7	7	7	8
	3407	4	6	8	15	13	8	8	7	7	6	6	8
CAAT-RPC scoring	1993	8	5	10	20	19	7	7	5	4	3	3	5
	2027	8	5	8	18	18	8	8	6	4	4	3	6
	3407	7	5	9	21	17	8	7	6	5	3	3	5
CLAT-criticality scoring	1993	0	17	20	7	6	6	6	6	7	7	7	7
	2027	0	17	20	7	6	6	6	6	7	7	7	7
	3407	0	17	19	7	6	6	6	6	7	8	7	7
Gradient norm	1993	9	6	9	13	13	8	8	7	6	5	5	7
	2027	8	6	8	13	13	8	8	7	6	6	5	8
	3407	8	6	9	14	12	8	8	7	7	5	5	7
Gradient-SVD	1993	4	8	11	10	9	7	7	8	8	8	7	9
	2027	5	9	9	9	10	7	9	9	7	8	6	8
	3407	4	9	9	11	9	6	8	8	7	8	7	10
Ours	1993	22	12	8	5	4	6	5	7	7	7	7	6
	2027	22	13	8	5	4	6	6	6	7	7	7	5
	3407	21	13	8	5	4	6	5	6	7	8	7	6

Table 11: Rank allocations: ImageNet-R, ViT-B/16, $\bar{r} = 2$ (budget 24).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	2	2	2	2	2	2	2	2	2	2	2	2
	2027	2	2	2	2	2	2	2	2	2	2	2	2
	3407	2	2	2	2	2	2	2	2	2	2	2	2
EVA-style allocation	1993	2	2	1	1	1	3	3	2	2	2	2	3
	2027	2	2	1	1	1	3	3	2	2	2	2	3
	3407	2	2	1	1	1	3	3	2	2	2	2	3
GoRA/SNIP scoring	1993	1	1	2	2	3	2	3	2	2	2	2	2
	2027	1	1	2	2	3	2	3	2	2	2	2	2
	3407	1	1	2	2	3	2	3	2	2	2	2	2
CAAT-RPC scoring	1993	2	1	2	2	2	2	3	3	2	2	2	1
	2027	3	1	1	2	2	2	3	3	2	2	2	1
	3407	3	2	2	2	2	2	2	3	2	2	1	1
CLAT-criticality scoring	1993	0	8	2	2	2	1	1	1	2	1	2	2
	2027	0	8	2	2	2	1	1	1	2	1	2	2
	3407	0	8	2	2	2	1	1	1	2	1	2	2
Gradient norm	1993	4	2	2	2	3	2	2	2	1	1	1	2
	2027	3	2	2	3	3	2	2	2	1	1	1	2
	3407	4	2	2	2	3	2	2	2	1	1	1	2
Gradient-SVD	1993	3	3	2	2	2	2	2	2	2	2	1	1
	2027	3	3	2	2	2	2	2	2	2	2	1	1
	3407	3	3	2	2	2	2	2	2	2	2	1	1
Ours	1993	8	3	2	2	2	1	1	1	1	1	1	1
	2027	8	4	2	2	1	1	1	1	1	1	1	1
	3407	9	4	2	2	1	1	1	1	1	1	1	0

Table 12: Rank allocations: ImageNet-R, ViT-B/16, $\bar{r} = 4$ (budget 48).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	4	4	4	4	4	4	4	4	4	4	4	4
	2027	4	4	4	4	4	4	4	4	4	4	4	4
	3407	4	4	4	4	4	4	4	4	4	4	4	4
EVA-style allocation	1993	2	3	1	6	6	6	6	5	4	2	3	4
	2027	2	3	1	6	6	6	6	5	4	2	3	4
	3407	2	3	1	6	6	6	6	5	4	2	3	4
GoRA/SNIP scoring	1993	2	3	4	5	5	5	5	4	3	3	4	5
	2027	2	3	4	5	5	5	5	4	3	3	4	5
	3407	2	3	4	5	5	5	5	4	3	3	4	5
CAAT-RPC scoring	1993	5	3	3	3	5	4	5	6	4	4	3	3
	2027	5	3	3	3	5	4	5	6	4	4	3	3
	3407	5	5	4	3	4	4	5	5	4	4	3	2
CLAT-criticality scoring	1993	0	15	3	4	4	3	3	3	3	3	3	4
	2027	0	15	3	4	4	3	3	3	3	3	3	4
	3407	0	15	3	4	4	3	3	3	3	3	3	4
Gradient norm	1993	7	4	4	5	5	5	4	4	2	2	3	3
	2027	7	4	4	5	5	5	4	4	2	2	3	3
	3407	7	4	4	5	5	5	4	4	2	2	3	3
Gradient-SVD	1993	5	5	5	4	4	5	5	5	3	2	2	3
	2027	5	5	5	4	4	5	5	5	3	2	2	3
	3407	5	6	5	4	4	5	5	5	3	2	2	2
Ours	1993	15	7	5	4	3	3	2	2	2	2	2	1
	2027	16	7	5	4	3	2	2	2	2	2	2	1
	3407	19	7	5	4	3	2	2	1	2	1	1	1

