A TRAINING-DYNAMICS VIEW OF CATASTROPHIC OVERFITTING: UNDERSTANDING AND PREVENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the underlying mechanism of catastrophic overfitting, a phenomenon in which models overfit to weak adversarial examples and lose true robustness, from the perspective of training dynamics. While prior works explain *when* catastrophic overfitting occurs through input-space quantities, the mechanism by which parameter updates produce the sudden collapse remains unstudied. We study the collapse *where* it happens—in *parameter space*: catastrophic overfitting is a failure of *training*, and what corrupts training is the parameter update; we therefore analyze the *mixed Hessian*, which carries an input-space perturbation error into the parameter-update error. We provide analytical and interventional evidence that rapid amplification of the mixed Hessian drives the parameter-update misalignment underlying catastrophic overfitting, and show that directly suppressing its spectral norm prevents the collapse. Based on this, we propose a KL divergence-based regularizer that stabilizes training dynamics and prevents catastrophic overfitting. Our method achieves the strongest robustness among single-step methods and, when combined with multi-step methods, further improves robustness, indicating that mixed-curvature stabilization is a general principle rather than a single-step remedy.

1 INTRODUCTION

Adversarial vulnerability (Szegedy et al., 2013; Goodfellow et al., 2014; Madry et al., 2018) remains a critical threat to deploying deep neural networks reliably. Adversarial attacks produce small, even imperceptible, perturbations to input data that cause models to predict incorrectly. To defend against such perturbations, Adversarial Training (AT) (Goodfellow et al., 2014; Madry et al., 2018) has emerged as one of the most effective defenses. By generating adversarial examples on-the-fly during training and optimizing model parameters on these examples, AT significantly improves robustness.

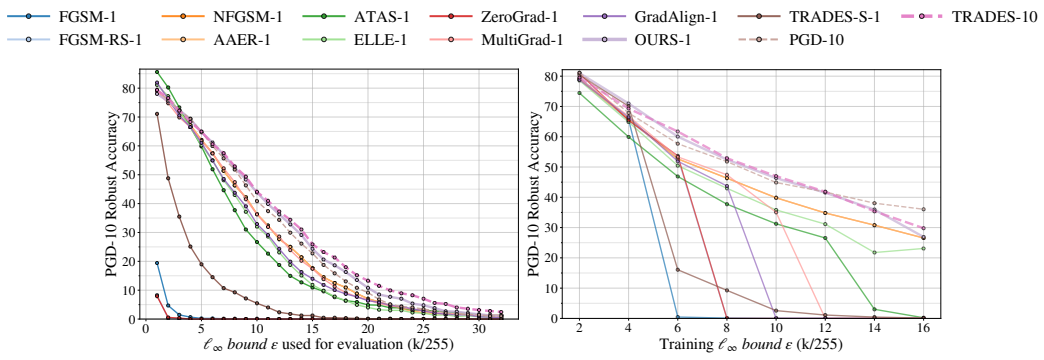
To propose an efficient and effective AT, a line of work investigated methods based on the *Fast Gradient Sign Method (FGSM)* (Wong et al., 2020; Shafahi et al., 2019). They investigated training a robust model on single-step adversarial examples. However, a critical failure mode, *"catastrophic overfitting"* (Wong et al., 2020) emerges, robust accuracy against *Projected Gradient Descent (PGD)* (Madry et al., 2018) rapidly drops to 0%, whereas robust accuracy against FGSM (Goodfellow et al., 2014) strictly increases, indicating the model overfit to the weak FGSM adversaries constructed on-the-fly at each training step, rather than learning genuinely robust features.

Several methods have been proposed to mitigate catastrophic overfitting (Andriushchenko & Flammarion, 2020; Kim et al., 2021; Li et al., 2022; de Jorge Aranda et al., 2022; Huang et al., 2023; Lin et al., 2023; Rocamora et al., 2024; Golgooni et al., 2021), and they characterize *when* the collapse occurs, through input-space quantities. The complementary question—*how* an input-space pathology propagates into the parameter updates that actually produce the collapse—has received little attention. This matters because **catastrophic overfitting is a failure of *training*, and what corrupts training is the parameter update.** This motivates us to investigate the following questions:

Which changes in training dynamics trigger catastrophic overfitting during single-step AT?
How can we efficiently prevent catastrophic overfitting and improve adversarial robustness?

The operator that carries an input-space attack error into the parameter update is the *mixed Hessian* $H_{x\theta}$, and it is the lens we propose. It characterizes how input perturbations influence parameter

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.



(a) Robustness under increasing evaluation ϵ (b) Robustness across diverse training ϵ

Figure 1: **PGD-10 robust accuracy and catastrophic overfitting across evaluation and training radii.** Solid lines denote single-step methods, while dashed lines denote multi-step methods. (a) We train each method with fixed perturbation radius ϵ as 8/255 and evaluate with various ϵ. The suffix -n indicates the number of adversarial gradient steps used to generate perturbation. Several single-step methods suffer catastrophic overfitting; ours prevents it, and improving PGD accuracy. (b) We vary the training radius (from $2/255$ to $16/255$) and evaluate at the corresponding training radius. This setting directly tests whether each method remains stable across different training configurations; our method maintains strong robustness without catastrophic overfitting across the various radii.

Table 1: **Which curvature block does each method control?** Prior single-step defenses regularize input-space quantities, which are proxies for the input Hessian H_{xx}; none reach the mixed block $H_{x\theta}$ that converts input-space error into parameter-update error.

Method	Regularized quantity	H_{xx}	$H_{x\theta}$	$H_{\theta\theta}$	PGD-10
ZeroGrad (Golgooni et al., 2021)	input-gradient sparsification	✓	✗	✗	0.00 ± 0.00
TRADES-S (Zhang et al., 2019)	input-space consistency	✓	✗	✗	8.91 ± 1.51
ELLE (Rocamora et al., 2024)	local linearity of $\mathcal{L}$ in x	✓	✗	✗	42.19 ± 0.04
GradAlign (Andriushchenko & Flammarion, 2020)	input-gradient alignment	✓	✗	✗	43.44 ± 0.86
Input consistency only (section 5.3 and Appendix C)	$\mathrm{KL}(p_{\eta_x}(y) \,\|\, p_0(y))$	✓	✗	✗	47.10 ± 0.31
Mixed Hessian only (section 4.4)	mixed Hessian $H_{x\theta}$	✗	✓	✗	47.44 ± 1.21
Ours	joint input–parameter consistency	✓	✓	✓	$\mathbf{53.00 \pm 0.52}$

gradients. We provide analytical and interventional evidence that rapid amplification of $\sigma_1(H_{x\theta})$ drives the parameter-update misalignment underlying catastrophic overfitting, and that suppressing it directly prevents the collapse. Table 1 summarizes the resulting picture: prior works control the input block H_{xx}, and **which curvature block a method controls bounds its robustness.**

We then propose a *KL divergence-based consistency regularizer* that stabilizes training dynamics by directly suppressing the joint curvature—and thereby the mixed Hessian—preventing catastrophic overfitting and improving robustness (see fig. 1).

2 Related Works

2.1 Adversarial Robustness

Adversarial attack was first investigated by Szegedy et al. (2013) and was broadly investigated over a decade (Madry et al., 2018; Goodfellow et al., 2014; Lee et al., 2020; 2021; Schmidt et al., 2018; Tramer et al., 2018; Kim et al., 2020; Zhu et al., 2023; Li & Li). Despite this progress, training models reliably robust to adversarial attacks remains an open problem. Among existing approaches, PGD-training (Madry et al., 2018) has emerged as one of the most reliable empirical directions:

$$\min_{\theta} \mathbb{E}_{x,y} \left[\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(x+\delta, y; \theta) \right]. \tag{1}$$

However, solving the inner maximization problem is computationally expensive. This has motivated a line of work on more efficient AT methods (Shafahi et al., 2019; Wong et al., 2020). Although these methods substantially reduce training cost, they are known to suffer from *catastrophic overfitting*.

2.2 CATASTROPHIC OVERFITTING

Wong et al. (2020) observed that during single-step AT, robust accuracy against PGD (Madry et al., 2018) rapidly drops to 0% over a few epochs, whereas accuracy against FGSM (Goodfellow et al., 2014) increases to nearly 100%. This phenomenon was termed *catastrophic overfitting*. To mitigate this issue, Wong et al. (2020) suggested adding a random step to FGSM and employing an early-stopping scheme. Similarly, de Jorge Aranda et al. (2022) also explored input randomization strategies. However, early-stopping does not fundamentally mitigate catastrophic overfitting. Moreover, Andriushchenko & Flammarion (2020) argued that the benefit of adding a random step mainly comes from reducing the effective perturbation magnitude.

Therefore, many lines of work investigated to mitigate catastrophic overfitting (Andriushchenko & Flammarion, 2020; Kim et al., 2021; Li et al., 2022; de Jorge Aranda et al., 2022; Huang et al., 2023; Lin et al., 2023; Rocamora et al., 2024; Golgooni et al., 2021). Notably, *"GradAlign"* (Andriushchenko & Flammarion, 2020), regularizes gradient similarity between pairs of nearby points in the input space. Related strategies that aim to preserve local linearity have been explored in Rocamora et al. (2024). However, existing approaches regularize input space gradients, whereas our analysis focuses on *joint input-parameter curvature* through the *mixed Hessian dynamics.*

Another line of research, Kim et al. (2021) observed that the worst-case adversarial examples do not lie at the maximum perturbation magnitude. Based on this observation, they proposed *"FGSM-CKPT"*, performs a line search over perturbation magnitudes for each adversarial example. Related adaptive-perturbation strategies have been further explored in Huang et al. (2023).

Positioning: *when* versus *how*. These lines of work characterize catastrophic overfitting through **input-space** quantities and answer *when* it occurs: above which perturbation radius, at which point local linearity drops, under which input-gradient condition the single-step adversary ceases to approximate the worst case. Meanwhile, we address a different question: *how* an **input-space** pathology propagates into the **parameter updates** that produce the collapse. Because catastrophic overfitting is a failure of training, that propagation must be carried by an operator mapping input-space error to parameter-space error, and eq. (6) shows this operator to be the mixed Hessian $H_{x\theta}$. The two questions are complementary, and table 1 contrasts which curvature each line of work controls.

3 BACKGROUND

We consider an image classification model f_θ that maps an input $x \in \mathbb{R}^{d_x}$ to logits $z = f_\theta(x) \in \mathbb{R}^{|\mathcal{C}|}$, where $\mathcal{C}$ denotes the set of classes. We denote the predictive class probabilities by $p_\theta(y|x) = \mathrm{softmax}(f_\theta(x))_y$. Let $\ell(z, y)$ denote the cross-entropy loss applied to the logits. The overall loss can be written as $\mathcal{L}(x, y; \theta) = -\log p_\theta(y|x) = \ell(f_\theta(x), y)$. The model is trained using Stochastic Gradient Descent (SGD). We denote the parameter update at step t as $\Delta_t = \theta_{t+1} - \theta_t$.

We define three Hessians as follows:

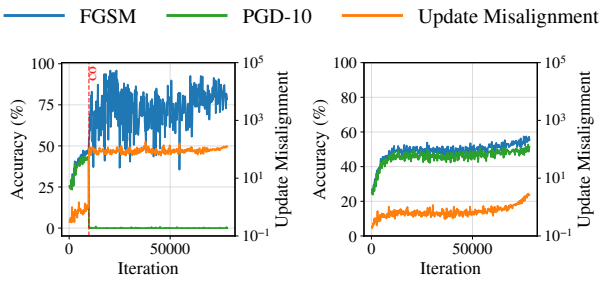
$$H_{x\theta} = \frac{\partial^2 \mathcal{L}}{\partial \theta \, \partial x} \in \mathbb{R}^{d_\theta \times d_x}, \; H_{xx} = \frac{\partial^2 \mathcal{L}}{\partial x^2} \in \mathbb{R}^{d_x \times d_x}, \; H_{\theta\theta} = \frac{\partial^2 \mathcal{L}}{\partial \theta^2} \in \mathbb{R}^{d_\theta \times d_\theta}. \tag{2}$$

The mixed Hessian $H_{x\theta}$ measures how the parameter gradient changes with respect to the input. Unless otherwise specified, $H_{x\theta}$ denotes $H_{x\theta}(x + \delta_t, y; \theta_t)$.

In FGSM-training, a single-step perturbation is constructed as $\delta_t = \epsilon \cdot \mathrm{sgn}(\nabla_x \mathcal{L}(x, y; \theta_t))$, where the subscript t denotes the training step. The model parameters are updated by minimizing the loss at the perturbed input $x + \delta_t$. Accordingly, the parameter update direction is given by the gradient

$$\nabla_{\theta_t} \mathcal{L}(x + \delta_t, y; \theta_t), \tag{3}$$

up to a learning rate factor. In contrast, PGD-training employs a multi-step adversarial perturbation δ_t^*. Starting from $\delta_t^{(0)} = 0$, PGD iteratively updates $\delta_t^{(k+1)} = \Pi_{\|\delta\|_\infty \leq \epsilon}(\delta_t^{(k)} + \alpha \cdot \mathrm{sgn}(\nabla_x \mathcal{L}(x +$



(a) FGSM-training (b) PGD-training

Figure 2: **Parameter update misalignment leads to catastrophic overfitting.** FGSM/PGD-accuracy (blue/green line), and parameter update misalignment (orange line) during (a) FGSM and (b) PGD-training. Catastrophic overfitting emerges at iteration 9800 in (a), which is indicated as a red dashed vertical line (denoted as CO in the figure). Catastrophic overfitting coincides *with* parameter update misalignment.

$\delta_t^{(k)}, y; \theta_t)))$, and we denote the final perturbation after K steps as $\delta_t^* = \delta_t^{(K)}$. The corresponding parameter update direction during PGD-training is given by the gradient

$$\nabla_{\theta_t} \mathcal{L}(x + \delta_t^*, y; \theta_t). \tag{4}$$

We denote $\sigma_1(A)$ as the spectral norm (the largest singular value) of a matrix A. Let $A = \sum_i \sigma_i u_i v_i^\top$ be its singular value decomposition, where u_i, v_i denotes the left, right singular vectors, respectively. In particular, $v_1(A)$ denotes the top singular vector.

4 Training Dynamics and Mixed Hessian

This section establishes the mechanism, in the following order:

$$\underbrace{\|\nabla_\theta \mathcal{L}(x+\delta_t)\| \uparrow \longleftrightarrow \underbrace{\sigma_1(H_{x\theta}) \uparrow}_{\text{sections 4.2 and 4.4}} \longrightarrow \underbrace{\|\Delta_t^{\mathrm{UM}}\| \uparrow}_{\text{section 4.1}}}_{\text{positive feedback loop (section 4.3)}} \longrightarrow \textbf{catastrophic overfitting} \tag{5}$$

Section 4.1 derives the misalignment identity and shows it leads to catastrophic overfitting. Section 4.2 decomposes the misalignment into multiplicative factors and identifies $\sigma_1(H_{x\theta})$ as the dominant one. Section 4.3 verifies the feedback loop with *interventions*. Section 4.4 then tests the mechanism—suppressing $\sigma_1(H_{x\theta})$ alone—and asks whether that is enough to prevent the collapse.

The organizing claim is that the mixed block is the central operator of eq. (5). It is what converts an input-space attack error into a **parameter-space update** error, so a *dynamics* account of catastrophic overfitting requires it, and no input-space criterion contains it.

4.1 Parameter update misalignment leads to Catastrophic Overfitting

First, we investigate the parameter update direction misalignment. This misalignment between FGSM/PGD-training induced by eqs. (3) and (4), is governed by two factors:

$$\nabla_\theta \mathcal{L}(x + \delta_t, y; \theta_t) - \nabla_\theta \mathcal{L}(x + \delta_t^*, y; \theta_t) \;\approx\; H_{x\theta}(\delta_t - \delta_t^*). \tag{6}$$

Assume that PGD-training produces a worst-case adversarial perturbation δ^*. Robust optimization relies on alignment between the parameter update directions induced by FGSM and PGD perturbations. When this alignment collapses, **parameter updates are driven toward FGSM-specific directions instead of worst-case adversarial directions, leading to catastrophic overfitting.**

Equation (6) reveals that this update direction misalignment is governed by two factors: (i) the perturbation misalignment $(\delta_t - \delta_t^*)$ and (ii) the mixed Hessian operator $H_{x\theta}$. While prior works (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024) primarily investigates the perturbation

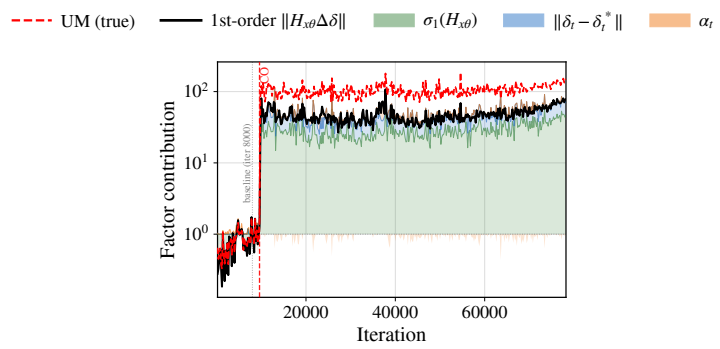


Figure 3: **Decomposition of catastrophic overfitting.** The y-axis shows the multiplicative ratio of each quantity relative to a baseline iteration just before catastrophic overfitting (gray dotted vertical line). By the multiplicative decomposition in eq. (8), the first-order term factorizes into $\sigma_1(H_{x\theta})$, $\|\delta_t - \delta_t^*\|$, and α_t, shown as stacked log-area bands. Among the three factors, $\sigma_1(H_{x\theta})$ amplifies by orders of magnitude near catastrophic overfitting, far exceeding the other two.

misalignment via input-space analysis—essentially proxies for the input Hessian H_{xx}—**our analysis identifies a different object—the *mixed Hessian* $H_{x\theta}$**—and explicitly separates and measures both factors, revealing their coupled role in catastrophic overfitting.

Figure 2 illustrates the training dynamics of FGSM/PGD-training. At the onset of catastrophic overfitting (fig. 2 (a) iteration 9800), the parameter update misalignment sharply increases. In contrast, the parameter update directions stay well-aligned during PGD-training. These observations realize the mechanism derived in eq. (6): **update misalignment leads to catastrophic overfitting.**

4.2 Mixed Hessian is the Dominant Factor

In eq. (6), we show that the update misalignment is governed by the mixed Hessian $H_{x\theta}$ and the perturbation mismatch $\delta_t - \delta_t^*$. To identify the *dominant* factor driving catastrophic overfitting, we expand the update misalignment into its constituent components and empirically track each one during training. Taking norms of both sides of eq. (6) and applying the Taylor expansion explicitly:

$$\underbrace{\|\nabla_\theta \mathcal{L}(x+\delta_t) - \nabla_\theta \mathcal{L}(x+\delta_t^*)\|}_{\text{update misalignment (UM)}} = \underbrace{\|H_{x\theta}(\delta_t - \delta_t^*)\|}_{\text{first-order term}} + O(\|\delta_t - \delta_t^*\|^2), \tag{7}$$

where the first-order term admits the multiplicative decomposition

$$\|H_{x\theta}(\delta_t - \delta_t^*)\| = \alpha_t \cdot \sigma_1(H_{x\theta}) \cdot \|\delta_t - \delta_t^*\|, \tag{8}$$

with $\alpha_t \in [0,1]$ denoting the alignment between $\delta_t - \delta_t^*$ and $v_1(H_{x\theta})$. The update misalignment thus decomposes into three factors—$\sigma_1(H_{x\theta})$, $\|\delta_t - \delta_t^*\|$, and α_t—plus a higher-order residual.

Figure 3 visualizes the resulting decomposition: the first-order term closely tracks the true update misalignment, confirming the validity of the Taylor expansion. Among the multiplicative factors, $\sigma_1(H_{x\theta})$ amplifies by orders of magnitude near the onset of catastrophic overfitting, while $\|\delta_t - \delta_t^*\|$ and α_t grow at substantially smaller rates. The decomposition therefore identifies $\sigma_1(H_{x\theta})$ as the dominant factor of update misalignment. We conclude that the **rapid spectral amplification of the mixed Hessian is the central driver of catastrophic overfitting** in single-step AT.

4.3 The Amplification is Causal: A Bidirectional Feedback Loop

Figure 3 establishes that the amplification of $\sigma_1(H_{x\theta})$ is the dominant driver of update misalignment, underlying catastrophic overfitting. A natural follow-up question is *why* $\sigma_1(H_{x\theta})$ amplifies. We give a first-order analysis suggesting a positive feedback loop, and then verify that loop by intervention.