Table 13: Rank allocations: ImageNet-R, ViT-B/16, $\bar{r} = 8$ (budget 96).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	8	8	8	8	8	8	8	8	8	8	8	8
	2027	8	8	8	8	8	8	8	8	8	8	8	8
	3407	8	8	8	8	8	8	8	8	8	8	8	8
EVA-style allocation	1993	3	5	5	11	9	9	11	9	8	9	9	8
	2027	3	4	5	11	9	9	11	9	8	9	9	9
	3407	3	4	5	11	9	9	11	9	8	9	9	9
GoRA/SNIP scoring	1993	4	6	8	10	10	10	11	9	6	6	7	9
	2027	4	6	8	10	10	10	10	9	7	6	7	9
	3407	4	6	8	10	10	10	10	9	6	6	7	10
CAAT-RPC scoring	1993	9	5	6	6	10	8	11	11	9	8	7	6
	2027	10	5	6	6	9	8	11	11	9	8	7	6
	3407	11	9	8	6	9	7	10	10	8	7	6	5
CLAT-criticality scoring	1993	0	30	6	8	8	6	6	6	6	6	6	8
	2027	0	30	6	8	8	6	6	6	6	6	6	8
	3407	0	30	6	8	8	6	6	6	6	6	6	8
Gradient norm	1993	14	8	7	10	11	9	9	7	5	5	5	6
	2027	13	8	8	10	11	9	9	7	5	5	5	6
	3407	15	8	7	10	10	9	9	7	5	5	5	6
Gradient-SVD	1993	8	9	7	7	9	10	11	10	9	7	5	4
	2027	8	9	7	7	9	10	11	10	9	7	5	4
	3407	8	8	7	7	9	10	11	11	9	7	5	4
Ours	1993	31	14	10	8	7	5	4	4	4	3	3	3
	2027	33	15	9	7	6	5	4	4	4	3	3	3
	3407	38	15	9	7	6	4	4	3	3	2	3	2

Table 14: Rank allocations: ImageNet-R, DeiT-Tiny/16, $\bar{r} = 2$ (budget 24).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	2	2	2	2	2	2	2	2	2	2	2	2
	2027	2	2	2	2	2	2	2	2	2	2	2	2
	3407	2	2	2	2	2	2	2	2	2	2	2	2
EVA-style allocation	1993	1	2	5	3	1	1	2	2	2	2	2	1
	2027	1	2	5	3	1	1	2	2	2	2	2	1
	3407	1	2	5	2	1	1	2	2	2	2	2	2
GoRA/SNIP scoring	1993	2	2	1	2	2	2	2	2	2	2	2	3
	2027	2	2	1	2	2	2	2	2	2	2	2	3
	3407	2	2	1	2	2	2	2	2	2	2	2	3
CAAT-RPC scoring	1993	5	1	1	2	2	2	2	2	1	1	1	4
	2027	4	1	1	2	2	2	2	2	2	1	1	4
	3407	5	1	1	2	2	2	2	2	1	1	1	4
CLAT-criticality scoring	1993	0	2	5	2	2	1	2	2	2	2	2	2
	2027	0	2	5	2	2	1	2	2	2	2	2	2
	3407	0	2	5	2	2	1	2	2	2	2	2	2
Gradient norm	1993	3	2	1	2	2	2	2	2	2	1	1	4
	2027	3	2	1	2	2	2	2	2	2	1	1	4
	3407	3	2	1	2	2	2	2	2	2	1	1	4
Gradient-SVD	1993	2	2	2	3	3	2	2	2	1	1	1	3
	2027	2	2	2	3	3	2	2	2	1	1	1	3
	3407	2	2	2	3	3	2	2	2	1	1	1	3
Ours	1993	5	3	3	2	2	2	1	2	1	1	1	1
	2027	5	3	3	2	1	2	1	1	2	1	2	1
	3407	6	3	3	2	2	2	1	1	1	1	1	1