A progressive positive feedback between σ_1 and the parameter gradient. A first-order analysis of the mixed Hessian dynamics suggests

$$H_{x\theta}(\,\cdot\,;\theta_{t+1}) \approx H_{x\theta}(\,\cdot\,;\theta_t) - \eta\, \nabla_\theta H_{x\theta}(\,\cdot\,;\theta_t)\, \nabla_\theta \mathcal{L}(x+\delta_t, y; \theta_t), \tag{9}$$

Table 2: **Interventions on the feedback loop of eq. (5).** *Top:* driving either quantity upward raises the other and raises $\|\Delta^{\mathrm{UM}}\|$, while a matched-budget control barely moves any of them. *Bottom:* the intervention with the opposite sign, applied to a checkpoint that has already collapsed—suppressing $\sigma_1(H_{x\theta})$ recovers PGD-10 accuracy. $\|\Delta^{\mathrm{UM}}\|$ indicates update misalignment in eq. (7)

Intervention	$\sigma_1(H_{x\theta})$	$\|\nabla_\theta \mathcal{L}\|$	$\|\Delta^{\mathrm{UM}}\|$	PGD-10
Amplifying (ratio after/before)				
Optimize $\sigma_1(H_{x\theta})$ upward	$\times 8.76$	$\times 44.00$	$\times \mathbf{9.97}$	—
Optimize $\|\nabla_\theta \mathcal{L}\|$ upward	$\times 5.03$	$\times 21.94$	$\times \mathbf{4.34}$	—
Control: training loss only	$\times 0.97$	$\times\ 1.35$	$\times 0.98$	—
Suppressing, on a collapsed checkpoint (absolute values)				
before intervention	3.32	—	—	16%
after suppressing $\sigma_1(H_{x\theta})$	**0.09**	—	—	**59%**

which yields the singular value update

$$\Delta\sigma_i \approx -\eta\, u_i^\top (\nabla_\theta H_{x\theta}(\,\cdot\,;\theta_t)\, \nabla_\theta \mathcal{L}(x+\delta_t, y; \theta_t)) v_i. \tag{10}$$

Furthermore, expanding the parameter gradient at the perturbed input to first order:

$$\nabla_\theta \mathcal{L}(x+\delta_t, y; \theta_t) \approx \nabla_\theta \mathcal{L}(x, y; \theta_t) + H_{x\theta}(x, y; \theta_t)\, \delta_t, \tag{11}$$

where $H_{x\theta}(x, y; \theta_t)\, \delta_t = \sum_i \sigma_i \langle v_i, \delta_t \rangle u_i$. Together, eqs. (10) and (11) are consistent with a progressive positive feedback: an amplification of σ_1 enlarges the parameter gradient via eq. (11), which in turn further amplifies σ_1 via eq. (10).

Intervention: manipulating each quantity directly. Starting from an FGSM training trajectory, we run K gradient steps to increase a target quantity—either $\sigma_1(H_{x\theta})$ or $\|\nabla_\theta \mathcal{L}(x+\delta_t)\|$—with the training loss as an anchor, then measure all quantities of eq. (5) (Full protocol in Appendix B.3).

Table 2 yields four conclusions. First, **the loop runs in both directions**: driving σ_1 up raises $\|\nabla_\theta \mathcal{L}\|$, and driving $\|\nabla_\theta \mathcal{L}\|$ up raises σ_1. Second, **the loop reaches the update misalignment**, which rises under both interventions; the misalignment is thus downstream of the loop rather than merely co-occurring with it. Third, **the effect is attributable to the targeted quantity and not to displacing** θ, since the matched-budget control leaves all three quantities essentially unchanged. We further verify that the perturbation mismatch $\|\delta_t - \delta_t^*\|$ does *not* increase under either intervention (it changes by $\times 0.90$–0.95), so the misalignment grows through the σ_1 factor of eq. (8). Finally, **the loop can be run backwards**: suppressing $\sigma_1(H_{x\theta})$ on a checkpoint that has collapsed recovers PGD-10 accuracy from 16% to 59%, closing the loop in the direction that matters for a defense.

4.4 Suppressing the mixed Hessian alone prevents the collapse

If $\sigma_1(H_{x\theta})$ drives the update misalignment, then acting on it should prevent catastrophic overfitting. We train with a penalty on the mixed spectral norm added to the single-step loss,

$$\mathcal{L}_{\text{mixed}} = \mathcal{L}\big(x+\delta_t, y; \theta\big) + \lambda\, \sigma_1(H_{x\theta}). \tag{12}$$

Table 3 shows this is sufficient: training completes without collapse and reaches 47.44% PGD-10 accuracy. The mechanism therefore supports intervention, not only explanation—**controlling the mixed block that eq. (6) identifies is enough to avert the failure it predicts.**

The penalty requires third-order automatic differentiation at every iteration, which is expensive. More importantly, it addresses only one of the two factors in eq. (8), while prior input-space methods successfully address the other. Section 5 therefore develops an objective that suppresses both.

5 Stabilizing training dynamics via KL divergence consistency

Section 4 show that catastrophic overfitting is driven by the rapid amplification of the mixed Hessian, and the perturbation misalignment. This suggests that preventing catastrophic overfitting requires controlling curvature not only in the input space, but also in the parameter space and their interaction. We now introduce a KL divergence-based consistency regularizer that simultaneously suppresses three curvature components: (i) the mixed Hessian, which we identified as the main driver

Table 3: **Training with the mixed-Hessian penalty alone.** Suppressing $\sigma_1(H_{x\theta})$ is by itself enough to train a robust model: no collapse occurs.

Training objective	FGSM	PGD-10
FGSM	84.36 ± 3.09	0.19 ± 0.10
FGSM + $\sigma_1(H_{x\theta})$ penalty (eq. (12))	52.43 ± 1.96	**47.44 ± 1.21**

of catastrophic overfitting, (ii) the input Hessian, which characterizes local linearity of the model in the input space, and (iii) the parameter Hessian, which reflects the sharpness of the loss-landscape.

5.1 KL DIVERGENCE AS A JOINT CURVATURE REGULARIZER

Define the predictive distribution $p_0(y) = p_\theta(y|x)$, $p_{\eta_x}(y) = p_\theta(y|x+\eta_x)$, $q_{\eta_\theta}(y) = p_{\theta+\eta_\theta}(y|x)$. We consider the KL divergence consistency term

$$\mathrm{KL}(p_{\eta_x} \,\|\, q_{\eta_\theta}) = \sum_y p_{\eta_x}(y) \log \frac{p_{\eta_x}(y)}{q_{\eta_\theta}(y)}. \tag{13}$$

Theorem 1 (KL divergence controls joint curvature) *Assume that the predictive distribution $p_\theta(y|x)$ is positive and twice continuously differentiable. For sufficiently small (η_x, η_θ),*

$$\mathrm{KL}(p_{\eta_x} \,\|\, q_{\eta_\theta}) = \frac{1}{2}\eta^\top F(x,\theta)\eta + o(\|\eta\|^2), \qquad \eta := [\eta_x; \eta_\theta]. \tag{14}$$

where

$$F(x,\theta) = \begin{bmatrix} F_{xx} & F_{x\theta} \\ F_{\theta x} & F_{\theta\theta} \end{bmatrix} \tag{15}$$

is the joint Fisher information matrix of the predictive distribution with respect to (x, θ). Under cross-entropy loss, each block equals the expected loss Hessian block:

$$F_{xx} = \mathbb{E}[\nabla^2_{xx}\mathcal{L}], \quad F_{\theta\theta} = \mathbb{E}[\nabla^2_{\theta\theta}\mathcal{L}], \quad F_{x\theta} = \mathbb{E}[\nabla^2_{x\theta}\mathcal{L}]. \tag{16}$$

The proof is deferred to Appendix A. This result shows that minimizing a single KL divergence consistency term locally suppresses three curvature simultaneously.

These three forms can also be tracked during training. In Appendix C.4 we measure the input, parameter and mixed consistency terms along FGSM/PGD-training: all three grow, consistent with progressive loss sharpening (Cohen et al., 2021; Jastrzebski et al., 2019) and input curvature growth (Andriushchenko & Flammarion, 2020), but under FGSM-training the *mixed* term amplifies sharply at the onset of catastrophic overfitting, whereas PGD-training keeps all three stable.

5.2 METHOD: KL DIVERGENCE-BASED CURVATURE REGULARIZATION

By theorem 1, the mixed Hessian $H_{x\theta}$—identified in section 4 as the primary driver of catastrophic overfitting—can be suppressed in expectation by minimizing the KL divergence $\mathrm{KL}(p_{\eta_x} \| q_{\eta_\theta})$. We therefore augment single-step adversarial training with a KL divergence consistency regularizer

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{adv}} + \lambda\, \mathrm{KL}\big(p_{\eta_x}(y) \,\|\, q_{\eta_\theta}(y)\big), \tag{17}$$

where η_x and η_θ are small perturbations sampled in the input and parameter space, respectively; their scales and the KL weight are given in Appendix B and C. Unlike prior works control input sensitivity (Andriushchenko & Flammarion, 2020; Rocamora et al., 2024), this regularizer directly targets joint input-parameter curvature, which we identified as a key driver of catastrophic overfitting.

Difference between TRADES (Zhang et al., 2019). Our objective is related in form to TRADES:

$$\min_\theta \mathbb{E}\left[\phi(p_0(y)) + \max_{\|\delta\|_\infty \le \epsilon} \phi(p_0(y), p_{\eta_x}(y))\right], \tag{18}$$

where ϕ is a non-negative loss function. However, TRADES enforces consistency under input perturbations only and does not explicitly regularize mixed consistency. Its success in preventing catastrophic overfitting comes from the multi-step inner maximization: when we replace it with a single-step perturbation (**TRADES-S**), catastrophic overfitting still occurs (fig. 4a, around iteration 9200),

Table 4: **Single-step methods.** FGSM, PGD-10 and AutoAttack robust accuracy (%) and training time. Best PGD-10 and AutoAttack in bold. FGSM accuracy is not indicative of true robustness: catastrophic overfitting yields artificially high FGSM accuracy while PGD accuracy collapses.

Method	CIFAR-10			CIFAR-100			Time (h)
	FGSM	PGD-10	AutoAttack	FGSM	PGD-10	AutoAttack	
FGSM	84.36 ± 3.09	0.19 ± 0.10	0.00	58.93	0.11	0.00	1.03
FGSM-RS	87.70 ± 1.55	0.00 ± 0.00	0.00	66.63	0.00	0.00	1.00
GradAlign	54.34 ± 0.88	43.44 ± 0.86	38.90	26.67	19.87	17.05	3.46
ZeroGrad	91.28 ± 3.98	0.00 ± 0.00	0.00	75.00	0.00	0.00	1.14
MultiGrad	54.53 ± 0.30	44.95 ± 0.28	40.37	25.78	25.54	17.84	2.09
NFGSM	53.61 ± 0.52	45.43 ± 0.49	40.70	24.44	20.76	18.53	1.00
ATAS	70.52 ± 15.66	21.50 ± 1.27	33.04	24.88	18.42	14.73	1.29
AAER	53.49 ± 0.49	45.38 ± 0.49	40.72	24.44	20.76	18.54	1.17
ELLE	52.85 ± 0.31	42.19 ± 0.04	37.75	25.11	20.76	16.22	2.63
TRADES-S	61.29 ± 7.68	8.91 ± 1.51	0.11	30.25	5.92	0.49	2.19
Ours	57.65 ± 0.65	$\mathbf{53.00 \pm 0.52}$	**45.08**	29.13	**26.56**	**22.41**	2.11

Table 5: **Multi-step methods, with and without our regularizer.** Adding our regularizer on top of a multi-step adversary improves over that adversary alone.

Method	CIFAR-10			CIFAR-100			Time (h)
	FGSM	PGD-10	AutoAttack	FGSM	PGD-10	AutoAttack	
PGD-10	56.21 ± 0.74	49.69 ± 0.36	46.09	27.79	25.45	20.79	5.99
PGD-10 + Ours	56.25 ± 0.35	$\mathbf{52.79 \pm 0.72}$	**46.57**	29.91	**27.90**	**23.65**	7.06
TRADES	57.05 ± 0.16	52.42 ± 0.09	48.38	30.36	28.01	23.33	6.87
TRADES + Ours	57.61 ± 0.08	$\mathbf{52.83 \pm 0.19}$	**48.51**	32.14	**30.25**	**24.13**	8.29

with amplification of $\sigma_1(H_{x\theta})$ and mixed consistency. This contrast shows that input-only consistency is insufficient in the single-step regime; without explicitly controlling joint input-parameter curvature, $\sigma_1(H_{x\theta})$ amplifies and ultimately triggers catastrophic overfitting.

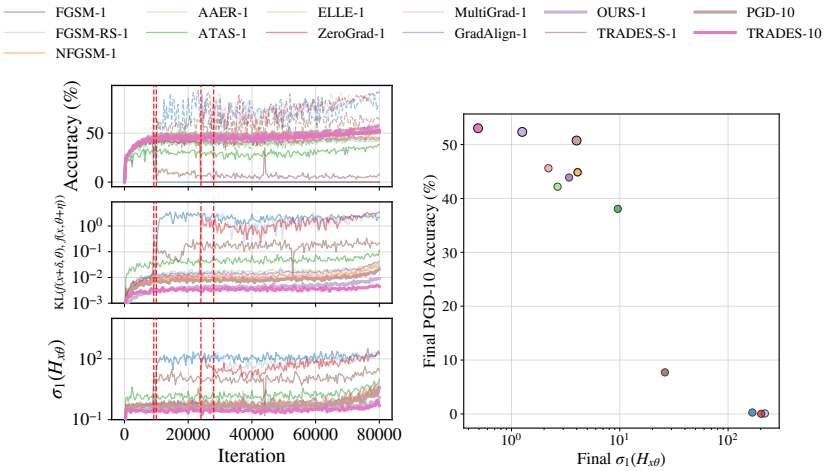
Notably, although TRADES does not explicitly regularize mixed curvature, fig. 4a shows that it effectively stabilizes mixed consistency and $\sigma_1(H_{x\theta})$, thereby preventing catastrophic overfitting. We hypothesize that multi-step adversaries implicitly suppress mixed curvature: each PGD step recomputes input gradients along the loss surface, exposing regions where input perturbations strongly affect parameter gradients, and thereby penalizing excessive mixed curvature. A single-step adversary relies on a local first-order linearization and lacks this implicit signal. Our KL divergence-based regularizer explicitly regularize this, achieving comparable stabilization in the single-step regime.

5.3 Experimental results

We now report our experimental results. Detailed setups, implementation details, additional experiments are in Appendix B, D and E.

Mixed Hessian control determines robustness. Table 4 and fig. 4b jointly confirm the picture from section 4. Single-step methods that fail to control mixed curvature—FGSM, FGSM-RS, ZeroGrad, and TRADES-S—all exhibit catastrophic overfitting, accompanied by rapid amplification of mixed consistency and $\sigma_1(H_{x\theta})$. In contrast, methods that successfully prevent catastrophic overfitting implicitly stabilize both quantities, even when not explicitly formulated to do so. The strength of this stabilization further links with final robustness under strong attacks: methods that more consistently suppress mixed curvature achieve higher AutoAttack accuracy (Croce & Hein, 2020).

Which curvature block a method controls determines how far its robustness reaches. Table 1 isolates the three regimes on a common setting. Controlling the input block alone—by gradient alignment, or by our own KL term with the parameter perturbation removed ($\eta_\theta = 0$), which reduces eq. (13) to the input-only form $\mathrm{KL}(p_{\eta_x} \,\|\, p_0)$ (see Appendix C)—or the mixed block alone (eq. (12)) plateaus in the mid-forties; controlling both jointly is what reaches the best robustness. This is the empirical counterpart of eq. (8): either factor suffices to *prevent* catastrophic overfitting, but suppressing only one leaves the other free to inflate the residual misalignment.



(a) Training Dynamics (b) Final $\sigma_1(H_{x\theta})$ and PGD Accuracy

Figure 4: **Mixed Hessian control determines robustness.** (a) Training dynamics of robust accuracy under PGD-10 (top), mixed consistency (middle), and $\sigma_1(H_{x\theta})$ (bottom). Catastrophic overfitting coincides with the amplification of mixed consistency and $\sigma_1(H_{x\theta})$ under FGSM, FGSM-RS, ZeroGrad, and even TRADES-S (see red dashed vertical lines). In contrast, methods that successfully prevent catastrophic overfitting consistently suppress both quantities. (b) Final $\sigma_1(H_{x\theta})$ at the end of training versus final PGD-10 robust accuracy across all methods. Methods with smaller final $\sigma_1(H_{x\theta})$ achieve higher PGD-10 robustness, indicating that stronger robustness is closely associated with better suppression of $\sigma_1(H_{x\theta})$. Together, these results show that controlling $\sigma_1(H_{x\theta})$ is critical for both preventing catastrophic overfitting and achieving strong adversarial robustness.

Our regularizer: best single-step robustness, with further gains in multi-step. Among single-step methods, our KL-based regularizer achieves the highest PGD-10 and AutoAttack robustness, surpassing PGD-10 in robust accuracy despite using single-step training. The benefit extends beyond the single-step regime: applying our regularizer on top of multi-step AT (**PGD-10 + Ours**, **TRADES + Ours**) yields *additional* robustness improvements over the strongest multi-step baselines (table 5). This indicates that mixed Hessian stabilization is not a single-step-specific remedy but a universal mechanism for robust training: explicitly controlling $\sigma_1(H_{x\theta})$ benefits any adversarial training procedure, regardless of whether the inner adversary is single-step or multi-step.

6 Conclusion

We have given a training-dynamics account of catastrophic overfitting. We showed that the collapse is produced by a misalignment of the parameter update, and that this misalignment is driven by spectral amplification of the mixed Hessian $H_{x\theta}$. This is a different question from the one prior work asks: it explains *when* the collapse occurs, through **input-space quantities**, while we ask *how* an input-space pathology reaches the **parameter updates** that produce it.

We supported this account at three levels. Analytically, the update misalignment factorizes into $\sigma_1(H_{x\theta})$ and the perturbation mismatch, and tracking the factors during training identifies the former as dominant. Interventionally, manipulating $\sigma_1(H_{x\theta})$ or the parameter-gradient norm directly on a checkpoint raises the other and raises the misalignment with it—so the dependence is causal and bidirectional. And constructively, penalizing $\sigma_1(H_{x\theta})$ alone is enough to prevent the collapse.

We then proposed a KL divergence-based regularizer whose second-order expansion is the joint Fisher information, suppressing the input, parameter, and mixed blocks through a single consistency term. It achieves the strongest robustness among single-step methods (see fig. 1 and table 4) and improves multi-step training (see table 5) when applied on top of it.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing. Specifically, we used a large language model to improve the clarity and concision of text drafted by the authors and to suggest alternative phrasings; all scientific content, claims, and conclusions originate from the authors. We have not used generative AI tools for research ideation, experimental design, or the derivation of any theoretical result, and retrieval or generation of citations is not applicable to this work. We have reviewed all AI-assisted work: every suggested edit was checked against the intended meaning by the authors, and no text was incorporated without verification. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 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.

Pau de Jorge Aranda, Adel Bibi, Riccardo Volpi, Amartya Sanyal, Philip Torr, Grégory Rogez, and Puneet Dokania. Make some noise: Reliable and efficient single-step adversarial training. *Advances in Neural Information Processing Systems*, 35:12881–12893, 2022.

Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban. Zerograd: Mitigating and explaining catastrophic overfitting in fgsm adversarial training. *arXiv preprint arXiv:2103.15476*, 2021.

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

Zhichao Huang, Yanbo Fan, Chen Liu, Weizhong Zhang, Yong Zhang, Mathieu Salzmann, Sabine Süsstrunk, and Jue Wang. Fast adversarial training with adaptive step size. *IEEE Transactions on Image Processing*, 32:6102–6114, 2023.

Stanisław Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of dnn loss and the sgd step length. In *International Conference on Learning Representations*, 2019.

Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8119–8127, 2021.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in neural information processing systems*, 33:2983–2994, 2020.

Sungyoon Lee, Jaewook Lee, and Saerom Park. Lipschitz-certifiable training with a tight outer bound. *Advances in Neural Information Processing Systems*, 33:16891–16902, 2020.

Sungyoon Lee, Woojin Lee, Jinseong Park, and Jaewook Lee. Towards better understanding of training certifiably robust models against adversarial examples. *Advances in Neural Information Processing Systems*, 34:953–964, 2021.

Binghui Li and Yuanzhi Li. Adversarial training can provably improve robustness: Theoretical analysis of feature learning process under structured data. In *The Thirteenth International Conference on Learning Representations*.

Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13409–13418, 2022.

Runqi Lin, Chaojian Yu, and Tongliang Liu. Eliminating catastrophic overfitting via abnormal adversarial examples regularization. *Advances in Neural Information Processing Systems*, 36: 67866–67885, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Elias Abad Rocamora, Fanghui Liu, Grigorios Chrysos, Pablo M Olmos, and Volkan Cevher. Efficient local linearity regularization to overcome catastrophic overfitting. In *The Twelfth International Conference on Learning Representations*, 2024.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

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

Florian Tramer, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *stat*, 1050:22, 2018.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Kaijie Zhu, Xixu Hu, Jindong Wang, Xing Xie, and Ge Yang. Improving generalization of adversarial training via robust critical fine-tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4424–4434, 2023.