Table 15: Rank allocations: ImageNet-R, DeiT-Tiny/16, $\bar{r} = 4$ (budget 48).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	4	4	4	4	4	4	4	4	4	4	4	4
	2027	4	4	4	4	4	4	4	4	4	4	4	4
	3407	4	4	4	4	4	4	4	4	4	4	4	4
EVA-style allocation	1993	2	5	7	6	5	5	4	3	4	3	2	2
	2027	2	5	7	6	5	5	4	3	4	3	2	2
	3407	2	5	7	6	5	5	4	3	4	3	2	2
GoRA/SNIP scoring	1993	4	3	3	4	5	4	4	4	4	4	3	6
	2027	4	3	3	4	5	4	4	4	4	3	4	6
	3407	4	3	3	4	5	4	4	4	4	3	4	6
CAAT-RPC scoring	1993	9	2	2	3	4	4	3	4	3	3	3	8
	2027	9	2	2	3	5	4	3	4	3	2	3	8
	3407	10	2	2	4	5	4	3	3	3	3	2	7
CLAT-criticality scoring	1993	0	4	10	5	4	3	3	3	4	4	4	4
	2027	0	4	10	5	4	3	3	3	4	4	4	4
	3407	0	4	10	5	4	3	3	3	4	4	4	4
Gradient norm	1993	7	3	3	4	4	4	3	3	4	3	3	7
	2027	7	3	3	4	4	3	4	3	4	3	3	7
	3407	7	3	3	4	4	4	4	3	3	3	3	7
Gradient-SVD	1993	3	3	4	5	5	4	5	5	5	3	3	3
	2027	3	3	4	5	5	4	4	5	5	3	3	4
	3407	3	3	5	5	5	4	4	5	4	3	3	4
Ours	1993	10	7	6	3	3	3	3	3	2	3	3	2
	2027	9	7	6	4	3	3	2	3	3	3	3	2
	3407	11	7	5	4	3	3	2	3	3	3	2	2

Table 16: Rank allocations: ImageNet-R, DeiT-Tiny/16, $\bar{r} = 8$ (budget 96).

Method	Seed	Transformer block											
		1	2	3	4	5	6	7	8	9	10	11	12
Uniform LoRA	1993	8	8	8	8	8	8	8	8	8	8	8	8
	2027	8	8	8	8	8	8	8	8	8	8	8	8
	3407	8	8	8	8	8	8	8	8	8	8	8	8
EVA-style allocation	1993	5	8	9	12	13	10	11	9	7	5	4	3
	2027	5	8	9	12	13	10	11	9	7	5	4	3
	3407	5	8	9	12	13	10	11	9	7	5	4	3
GoRA/SNIP scoring	1993	8	6	6	8	9	8	8	8	8	7	7	13
	2027	9	6	6	8	9	8	8	8	8	7	7	12
	3407	9	6	6	8	10	7	8	8	8	7	7	12
CAAT-RPC scoring	1993	19	4	4	6	9	8	7	7	6	5	5	16
	2027	18	4	4	7	9	8	7	7	6	5	5	16
	3407	19	4	4	8	10	7	7	7	6	5	5	14
CLAT-criticality scoring	1993	0	8	19	10	7	6	6	7	8	9	9	7
	2027	0	7	20	10	7	6	6	7	8	9	9	7
	3407	0	8	19	10	7	6	6	7	8	9	9	7
Gradient norm	1993	13	6	6	8	9	7	7	7	7	6	6	14
	2027	14	6	6	8	8	7	7	7	7	6	6	14
	3407	13	6	6	8	9	7	7	7	7	6	6	14
Gradient-SVD	1993	7	7	9	11	12	8	8	8	7	7	6	6
	2027	7	7	10	11	11	8	9	8	7	7	6	5
	3407	7	9	10	11	12	7	8	8	7	6	5	6
Ours	1993	20	14	11	7	6	7	5	6	5	6	6	3
	2027	18	14	12	8	6	7	5	5	6	6	6	3
	3407	22	13	10	7	6	7	5	6	6	6	5	3