GUIDE TO THE APPENDIX

All figures in the main paper—figs. 2 to 4—are measured on CIFAR-10 with ResNet-18 at $\varepsilon = 8/255$, the canonical single-step adversarial training benchmark on which the catastrophic-overfitting literature is built. The corresponding measurements on other backbones, which show the same rapid amplification of $\sigma_1(H_{x\theta})$ at the onset of the collapse and are therefore consistent with our main claim, are reported in section E.

The appendix is organized as follows. Section A proves theorem 1. Section B gives the experimental setup: the common protocol, the configuration of every baseline, the hyperparameters of our regularizer, and the protocols of the intervention and mixed-only experiments of sections 4.3 and 4.4. Section C ablates the regularizer—the input perturbation η_x, the parameter perturbation η_θ (including $\eta_\theta = 0$, which reduces the objective to an input-only penalty), the KL weight λ, and the three consistency views it controls. Section D reports additional evaluation on the main setting: the clean–robustness trade-off, stronger attacks, and extended training. Section E reports generalization across datasets and architectures, including the training dynamics on other backbones.

A PROOF OF THEOREM 1

Define

$$p_0(y) = p_\theta(y \mid x), \quad p_{\eta_x}(y) = p_\theta(y \mid x + \eta_x), \quad q_{\eta_\theta}(y) = p_{\theta+\eta_\theta}(x).$$

Consider the consistency regularizer

$$\mathrm{KL}(p_{\eta_x} \| q_{\eta_\theta}) = \sum_y p_{\eta_x}(y) \log \frac{p_{\eta_x}(y)}{q_{\eta_\theta}(y)}. \tag{19}$$

We analyze its second-order behavior around $(\eta_x, \eta_\theta) = (0, 0)$. We denote evaluation at $(\eta_x, \eta_\theta) = (0, 0)$ by $\big|_{0,0}$. Assume $p_{\eta_x}(y \mid x)$ is positive and twice continuously differentiable.

First-order terms vanish. At $(\eta_x, \eta_\theta) = (0, 0)$ we have KL$= 0$. Differentiating w.r.t. η_θ:

$$\nabla_{\eta_\theta} \mathrm{KL} = -\sum_y p_{\eta_x}(y) \nabla_{\eta_\theta} \log q_{\eta_\theta}(y). \tag{20}$$

At $(0, 0)$:

$$\nabla_{\eta_\theta} \mathrm{KL}\big|_{0,0} = -\sum_y p_0(y) \nabla_\theta \log p_\theta(y \mid x). \tag{21}$$

$$\sum_y p_0(y) \nabla_\theta \log p_0(y) = \sum_y p_0(y) \frac{\nabla_\theta p_0(y)}{p_0(y)} \tag{22}$$

$$= \nabla_\theta \sum_y p_0(y) \tag{23}$$

$$= 0. \tag{24}$$

Hence all first-order terms vanish.

Second-order derivatives. Differentiating again:

$$\nabla^2_{\eta_\theta \eta_\theta} \mathrm{KL}\big|_{0,0} = -\sum_y p_0(y) \nabla^2_{\theta\theta} \log p_\theta(y \mid x) \tag{25}$$

and similarly

$$\nabla^2_{\eta_x \eta_x} \mathrm{KL}\big|_{0,0} = -\sum_y p_0(y) \nabla^2_{xx} \log p_\theta(y \mid x), \tag{26}$$

$$\nabla^2_{\eta_x \eta_\theta} \mathrm{KL}\big|_{0,0} = -\sum_y p_0(y) \nabla^2_{x\theta} \log p_\theta(y \mid x). \tag{27}$$

Quadratic expansion. Since $\mathrm{KL}(p_{\eta_x} \| q_{\eta_\theta})\big|_{0,0} = 0$ and all first-order terms vanish, the second-order Taylor expansion around $(\eta_x, \eta_\theta) = (0, 0)$ yields

$$\mathrm{KL}(p_{\eta_x} \,\|\, q_{\eta_\theta}) = \frac{1}{2}\eta^\top F(x,\theta)\eta + o(\|\eta\|^2), \qquad \eta := [\eta_x; \eta_\theta]. \tag{28}$$

Using the second derivatives derived above, the Hessian with respect to (η_x, η_θ) at $(0, 0)$ can be written in block form as

$$\nabla^2_{(\eta_x,\eta_\theta)}\mathrm{KL}\big|_{0,0} = \begin{bmatrix} F_{xx} & F_{x\theta} \\ F_{\theta x} & F_{\theta\theta} \end{bmatrix}, \tag{29}$$

where

$$F_{xx} = -\mathbb{E}_{y\sim p_0}\left[\nabla^2_{xx} \log p_\theta(y \mid x)\right], \tag{30}$$

$$F_{\theta\theta} = -\mathbb{E}_{y\sim p_0}\left[\nabla^2_{\theta\theta} \log p_\theta(y \mid x)\right], \tag{31}$$

$$F_{x\theta} = -\mathbb{E}_{y\sim p_0}\left[\nabla^2_{x\theta} \log p_\theta(y \mid x)\right]. \tag{32}$$

Relation to Fisher information and loss Hessians. The block matrix in equation 29 is precisely the (joint) Fisher information of the predictive distribution with respect to (x, θ), since under standard regularity conditions,

$$-\mathbb{E}_{y\sim p_0}\left[\nabla^2 \log p_\theta(y \mid x)\right] = \mathbb{E}_{y\sim p_0}\left[\nabla \log p_\theta(y \mid x)\, \nabla \log p_\theta(y \mid x)^\top\right]. \tag{33}$$

Moreover, for cross-entropy loss $\mathcal{L}(x, y; \theta) = -\log p_\theta(y \mid x)$, we have the exact identity

$$\nabla^2_{x\theta}\mathcal{L}(x,y;\theta) = -\nabla^2_{x\theta}\log p_\theta(y \mid x), \quad \nabla^2_{xx}\mathcal{L}(x,y;\theta) = -\nabla^2_{xx}\log p_\theta(y \mid x), \tag{34}$$

(and similarly for $\nabla^2_{\theta\theta}$), so each Fisher block coincides with the *expected* loss-Hessian block:

$$F_{x\theta} = \mathbb{E}[\nabla^2_{x\theta}\mathcal{L}], \quad F_{xx} = \mathbb{E}[\nabla^2_{xx}\mathcal{L}], \quad F_{\theta\theta} = \mathbb{E}[\nabla^2_{\theta\theta}\mathcal{L}]. \tag{35}$$

Therefore minimizing

$$\mathrm{KL}(p_{\eta_x}(y) \,\|\, q_{\eta_\theta}(y)) \tag{36}$$

locally minimizes the Fisher / expected Hessian blocks $H_{xx}, H_{x\theta}, H_{\theta\theta}$ simultaneously, which suppresses joint curvature and stabilizes gradient dynamics, helping prevent catastrophic overfitting.

B Experimental Setup

B.1 Common protocol and baseline configurations

We conduct a set of experiments on CIFAR-10 and CIFAR-100 with ResNet-18 at section 5.3 (additional experiments are reported in Appendix E.1). During training, we set the maximum perturbation magnitude ϵ as 8/255 (see fig. 1 for experiments with varying ϵ). We evaluate train-time PGD-10/FGSM accuracy every 200 iterations. At the end of training, we evaluate robustness on the test set using FGSM, PGD-10, and AutoAttack, proposed by Croce & Hein (2020), to assess robustness under stronger attacks. We compare our method with several representative single-step and multi-step adversarial training methods. We use a SGD with cyclic learning rate scheduling, momentum of 0.9 and weight decay of $5e-4$. To examine whether catastrophic overfitting emerges at later stages of training, we train for 200 epochs. We train all methods on single A100 GPU.

FGSM AT Goodfellow et al. (2014) Fast Gradient Sign Method adversarial training uses a single-step perturbation

$$\delta = \epsilon \cdot \mathrm{sgn}(\nabla_x \mathcal{L}(x, y; \theta)), \tag{37}$$

and optimizes $\mathcal{L}(x + \delta, y; \theta)$.

FGSM-RS Wong et al. (2020) Fast adversarial training introduces a random start $\eta \sim \mathcal{U}(-\epsilon, \epsilon)^d$ before the FGSM update. We follow the recommended setting with step size $\alpha = 1.25\epsilon$ and project perturbation into ϵ ball.

GradAlign Andriushchenko & Flammarion (2020) GradAlign introduces a regularizer that maximizes the cosine similarity between input gradients at x and $x + \eta$, where $\eta \sim \mathcal{U}(-\epsilon, \epsilon)^d$:

$$\text{GradAlign}(x, y, \theta) = \cos(\nabla_x \mathcal{L}(x, y; \theta), \nabla_x \mathcal{L}(x + \eta, y; \theta)). \tag{38}$$

The objective is $\mathcal{L} + \lambda(1 - \text{GradAlign})$. We use $\lambda = 0.2$.

ZeroGrad and MultiGrad Golgooni et al. (2021) ZeroGrad suppresses small components of the input gradient when generating perturbations. We follow the recommendation setting with $q = 0.35$ for CIFAR-10 and $q = 0.45$ for CIFAR-100, which is threshold for zero out gradient, and step size $\alpha = 2\epsilon$. MultiGrad aggregates gradients from N randomly initialized perturbations and retains directions with consistent signs. We use $N = 3$ and $\alpha = 2\epsilon$.

N-FGSM de Jorge Aranda et al. (2022) N-FGSM increases the random initialization range and removes the final projection step. We follow the recommended setting with initialization $\eta \sim \mathcal{U}(-k, k)^d$ where $k = 2\epsilon$ and step size $\alpha = \epsilon$.

ATAS Huang et al. (2023) ATAS adapts the step size based on a moving average of squared gradient norms

$$v_i^j = \beta v_i^{j-1} + (1 - \beta)\|\nabla_x \mathcal{L}\|_2^2. \tag{39}$$

The step size is set to $\alpha_i^j = \frac{\gamma}{c + \sqrt{v_i^j}}$. We follow the recommended hyperparameters $\beta = 0.5$, $c = 0.01$, and $\gamma = 2c\epsilon$.

AAER Lin et al. (2023) AAER identifies abnormal adversarial examples (AAEs) as a primary cause of catastrophic overfitting. AAEs are defined as adversarial examples whose loss is lower than that of their corresponding clean samples:

$$\text{AAE}: \ \mathcal{L}(x + \eta, y) > \mathcal{L}(x + \eta + \delta, y), \quad \text{NAE}: \ \mathcal{L}(x + \eta, y) \leq \mathcal{L}(x + \eta + \delta, y). \tag{40}$$

To mitigate the influence of AAEs, AAER introduces a regularization term that penalizes abnormal loss variations and logit discrepancies between AAEs and normal adversarial examples. The AAER regularizer is defined as

$$\text{AAER} = \lambda_1 \frac{n}{m} \left(\lambda_2 \, \text{AAE-CE} + \lambda_3 \max(\text{AAE-L2} - \text{NAE-L2}, 0)\right), \tag{41}$$

where m is the batch size and n is the number of AAEs. We adopt the N-AAER variant and follow the authors' recommendations with $\lambda_1 = 1$, $\lambda_2 = 1.5$, and $\lambda_3 = 0.15$.

ELLE Rocamora et al. (2024) ELLE (Efficient Local Linearity Enforcement) encourages local linearity of the loss function by penalizing deviations from linear interpolation between adversarial samples. Two perturbed samples x_a and x_b are drawn as

$$x_a, x_b \sim x + \mathcal{U}(-\epsilon, \epsilon)^d, \quad \alpha \sim \mathcal{U}(0, 1), \tag{42}$$

and an interpolated point is formed as

$$x_c = (1 - \alpha)x_a + \alpha x_b. \tag{43}$$

The ELLE regularizer penalizes deviations from linear interpolation of losses:

$$E_{\text{lin}} = \left(\mathcal{L}(x_c, y) - (1 - \alpha)\mathcal{L}(x_a, y) - \alpha\mathcal{L}(x_b, y)\right)^2. \tag{44}$$

The overall objective augments the adversarial training loss with the ELLE penalty:

$$\min_\theta \ \mathcal{L}_{\text{adv}}(x, y; \theta) + \lambda \, E_{\text{lin}}(x, y; \theta), \tag{45}$$

so the parameter update includes $\nabla_\theta \mathcal{L}_{\text{adv}} + \lambda \nabla_\theta E_{\text{lin}}$. We follow the recommended setting $\lambda = 1000$.

PGD Madry et al. (2018) Projected Gradient Descent (PGD) adversarial training optimizes the worst-case loss within an ℓ_∞ perturbation set:

$$\min_\theta \left[\max_\delta f(x+\delta, y; \theta) \right]. \tag{46}$$

The inner maximization is approximated by K steps of PGD:

$$\delta_t^{(k+1)} = \Pi_{\|\delta\| \le \epsilon} \Big(\delta_t^{(k)} + \alpha \cdot \operatorname{sgn}\big(\nabla_x \mathcal{L}(x + \delta_t^{(k)}, y; \theta_t)\big) \Big), \tag{47}$$

In our experiments we use $K = 10$ attack steps with step size $\alpha = 2\epsilon/10$.

TRADES Zhang et al. (2019) TRADES improves the robustness-accuracy trade-off by decomposing the objective into a natural accuracy term and a robustness regularization term:

$$\min_\theta \mathbb{E}[\phi(p_0(y)) + \max_{\|\delta\|_\infty \le \epsilon} \phi(p_0(y), p_\delta(y))] \tag{48}$$

The adversarial perturbation is generated using PGD on the KL divergence term. We follow the standard setting with 10 attack steps and step size $\alpha = \epsilon/4$.

And for TRADES-S, we use one attack step and step size $\alpha = \epsilon$, which is consistent with single-step adversarial training.

B.2 Our regularizer: hyperparameters

Our method introduces a KL-based regularization term that enforces joint input-parameter consistency:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{adv}} + \lambda \, \mathrm{KL}\big(p_\delta(y) \,\|\, q_\eta(y)\big) \tag{49}$$

The KL regularization weight is fixed to $\lambda = 50$ in all experiments; section C.3 sweeps λ and shows why this value is the operating point we adopt.

B.3 Intervention protocols

Intervention experiments (section 4.3). Interventions start from checkpoints saved along an FGSM training trajectory on CIFAR-10 with ResNet-18 at $\varepsilon = 8/255$. At each intervention step we optimize the parameters toward a target quantity—either $\sigma_1(H_{x\theta})$ or $\|\nabla_\theta \mathcal{L}(x+\delta_t)\|$—while holding the training loss as an anchor, so that the objective is

$$\min_\theta \; \mathcal{L}(x+\delta_t, y; \theta) \; - \; \lambda_{\text{int}} \, T(\theta), \qquad T \in \{\sigma_1(H_{x\theta}), \; \|\nabla_\theta \mathcal{L}(x+\delta_t)\|\}, \tag{50}$$

with λ_{int} chosen so that the two gradient contributions are of comparable norm at the first step. We use $K = 2000$ optimization steps.

The control arm is identical in step count, step size, normalization and batch stream, and differs only in that the target term is removed ($\lambda_{\text{int}} = 0$), so that the parameters are optimized on the training loss alone over the same budget. The suppression arm of table 2 reverses the sign of λ_{int} and is applied to a checkpoint taken after catastrophic overfitting has already occurred.

C Ablations of the Regularizer

C.1 Input Perturbation η_x

Two variants of η_x are considered:

Random η_x. A random perturbation is sampled uniformly within the ℓ_∞ ball:

$$\eta_x \sim \mathcal{U}(-\epsilon', \epsilon')^d, \quad \|\eta_x\|_\infty \le \epsilon', \tag{51}$$

where $\epsilon' = 2/255$.

Table 6: Effect of input perturbation choice in the KL regularizer on CIFAR-10 with ResNet18.

Method	Clean	FGSM	PGD-10
Ours (Random η_x)	79.69	53.57	45.42
Ours (FGSM δ/4)	81.14	57.03	52.46

Table 7: **Ablation over the parameter-perturbation scale σ.** CIFAR-10, ResNet-18. The $\sigma = 0$ row removes the perturbation entirely, reducing the regularizer to an input-only consistency penalty. Every nonzero σ exceeds $\sigma = 0$ by 4.0–6.9 points of PGD-10.

σ	Clean	FGSM	PGD-10
0 (input-only)	82.98	53.34	47.10
10^{-10}	81.14	57.03	52.46
10^{-8}	81.03	56.58	51.56
10^{-6} (default)	81.14	**58.48**	**53.79**
10^{-4}	81.03	57.81	53.01
10^{-2}	81.81	56.58	51.34

FGSM-based η_x. In the main experiments we instead use a scaled FGSM perturbation direction:

$$\eta_x = \frac{1}{4}\epsilon \cdot \operatorname{sgn}(\nabla_x \mathcal{L}(x, y; \theta)). \tag{52}$$

This choice provides a more informative local perturbation direction while maintaining a small perturbation magnitude, and empirically yields stronger robustness compared to random perturbations (see table 6).

C.2 Parameter perturbation η_θ.

Parameter perturbations are sampled from an isotropic Gaussian distribution:

$$\eta_\theta \sim \mathcal{N}(0, \sigma^2 I_d), \tag{53}$$

where d is the parameter dimension. We set $\sigma = 10^{-6}$ throughout.

The decisive row of this ablation is $\sigma = 0$, which removes the parameter perturbation entirely. The regularizer then reduces *exactly* to an input-space consistency penalty, $\mathrm{KL}\big(p_{\eta_x}(y) \,\|\, p_\theta(y \mid x)\big)$.

The $\sigma = 0$ row is also a check on the mechanism rather than only on the hyperparameter. With the parameter branch removed the objective becomes an input-space method, and it behaves like one: it prevents catastrophic overfitting—as input-space defenses do—but plateaus in the same band as GradAlign and ELLE in table 4, well below any setting in which the mixed block is weighted. This is consistent with eq. (8): controlling either factor averts the collapse, but controlling only the input factor leaves the residual misalignment unbounded by the other.

C.3 Effect of KL Regularization Weight λ

The KL regularization weight λ in eq. (17) controls the strength of joint curvature suppression. We fix $\lambda = 50$ in all main experiments. To validate this choice, we vary λ over $\{10, 20, 50, 100, 200\}$ and report robustness on CIFAR-10 with ResNet-18.

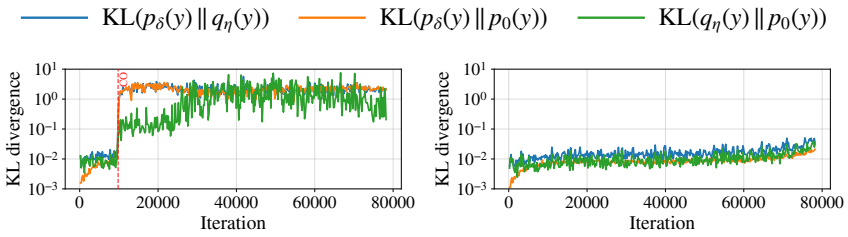
As expected from the joint curvature interpretation (theorem 1), excessively small λ fails to suppress mixed Hessian growth and leads to degraded PGD robustness, while overly large λ over-regularizes and reduces clean accuracy. $\lambda = 50$ provides a balanced trade-off across all evaluation metrics, and we adopt this value throughout the main experiments.

C.4 Three Consistency Views During Training

Building on theorem 1, the KL divergence formulation in eq. (13) admits a natural decomposition into three consistency views. Specifically, the input consistency $\mathrm{KL}(p_{\eta_x}(y) \,\|\, p_0(y))$ has second-

Table 8: Effect of the KL regularization weight λ on CIFAR-10 with ResNet-18, $\varepsilon = 8/255$. Rows are mean $\pm$ std over three seeds. Every setting from $\lambda = 10$ to 200 exceeds all competing single-step methods on PGD-10, so the method does not depend on a tuned weight; larger λ trades clean accuracy for robustness, and we adopt $\lambda = 50$ as the knee of that curve.