A.9 Optimization and sharing evidence

Tables 17–22 report all 12 matched comparisons between uniform and Ours: CIFAR-10 and ImageNet-R, standard-pretrained ViT-B/16 and DeiT-Tiny/16, and seeds 1993, 2027, and 3407, at $\epsilon = 3/255$. Both methods use 20 epochs of adversarial training and a total tied-Q/V rank budget $\sum_{\ell=1}^{12} r_\ell = 96$, equivalent to uniform rank 8; each block's rank applies to both Q and V. Every difference is $\Delta = \text{Ours} - \text{uniform}$, calculated before rounding, as are the direction counts. Cross-entropy and sharing use six decimals, accuracy uses two decimal percentage points, and conflict percentages use one decimal.

Loss and accuracy. Training cross-entropy is the online epoch-20 statistic accumulated over adversarial minibatches while parameters are being updated. Final-checkpoint PGD cross-entropy instead evaluates the saved final model on 200 CIFAR-10 or 1,000 ImageNet-R test images. Full-test accuracy and sharing use 10,000 or 6,000 images, respectively. All PGD attacks use 20 steps; APGD uses 20 steps and one restart. Each model is evaluated on its own saved attacks.

Conflict and original-pair sharing. Let P_0 be the original ordered positive class-pair set fixed at the rank-zero adversarial training reference used for allocation. Using the class-attribution cosines ρ_{cd} in Eq. 19, we report the signed mean

$$S_{P_0} = \frac{1}{|P_0|} \sum_{(c,d)\in P_0} \rho_{cd}. \tag{17}$$

The evaluated cosines retain their signs and the pairs remain fixed; this differs from S_+, whose positive pairs are refreshed for each model and condition. Class means use true labels and each model's own learned classifier, with bias excluded. For conflict, g_i is the adversarial cross-entropy gradient of image i and G_S is the gradient of S_{P_0} on the complete 200/1,000-image subset. We report the fraction with $\cos(-g_i, G_S) < 0$, using concatenated full Q/V weight gradients across all 12 blocks. These diagnostics use each model's own saved PGD attacks and its own sharing gradient, with attacked pixels held fixed during differentiation. The gradients are with respect to full Q/V matrices, not LoRA factors or optimizer updates. All reported global cosines are defined.

Table 17: Online epoch-20 adversarial-minibatch cross-entropy. Lower is favorable; Ours is lower in 12/12 comparisons.

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	1.312974	1.290207	−0.022766
CIFAR-10	ViT-B/16	2027	1.318241	1.285709	−0.032532
CIFAR-10	ViT-B/16	3407	1.314185	1.286289	−0.027896
CIFAR-10	DeiT-Tiny/16	1993	1.985925	1.892177	−0.093748
CIFAR-10	DeiT-Tiny/16	2027	1.998671	1.919827	−0.078843
CIFAR-10	DeiT-Tiny/16	3407	1.973142	1.874697	−0.098446
ImageNet-R	ViT-B/16	1993	3.093290	2.994212	−0.099078
ImageNet-R	ViT-B/16	2027	3.101487	3.000253	−0.101233
ImageNet-R	ViT-B/16	3407	3.135295	3.019047	−0.116248
ImageNet-R	DeiT-Tiny/16	1993	4.535248	4.375123	−0.160124
ImageNet-R	DeiT-Tiny/16	2027	4.487192	4.377566	−0.109626
ImageNet-R	DeiT-Tiny/16	3407	4.512109	4.338417	−0.173692

Table 18: Final-checkpoint PGD-20 test-subset cross-entropy on 200 CIFAR-10 or 1,000 ImageNet-R images. Lower is favorable; Ours is lower in 12/12 comparisons.

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	0.933242	0.903262	−0.029981
CIFAR-10	ViT-B/16	2027	0.948539	0.902574	−0.045965
CIFAR-10	ViT-B/16	3407	0.941576	0.907320	−0.034256
CIFAR-10	DeiT-Tiny/16	1993	1.846250	1.709199	−0.137052
CIFAR-10	DeiT-Tiny/16	2027	1.864845	1.743241	−0.121605
CIFAR-10	DeiT-Tiny/16	3407	1.839339	1.683649	−0.155691
ImageNet-R	ViT-B/16	1993	3.406672	3.336383	−0.070289
ImageNet-R	ViT-B/16	2027	3.395622	3.315002	−0.080620
ImageNet-R	ViT-B/16	3407	3.408443	3.327367	−0.081076
ImageNet-R	DeiT-Tiny/16	1993	4.603008	4.427817	−0.175191
ImageNet-R	DeiT-Tiny/16	2027	4.542414	4.414336	−0.128078
ImageNet-R	DeiT-Tiny/16	3407	4.572940	4.379062	−0.193878

Table 19: Full-test PGD-20 accuracy (%) on 10,000 CIFAR-10 or 6,000 ImageNet-R images. Higher is favorable; Ours is higher in 12/12 comparisons. Differences are percentage points.

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	63.40	64.94	+1.54
CIFAR-10	ViT-B/16	2027	63.14	65.27	+2.13
CIFAR-10	ViT-B/16	3407	63.37	65.21	+1.84
CIFAR-10	DeiT-Tiny/16	1993	31.15	35.99	+4.84
CIFAR-10	DeiT-Tiny/16	2027	30.52	34.68	+4.16
CIFAR-10	DeiT-Tiny/16	3407	32.17	36.68	+4.51
ImageNet-R	ViT-B/16	1993	32.47	34.07	+1.60
ImageNet-R	ViT-B/16	2027	32.95	34.47	+1.52
ImageNet-R	ViT-B/16	3407	32.70	34.37	+1.67
ImageNet-R	DeiT-Tiny/16	1993	12.30	14.55	+2.25
ImageNet-R	DeiT-Tiny/16	2027	13.00	14.83	+1.83
ImageNet-R	DeiT-Tiny/16	3407	12.63	15.17	+2.53

Table 20: Full-test APGD accuracy (%) on 10,000 CIFAR-10 or 6,000 ImageNet-R images. Higher is favorable; Ours is higher in 12/12 comparisons. Differences are percentage points.

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	63.39	64.91	+1.52
CIFAR-10	ViT-B/16	2027	63.13	65.21	+2.08
CIFAR-10	ViT-B/16	3407	63.33	65.21	+1.88
CIFAR-10	DeiT-Tiny/16	1993	31.08	35.94	+4.86
CIFAR-10	DeiT-Tiny/16	2027	30.42	34.66	+4.24
CIFAR-10	DeiT-Tiny/16	3407	32.13	36.65	+4.52
ImageNet-R	ViT-B/16	1993	31.80	33.72	+1.92
ImageNet-R	ViT-B/16	2027	32.55	34.15	+1.60
ImageNet-R	ViT-B/16	3407	32.65	33.87	+1.22
ImageNet-R	DeiT-Tiny/16	1993	12.28	14.55	+2.27
ImageNet-R	DeiT-Tiny/16	2027	13.02	14.78	+1.77
ImageNet-R	DeiT-Tiny/16	3407	12.62	15.15	+2.53

Table 21: Per-image global gradient-conflict frequency (%) on the same 200/1,000-image test subsets as Table 18. Lower is favorable; Ours is lower in 9/12 comparisons and higher in 3/12. Differences are percentage points.

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	35.5	35.0	−0.5
CIFAR-10	ViT-B/16	2027	41.5	36.0	−5.5
CIFAR-10	ViT-B/16	3407	33.0	33.5	+0.5
CIFAR-10	DeiT-Tiny/16	1993	37.5	35.5	−2.0
CIFAR-10	DeiT-Tiny/16	2027	38.0	40.5	+2.5
CIFAR-10	DeiT-Tiny/16	3407	37.0	32.5	−4.5
ImageNet-R	ViT-B/16	1993	40.1	39.5	−0.6
ImageNet-R	ViT-B/16	2027	41.3	39.0	−2.3
ImageNet-R	ViT-B/16	3407	39.3	39.2	−0.1
ImageNet-R	DeiT-Tiny/16	1993	44.2	44.4	+0.2
ImageNet-R	DeiT-Tiny/16	2027	45.4	44.4	−1.0
ImageNet-R	DeiT-Tiny/16	3407	44.9	43.7	−1.2

Table 22: Original-pair full-test sharing S_{P_0} on 10,000 CIFAR-10 or 6,000 ImageNet-R images. Higher is favorable; Ours is higher in 8/12 comparisons under each attack. Each panel retains all four lower-sharing outcomes. These comparisons are relative to uniform and do not imply an increase over the rank-zero reference.