λ	Clean	FGSM	PGD-10
10	82.09 ± 0.33	55.57 ± 0.98	48.70 ± 0.42
20	83.02 ± 0.12	56.35 ± 0.30	49.47 ± 0.35
50 (default)	81.55 ± 0.36	57.65 ± 0.65	53.00 ± 0.52
100	78.88 ± 0.08	56.46 ± 1.15	51.44 ± 2.98
200	75.24 ± 0.37	56.38 ± 1.05	$\mathbf{54.19 \pm 0.39}$



(a) FGSM-training (b) PGD-training

Figure 5: **KL divergence dynamics during FGSM/PGD-training.** The input, parameter, and joint KL divergence terms—corresponding to input curvature, loss-landscape sharpness, and joint input-parameter curvature—grow rapidly near catastrophic overfitting under FGSM-training, while remaining comparatively stable under PGD-training.

order form $\frac{1}{2}\eta_x^\top F_{xx}\eta_x$ and therefore penalizes input curvature, while the parameter consistency $\mathrm{KL}(q_{\eta_\theta}(y) \,\|\, p_0(y))$ yields $\frac{1}{2}\eta_\theta^\top F_{\theta\theta}\eta_\theta$ and corresponds to loss-landscape sharpness control. Finally, the mixed consistency $\mathrm{KL}(p_{\eta_x}(y) \,\|\, q_{\eta_\theta}(y))$ produces the full quadratic form containing the mixed block $F_{x\theta}$, which governs joint input-parameter curvature.

We next examine how these KL divergence consistency views evolve during FGSM/PGD-training. Empirically, we observe that:

- input consistency increases as local input curvature grows
- parameter consistency increases with progressive loss sharpening
- mixed consistency amplifies sharply near catastrophic overfitting

Figure 5 shows the evolution of the three KL divergence consistency terms during training. All KL divergence measures increase during FGSM/PGD-training, which is consistent with progressive loss sharpening (Cohen et al., 2021; Jastrzebski et al., 2019) and input curvature growth (Andriushchenko & Flammarion, 2020). However, under FGSM-training, these quantities exhibit a rapid amplification near catastrophic overfitting, whereas PGD-training shows a much more stable increase. This behavior aligns with our previous analysis of mixed Hessian dynamics and suggests that controlling KL divergence-based consistency can help prevent curvature amplification.

D Additional Evaluation

D.1 Clean Accuracy and the Robustness Trade-off

The main tables report robustness only, because that is the axis our analysis targets: section 4 concerns the update misalignment that destroys robustness, and clean accuracy is not the quantity the mixed Hessian governs. Clean accuracy is nonetheless a real cost of any adversarial training method, so we characterize it here.

Table 9: **The clean–robustness trade-off is a knob, not a fixed cost.** CIFAR-10, ResNet-18, $\varepsilon = 8/255$, mean $\pm$ std over three seeds, sorted by clean accuracy and interleaved with the single-step baselines that do not catastrophically overfit. Our configurations are marked in bold. At $w = 0.5$ we exceed NFGSM, AAER and MultiGrad on *both* axes simultaneously, and at every clean-accuracy level in the table our configuration at that level is the most robust.

Method	Clean	FGSM	PGD-10
ELLE	85.91 ± 0.18	52.85 ± 0.31	42.19 ± 0.04
GradAlign	85.60 ± 0.17	54.34 ± 0.88	43.44 ± 0.86
Ours ($w = 1.0$)	84.96 ± 0.81	55.84 ± 0.00	$\mathbf{48.31 \pm 0.73}$
Ours ($w = 0.5$)	84.31 ± 1.06	57.10 ± 0.35	$\mathbf{50.56 \pm 0.57}$
MultiGrad	84.02 ± 0.35	54.53 ± 0.30	44.95 ± 0.28
AAER	82.60 ± 0.23	53.49 ± 0.49	45.38 ± 0.49
NFGSM	82.40 ± 0.13	53.61 ± 0.52	45.43 ± 0.49
Ours ($w = 0$, default)	81.55 ± 0.36	57.65 ± 0.65	$\mathbf{53.00 \pm 0.52}$

Our objective exposes the trade-off through two hyperparameters that are already part of it. The KL weight λ (eq. (17)) moves toward robustness. In the opposite direction, adding a clean cross-entropy term

$$\mathcal{L}^{(w)}_{\text{total}} = \mathcal{L}_{\text{adv}} + \lambda \, \mathrm{KL}\big(p_{\eta_x} \| q_{\eta_\theta}\big) + w \cdot \mathcal{L}(x, y; \theta) \tag{54}$$

moves toward clean accuracy.

Read down the table 9, the trade-off does not hold against prior work. At $w = 0.5$ we reach 84.31% clean accuracy with 50.56% PGD-10, which *dominates* MultiGrad, AAER and NFGSM on both axes at once—higher clean accuracy *and* $+5.1$ to $+5.6$ points of robustness. Against the two highest-clean baselines, ELLE and GradAlign, we give up 0.6–1.6 points of clean accuracy for $+4.9$ to $+8.4$ points of PGD-10. The cost is therefore incurred relative to our own default, which sits deliberately at the robustness end of the curve, and not relative to prior work. The exchange is also smooth and monotone in w, so a practitioner with a clean-accuracy requirement can meet it by choosing w rather than by changing method.

D.2 Evaluation under Stronger Attacks

The main tables report PGD-10 for comparability with prior work. To check that our robustness is not an artifact of a weak evaluation, we additionally evaluate with PGD-100 using 10 random restarts (see table 10).

Table 10: **Robustness under a stronger attack.** PGD-100 with 10 random restarts. CIFAR-10, ResNet-18, $\varepsilon = 8/255$. Multi-step references are marked †.

Method	PGD-100 $\times$ 10 restarts
FGSM	0.0
FGSM-RS	0.0
ZeroGrad	0.0
TRADES-S	1.2
ATAS	33.9
ELLE	39.3
GradAlign	40.6
MultiGrad	42.4
NFGSM	42.7
AAER	42.7
Ours	**50.5**
PGD-10†	48.6
TRADES†	51.2

D.3 Robustness under Extended Training

A natural concern for any catastrophic-overfitting prevention method is whether the stabilization persists over substantially longer training. While the main experiments in section 5.3 adopt the standard 200-epoch protocol, we additionally train our method for 1000 epochs on CIFAR-10 with ResNet-18, keeping all other hyperparameters identical.

Table 11: Final robust accuracy of our method under extended training on CIFAR-10 with ResNet-18.

Epochs	Clean	FGSM	PGD-10
200 (default)	81.14	57.03	52.46
1000	82.70	56.03	49.22

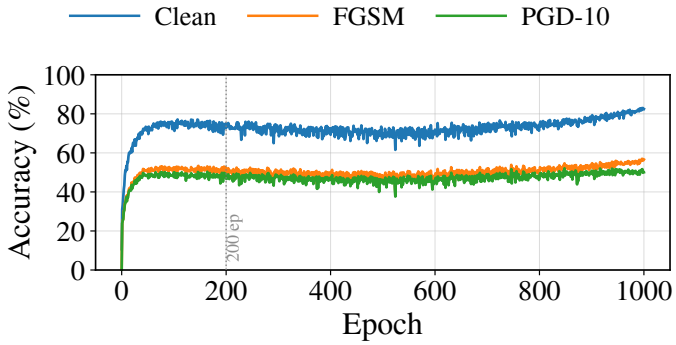


Figure 6: **Robust accuracy of our method throughout 1000-epoch training on CIFAR-10 with ResNet-18.** Clean, FGSM, and PGD-10 accuracy are evaluated periodically during training. The vertical dotted line marks the standard 200-epoch boundary (section 5.3). Our method exhibits no robustness collapse across the entire extended training horizon, in contrast to single-step baselines such as FGSM, FGSM-RS, ZeroGrad, and TRADES-S, which already exhibit catastrophic overfitting within the first 200 epochs (fig. 4).

Table 11 and fig. 6 together show that our method maintains robust accuracy throughout extended training, with no signs of catastrophic overfitting. Figure 6 visualizes the full training trajectory: PGD-10 accuracy stays stable well past the 200-epoch boundary and through 1000 epochs, while table 11 reports the final accuracies at both horizons. This stability is consistent with our analysis in section 4: by directly suppressing mixed Hessian spectral growth via joint KL consistency, our regularizer prevents the positive feedback loop that drives catastrophic overfitting, regardless of training duration. We note that single-step methods which fail to control mixed consistency (e.g., FGSM, FGSM-RS, ZeroGrad, TRADES-S) already exhibit catastrophic overfitting well within 200 epochs (fig. 4), so the relevant question at 1000 epochs is whether our method remains stable—which it does.

E Generalization Across Datasets and Architectures

Which configurations produce catastrophic overfitting is well documented for the canonical setting—CIFAR-10 and CIFAR-100 with ResNet-18 and WideResNet—on which the catastrophic-overfitting literature is built, and it is there that our analysis in section 4 is carried out. For other architectures no such recipes exist, and they are not easy to find: on several of the backbones below we swept learning rate, batch size and schedule without producing a collapse.

This splits the generalization question in two, and we treat both. Where catastrophic overfitting *does not* occur, the question is whether our regularizer still applies and still helps; the answer should be that training remains stable and robustness improves, since nothing is there to be prevented. Where catastrophic overfitting *does* occur, the question is the one the paper is about: single-step training

should break, and our regularizer should keep it stable while retaining strong robustness. We find an instance of each—ConvNeXt for the first, Swin-Base for the second—and report them in section E.3.

E.1 Other datasets and model scales

In the main paper, we report results on CIFAR-10 and CIFAR-100 with ResNet-18. To further evaluate the generality of our method, we additionally conduct experiments on Tiny-ImageNet with ResNet-18 (see table 12) and with a larger architecture, WideResNet-28 on CIFAR-10 and CIFAR-100 (see table 13). Tables 12 and 13 confirm that our approach consistently stabilizes training dynamics and maintains strong robustness across different datasets and model scales.

Table 12: Clean, FGSM, PGD, AutoAttack accuracy (%) on Tiny ImageNet with ResNet-18 across various methods.

Method	TinyImageNet			
	Clean	FGSM	PGD	AA
FGSM	47.32	19.20	16.07	13.37
FGSM-RS	47.66	17.86	14.96	13.00
GradAlign	48.66	80.25	0.0	0.0
ZeroGrad	47.21	18.08	14.29	13.66
MultiGrad	47.77	19.31	15.18	14.16
NFGSM	46.76	20.65	16.96	14.56
ATAS	56.81	16.85	10.94	9.32
AAER	46.76	20.65	16.96	14.60
ELLE	50.45	19.42	15.85	12.51
TRADES-S	58.15	13.62	0.45	0.00
(Ours)	41.63	21.21	**19.08**	**16.49**
PGD-10	45.98	19.87	17.52	16.02
TRADES	46.43	22.54	**19.53**	**16.03**

E.2 Vision Transformer Architecture

To assess whether our method generalizes beyond convolutional architectures, we additionally evaluate it on a Vision Transformer. We use ViT-Small with patch size 4 on CIFAR-10, trained from scratch. All training settings—optimizer, learning rate schedule, weight decay, batch size, and total epochs—are kept identical to our ResNet-18 main experiments, isolating the effect of architecture. We compare with FGSM (the canonical single-step baseline) and PGD-10 (the multi-step reference).