(a) PGD

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	0.276430	0.251751	−0.024678
CIFAR-10	ViT-B/16	2027	0.207458	0.212524	+0.005066
CIFAR-10	ViT-B/16	3407	0.219350	0.183415	−0.035935
CIFAR-10	DeiT-Tiny/16	1993	−0.044284	−0.039048	+0.005236
CIFAR-10	DeiT-Tiny/16	2027	−0.016731	−0.010998	+0.005733
CIFAR-10	DeiT-Tiny/16	3407	−0.056005	−0.040481	+0.015524
ImageNet-R	ViT-B/16	1993	0.128772	0.124993	−0.003779
ImageNet-R	ViT-B/16	2027	0.123624	0.120258	−0.003366
ImageNet-R	ViT-B/16	3407	0.118090	0.120293	+0.002203
ImageNet-R	DeiT-Tiny/16	1993	0.043449	0.057900	+0.014451
ImageNet-R	DeiT-Tiny/16	2027	0.046426	0.057944	+0.011518
ImageNet-R	DeiT-Tiny/16	3407	0.046521	0.062585	+0.016064

(b) APGD

Dataset	Backbone	Seed	Uniform	Ours	Δ
CIFAR-10	ViT-B/16	1993	0.276420	0.251748	−0.024671
CIFAR-10	ViT-B/16	2027	0.207427	0.212510	+0.005082
CIFAR-10	ViT-B/16	3407	0.219337	0.183394	−0.035944
CIFAR-10	DeiT-Tiny/16	1993	−0.044329	−0.039081	+0.005248
CIFAR-10	DeiT-Tiny/16	2027	−0.016812	−0.011003	+0.005809
CIFAR-10	DeiT-Tiny/16	3407	−0.056039	−0.040491	+0.015548
ImageNet-R	ViT-B/16	1993	0.127359	0.123789	−0.003570
ImageNet-R	ViT-B/16	2027	0.122282	0.119306	−0.002976
ImageNet-R	ViT-B/16	3407	0.118049	0.119137	+0.001088
ImageNet-R	DeiT-Tiny/16	1993	0.043354	0.057810	+0.014456
ImageNet-R	DeiT-Tiny/16	2027	0.046334	0.057872	+0.011538
ImageNet-R	DeiT-Tiny/16	3407	0.046429	0.062501	+0.016072

A.10 LAYER-WISE CONFLICT WITH GLOBAL ATTRIBUTION SHARING

Tables 23 and 24 expand Table 2 to all 12 blocks, combining Q/V gradients. Entries average seed percentages for 1993, 2027, and 3407 equally, then round to one decimal. Both adapted methods use the tied-Q/V budget $\sum_{\ell=1}^{12} r_\ell = 96$ (uniform rank 8) and start from a clean-warmed classifier, separately from the rank-zero scoring reference.

Matched diagnostic setup. Sharing $S = S_{P_0}$ in Eq. 17 uses the original fixed ordered positive class pairs. Test image identities/views, attack settings, and starting seeds match across methods; each checkpoint uses its own cached attacks and global sharing gradient. FP64 differentiation holds attacked pixels, LoRA factors, and all non-Q/V parameters fixed.

Layer-wise percentages. Restricting the gradients in Table 2 to block ℓ's concatenated Q/V weights gives $g_{i,\ell}$ and $G_{S,\ell}$. For each seed, conflict is

$$100\,\frac{\#\{i \in \mathcal{V}_\ell : \cos(-g_{i,\ell}, G_{S,\ell}) < 0\}}{|\mathcal{V}_\ell|}, \tag{18}$$

where $\mathcal{V}_\ell$ contains images with a defined cosine (all evaluated images here). The overall statistic concatenates all blocks before computing image-level cosines; it does not average block percentages. Improvements vary by block. These local gradient diagnostics share the subset used to define S and do not measure actual LoRA updates or establish a causal robustness gain.

Table 23: CIFAR-10: per-image conflict with global attribution sharing (%) by block, using combined Q/V gradients. Three-seed means; lower values indicate less conflict.