ViT-Small falls on the first side of the split above: single-step training does not collapse here—FGSM reaches 34.38% PGD-10—so there is no catastrophic overfitting for any method to prevent. On this backbone our regularizer trains stably and keeps PGD-10 accuracy at the same level as the multi-step reference, but it does not improve over FGSM, trailing it by 1.8 points.

We attribute this to hyperparameters rather than to the mechanism, and report it as a limitation. The settings here are transferred unchanged from ResNet-18 in order to isolate the effect of architecture, which is the right control for asking whether the method *applies*, but it leaves λ and the perturbation scales untuned for a transformer trained from scratch—a regime in which all three methods reach only the low thirties, well below their CIFAR-10 accuracy on convolutional backbones. We expect the tuning to close the gap here, and leave a systematic study of transformer-specific settings to future work.

E.3 Larger Models and Other Architectures

We train FGSM, PGD-10 and our regularizer on ConvNeXt-Tiny (27.8M), ConvNeXt-Base (87.6M) and Swin-Base (86.7M) on CIFAR-10 at $\varepsilon = 8/255$. Between them these backbones cover both sides of the split described above.

Where catastrophic overfitting does not occur: our method is stable and improves robustness. On ConvNeXt-Tiny and ConvNeXt-Base we swept learning rate, batch size and schedule

Table 13: Clean, FGSM, PGD, AutoAttack accuracy (%) on CIFAR-10, CIFAR-100, Tiny ImageNet with WideResNet-28 across various methods

Method	CIFAR-10				CIFAR-100				TinyImageNet			
	Clean	FGSM	PGD	AA	Clean	FGSM	PGD	AA	Clean	FGSM	PGD	AA
FGSM	88.73	84.71	0.00	0.00	64.17	61.27	0.11	0.00	42.63	15.96	12.17	9.30
FGSM-RS	84.93	95.31	0.00	0.00	60.83	75.89	0.00	0.00	16.74	10.16	0.00	0.00
GradAlign	89.06	92.52	0.11	0.00	68.08	71.09	0.45	0.00	43.19	46.32	0.11	0.00
ZeroGrad	83.93	96.21	0.00	0.00	63.17	81.47	0.22	0.00	54.02	31.36	0.00	0.00
MultiGrad	86.50	55.25	43.75	39.87	58.82	26.67	21.76	19.33	51.67	22.21	16.41	13.86
NFGSM	84.82	56.70	46.54	41.90	55.36	27.34	23.10	20.10	50.56	20.87	16.85	15.22
ATAS	91.52	54.46	41.74	37.14	66.74	28.35	21.32	17.44	58.82	18.08	10.83	10.15
AAER	84.82	56.70	46.54	41.88	55.36	27.34	23.10	20.10	50.56	20.87	16.85	**15.23**
ELLE	71.21	42.30	37.83	31.78	50.56	23.21	21.76	17.58	38.84	14.96	13.50	11.36
TRADES-S	92.97	58.71	5.80	0.06	73.88	29.91	3.13	0.25	60.04	17.97	0.67	0.00
(Ours)	79.24	55.25	**51.34**	**44.69**	50.45	31.14	**28.57**	**23.71**	36.38	17.97	**17.19**	14.27
PGD-10	85.94	58.15	48.88	46.24	56.58	29.91	25.45	21.84	50.78	22.43	**17.52**	**15.78**
TRADES	84.49	58.71	**53.68**	**50.88**	56.03	33.26	**30.13**	**25.81**	48.10	24.55	16.74	11.21

Table 14: Clean, FGSM, PGD-10 robust accuracy (%) on CIFAR-10 with ViT-Small. All methods share the same training protocol as our ResNet-18 experiments.

Method	Clean	FGSM	PGD-10
FGSM	52.79	35.38	34.38
Ours	46.88	32.59	32.59
PGD-10	53.12	35.71	34.71

and found *no* configuration in which FGSM training collapses: single-step training reaches 31.30% and 34.26% PGD-10 accuracy, far from the 0% that defines the phenomenon. We report this directly rather than presenting these backbones as evidence of prevention. What they test instead is whether the regularizer applies and helps where there is nothing to prevent, and it does on both: training is stable throughout, and PGD-10 accuracy improves over the single-step baseline it replaces by $+4.30$ points on ConvNeXt-Tiny and $+2.12$ on ConvNeXt-Base, matching or exceeding multi-step PGD-10 training (35.60 vs. 32.50 and 36.38 vs. 36.61) at single-step cost.

Where catastrophic overfitting does occur: single-step training breaks and our method does not. On Swin-Base we did find a configuration that produces the collapse, and it carries the textbook signature: FGSM training reports 98.50% FGSM accuracy alongside 0.00% PGD-10 accuracy, with clean accuracy simultaneously falling to 27.97%—the model has overfitted the single-step adversary so completely that it is no longer a classifier. Under our regularizer the collapse does not occur: training stays stable and PGD-10 accuracy reaches 39.08%, level with multi-step PGD-10 training (39.35%) at $2.7\times$ its training speed (182 vs. 500 seconds per epoch).

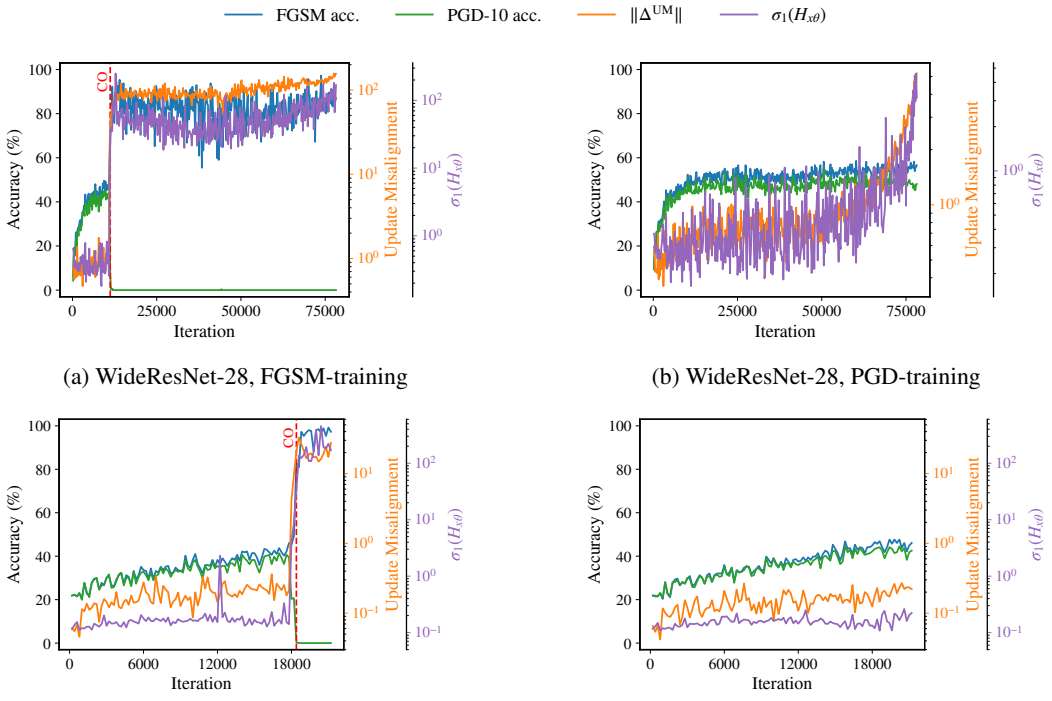
The collapse there is the same phenomenon. The dynamics in fig. 7 show that Swin-Base fails in the way our analysis predicts. As on ResNet-18, $\sigma_1(H_{x\theta})$ amplifies sharply at the transition, the update misalignment $\|\Delta^{\mathrm{UM}}\|$ rises with it, and the FGSM and PGD accuracies separate at that same iteration; under our regularizer $\sigma_1(H_{x\theta})$ instead remains controlled throughout. The mechanism is therefore not an artifact of convolutional architectures at ResNet scale—an 86.7M-parameter transformer collapses through the same mixed-curvature amplification, and suppressing that curvature is what keeps it stable. Taken with the ConvNeXt results, this verifies both halves of our claim outside the setting in which they were developed.

E.4 Training Dynamics on Other Architectures

The dynamics figures in section 4 use ResNet-18. To confirm that the mechanism is not specific to that backbone, we repeat the measurement on WideResNet-28 and Swin-Base, tracking $\sigma_1(H_{x\theta})$, the update misalignment $\|\Delta^{\mathrm{UM}}\|$, and FGSM/PGD-10 accuracy on a fixed probe batch at regular intervals throughout training. We train FGSM and PGD only: the figure concerns the phenomenon, not the method.

Table 15: **Larger models and other architectures.** CIFAR-10, $\varepsilon = 8/255$. Best PGD-10 within each backbone in bold. FGSM accuracy is not indicative of true robustness: on Swin-Base, FGSM training reports 98.50% FGSM accuracy alongside 0.00% PGD-10, the signature of catastrophic overfitting.

Backbone	Training	Clean	FGSM	PGD-10
ConvNeXt-Base (87.6M)	FGSM (single-step)	73.33	38.84	34.26
	PGD-10 (multi-step)	74.78	39.73	**36.61**
	Ours (single-step)	71.88	42.75	36.38
ConvNeXt-Tiny (27.8M)	FGSM (single-step)	72.80	36.80	31.30
	PGD-10 (multi-step)	75.70	36.70	32.50
	Ours (single-step)	70.50	42.90	**35.60**
Swin-Base (86.7M)	FGSM (single-step)	27.97	98.50	0.00
	PGD-10 (multi-step)	65.45	41.30	**39.35**
	Ours (single-step)	57.18	40.03	39.08



(a) WideResNet-28, FGSM-training

(b) WideResNet-28, PGD-training

(c) Swin-Base, FGSM-training

(d) Swin-Base, PGD-training

Figure 7: **The same transition appears on other backbones.** FGSM/PGD-10 accuracy (blue/green), update misalignment (orange) and $\sigma_1(H_{x\theta})$ (purple) during FGSM and PGD training on WideResNet-28 (top) and Swin-Base (bottom), CIFAR-10, $\epsilon = 8/255$. Under FGSM-training, $\sigma_1(H_{x\theta})$ and the update misalignment turn upward at the iteration at which the two accuracies separate (red dashed line, denoted CO in the figure); under PGD-training both remain controlled and the accuracies stay together.