	DeiT-Tiny			ViT-B/16		
Block	Rank zero	Uniform	Ours	Rank zero	Uniform	Ours
1	53.8	40.5	45.2	39.8	43.0	41.2
2	51.7	39.3	38.3	49.7	38.2	36.8
3	49.2	41.7	35.2	47.7	37.3	36.8
4	51.7	38.7	39.2	47.0	36.7	36.5
5	54.2	41.7	40.2	47.0	36.7	36.8
6	52.5	38.3	40.3	52.0	35.8	34.3
7	53.0	38.8	38.7	53.2	40.3	40.2
8	51.8	39.0	37.2	56.7	37.7	35.7
9	49.7	42.7	41.5	51.7	36.7	36.0
10	49.5	40.8	43.8	49.5	42.5	43.7
11	53.7	42.5	41.2	46.0	41.8	43.7
12	48.8	46.8	47.5	46.8	39.0	42.7

Table 24: ImageNet-R: per-image conflict with global attribution sharing (%) by block, using combined Q/V gradients. Three-seed means; lower values indicate less conflict.

	DeiT-Tiny			ViT-B/16		
Block	Rank zero	Uniform	Ours	Rank zero	Uniform	Ours
1	77.8	45.9	43.8	23.3	42.9	42.4
2	58.8	47.7	47.0	29.4	44.0	43.5
3	54.6	44.9	44.6	39.1	44.6	44.0
4	57.6	44.8	44.3	37.2	43.9	43.5
5	57.8	46.3	44.5	39.4	45.2	44.9
6	73.3	47.1	47.1	37.4	43.1	42.9
7	58.0	46.7	45.9	37.5	41.8	41.8
8	65.5	45.7	44.7	36.3	41.0	41.6
9	53.1	44.4	43.6	35.8	44.5	42.7
10	55.9	41.1	39.6	40.2	39.1	38.0
11	73.5	38.9	38.7	44.5	40.9	39.4
12	67.7	37.0	37.7	40.3	37.9	37.6

A.11 EXTENDED ADVERSARIAL TRAINING: IMPLEMENTATION DETAILS

We adversarially fine-tune the query and value projections of all 12 attention blocks using LoRA while keeping the remaining backbone parameters frozen. The classifier head is trained jointly with LoRA. This diagnostic uses uniform rank 64 and the setup in Table 25.

Table 25: Implementation Details of extended adversarial training.

Component	Setting
Dataset	ImageNet-R, 200 classes; 24,000 training and 6,000 test images.
Backbone	ViT-B/16 initialized from an ARES checkpoint.
LoRA	Rank $r = 64$ and $\alpha = 128$ for the query and value projections of every attention block; the classifier head is also trainable.
Training objective	Cross-entropy on PGD-20 adversarial examples; ℓ_∞ radius $\epsilon = 8/255$, step size 0.004, and random start.
Optimization	200 epochs; batch size 48; SGD with momentum 0.9 and weight decay 5×10^{-4}. The effective LoRA/head learning rate is 0.003, with cosine decay to zero.
Per-epoch evaluation	Clean and PGD-20 evaluation at $\epsilon = 8/255$ on a fixed 600-image test subset sampled once with seed 1993.

Evaluation and reporting. The test-loss curves and accuracy summaries refer to the same fixed 600-image subset of the 6,000 test images. Seed 1993 specifies the subset selection. The loss and accuracy panels in Figure 1(a) overlay two test metric curves on one shared axis. Both curves use that same y-axis scale.

Test deterioration is measured against the best value reached up to the current epoch, including that epoch:

$$D_L(t) = L_{\text{test}}(t) - \min_{1 \le s \le t} L_{\text{test}}(s),$$
$$D_A(t) = \max_{1 \le s \le t} A_{\text{test}}(s) - A_{\text{test}}(t).$$

Both measures are zero at a new best value and positive for deterioration. Accuracy differences are in percentage points. Every curve plots all 200 unsmoothed per-epoch differences against the fraction of total epochs, $t/200$. At the end of training, the loss gap is -0.763, the accuracy gap is 11.78 percentage points, and the test deteriorations are 0.222 in cross-entropy and 2.17 percentage points in accuracy.

For the robust test-accuracy summary, we average the reported PGD-20 accuracies equally over the 20 epochs in each of the windows 31–50 and 181–200. These means are 40.9005% and 40.0080%, respectively, giving a decrease of 0.8925 percentage points before rounding.

The adversarial test cross-entropy reaches 2.767 at epochs 29, 32, and 35, then increases to 2.989 at epoch 200. Accordingly, the observation of limited robust-accuracy deterioration is compatible with some overfitting in the adversarial loss.

A.12 ATTRIBUTION SHARING ACROSS RANKS.

Table 26 compares the rank-zero, head-only model with uniform rank-8 LoRA under the same adversarial training protocol. These are separately trained branches from the same clean-warmed head. We use the class attribution vectors of Wei et al. (2025), formed from each class's mean test feature $\bar{h}_c$ and the corresponding row of that model's learned classifier. We report the positive-pair mean:

$$a_c = \bar{h}_c \odot C_{c,:}, \qquad \rho_{cd} = \frac{a_c^\top a_d}{\|a_c\|_2 \|a_d\|_2},$$
$$\mathcal{P}_+ = \{(c,d) : c \neq d,\ \rho_{cd} > 0\}, \qquad S_+ = \frac{\sum_{(c,d)\in\mathcal{P}_+} \rho_{cd}}{\max\{1, |\mathcal{P}_+|\}}. \tag{19}$$

This normalizes the positive CAS sum by the number of strictly positive ordered class pairs, with zero assigned when none exist. Class means use true labels and all test examples; classifier bias

is excluded. Attacks are generated separately for each evaluated model. The positive-pair set is recomputed for each model and test condition, so S_+ measures mean alignment conditional on a pair being positive. Mean S_+ decreases in all 12 settings, by 7.43%–77.42%. These descriptive comparisons do not establish an effect on robust accuracy.

Ours uses sharing to choose an allocation: it fixes the positive class pairs of the rank-zero adversarial training reference, differentiates their mean cosine with respect to Q/V weights, and allocates ranks inversely to each block's sum of mean absolute weight–gradient products. Here, S_+ evaluates test representations after training with uniform ranks; its positive pairs are recomputed rather than fixed, and no gradient enters the reported score.

Table 26: Positive-pair mean attribution sharing S_+ at ranks 0 and 8 and its relative drop. Both models use standard-pretrained backbones and PGD adversarial training at $\epsilon = 3/255$; attacked evaluation uses the same radius. Rank 8 is uniform LoRA. Entries are means over seeds 1993, 2027, and 3407 on the full test splits. Scores retain their cosine scale and are shown to four decimals. Drop is $100(\bar{S}_{+,0} - \bar{S}_{+,8})/\bar{S}_{+,0}$, computed before rounding.

Dataset	Backbone	Evaluation	Rank 0	Rank 8	Drop (%)
CIFAR-10	ViT-B/16	PGD	0.4051	0.2421	40.23
		APGD	0.2615	0.2421	7.43
		Clean	0.6482	0.2699	58.37
CIFAR-10	DeiT-Tiny/16	PGD	0.2831	0.1404	50.40
		APGD	0.2570	0.1404	45.38
		Clean	0.5870	0.1382	76.45
ImageNet-R	ViT-B/16	PGD	0.5999	0.1355	77.42
		APGD	0.5720	0.1349	76.42
		Clean	0.5065	0.1690	66.63
ImageNet-R	DeiT-Tiny/16	PGD	0.3088	0.1058	65.75
		APGD	0.2527	0.1057	58.15
		Clean	0.3553	0.1249	64.85

A.13 Impact Statement and Limitations

This work improves the reliability of parameter-efficient fine-tuning (PEFT) for vision foundation models under adversarial perturbations. By enabling stronger robustness with a fixed PEFT budget, the proposed allocation method can reduce the cost of deploying more secure models in resource-constrained settings and may benefit safety-critical applications where robustness is required.

Potential negative impacts include misuse by adversaries to harden malicious models against defensive attacks or auditing, and the risk that robustness gains may not transfer to unseen threat models, datasets, or deployment shifts, leading to overconfidence if used without proper evaluation. Our method relies on gradient-based importance estimates that could inherit biases from the underlying data and training setup; thus, we encourage reporting robustness across diverse attacks and data distributions, and treating the sensitivity score as a diagnostic rather than a guarantee of security.