CLEAN FEATURE ANCHORING: MITIGATING THE ACCURACY–ROBUSTNESS TRADE-OFF BY SELF-DISTILLING A NATURALLY TRAINED MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training improves robustness but often sacrifices accuracy on clean data, leading to a persistent clean–robust trade-off. We address this trade-off by first learning a high-accuracy clean self-teacher from natural images and then preserving its clean knowledge while adversarially training the student. We propose Clean Feature Anchoring (CFA), which matches the student's representation of an adversarial input to the teacher's fixed representation of the corresponding clean input while retaining the teacher's classifier. We show that controlling this fixed clean anchor simultaneously preserves agreement with the teacher on clean inputs and limits representation variation within the perturbation neighborhood. We also find that teacher clean accuracy alone does not determine how well clean performance is retained after robust training, indicating that transferability depends on properties beyond teacher accuracy. Finally, because the same input-space perturbation can induce substantially different changes in the anchoring objective across examples, we introduce sensitivity-matched perturbation budgets that assign smaller radii to more sensitive examples and larger radii to less sensitive ones while preserving the average perturbation budget. Across CIFAR-10, CIFAR-100, and Tiny-ImageNet, CFA consistently combines high clean accuracy with strong adversarial robustness.

1 INTRODUCTION

Adversarial training is one of the most effective approaches to improving robustness against adversarial perturbations, but this robustness often comes at the cost of accuracy on clean data (Madry et al., 2017; Tsipras et al., 2018; Zhang et al., 2019). This persistent clean–robust trade-off raises a natural question: can a model acquire adversarial robustness while retaining more of the knowledge that supports strong clean performance? Several approaches seek to mitigate this trade-off by leveraging naturally trained models, whose accurate predictions and representations on clean inputs provide useful references during adversarial training. However, as the student adapts to adversarial perturbations, the clean knowledge inherited from such models can still be altered, leaving open how that knowledge can be preserved throughout robust training. We approach this problem by separating the learning of clean knowledge from the subsequent learning of robustness. A clean self-teacher is first trained on natural images and then frozen, providing a stable reference while the student is adversarially trained. We instantiate this perspective with **Clean Feature Anchoring (CFA)**. The student is initialized from the self-teacher, and during adversarial training CFA matches the student's representation of an adversarial input to the teacher's fixed representation of the corresponding clean input while retaining the inherited classifier. The teacher therefore need not provide robustness or targets on adversarial inputs: it specifies what clean knowledge should be preserved, while adversarial training determines the neighborhood over which the student learns to preserve it.

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.

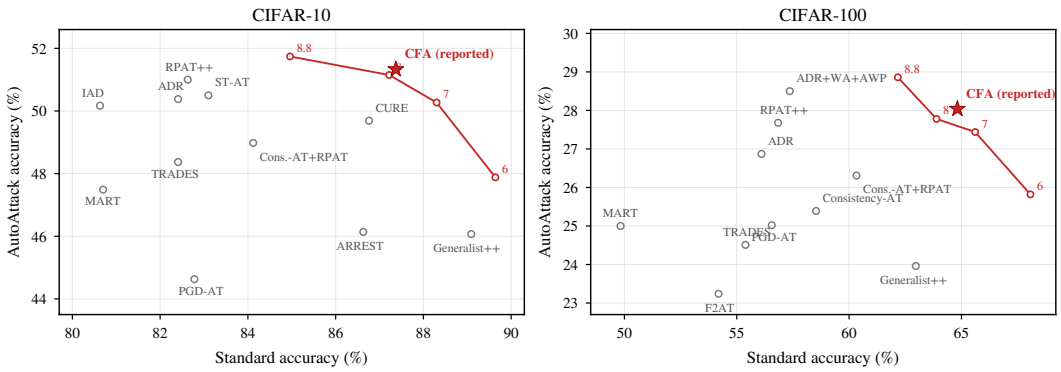

Figure 1: Clean–AutoAttack trade-off across CFA training radii on CIFAR-10 and CIFAR-100 (ResNet-18; evaluation at $8/255$). Labels show $255\epsilon_{\text{train}}$ for 100-epoch runs; stars mark the reported 150-epoch results. Grey points are published results.

This fixed clean anchor provides more than a target for adversarial representation matching. We show that controlling the worst-case discrepancy from the anchor simultaneously keeps the student's clean representation close to the teacher and limits its variation within the adversarial neighborhood. Importantly, this does not require the teacher itself to be robust. We further study what makes a useful clean reference and find that teacher clean accuracy alone is not sufficient to predict transfer: teachers with similar clean accuracy can yield different student outcomes, and increasing teacher accuracy does not consistently improve student clean accuracy. Thus, the quality of a clean teacher for robust transfer is determined by more than its clean accuracy alone. The perturbation neighborhood itself introduces a second challenge. The same input-space radius can induce substantially different changes in the anchoring objective across training samples. We therefore introduce sensitivity-matched perturbation budgets, assigning smaller perturbation radii to more sensitive samples and larger radii to less sensitive ones while preserving the average perturbation budget. This complements the fixed clean target with example-dependent adversarial neighborhoods, matching the strength of the perturbation to the local sensitivity of the objective being optimized. Across CIFAR-10, CIFAR-100, and Tiny-ImageNet, CFA consistently combines high clean accuracy with strong adversarial robustness and achieves strong clean–robust trade-offs among the evaluated methods. Our contributions are summarized as follows:

- We introduce a clean self-teacher framework that separates the learning of clean knowledge from the subsequent learning of robustness. CFA instantiates this principle by anchoring adversarial student representations to a fixed clean teacher representation.
- We characterize what the fixed clean anchor controls, showing that it jointly constrains clean fidelity to the teacher and representation variation within the adversarial neighborhood. We further show empirically that teacher clean accuracy alone does not determine transferability to the robust student.
- We introduce sensitivity-matched perturbation budgets, which redistribute a fixed average perturbation budget according to the local sensitivity of the anchoring objective, providing example-dependent adversarial neighborhoods around a fixed clean target.

2 Related Work

Adversarial training and its targets. PGD adversarial training optimizes a model on worst-case inputs (Madry et al., 2017); TRADES separates clean prediction from local consistency (Zhang et al., 2019), and MART reweights misclassified examples (Wang et al., 2020). Subsequent work improves generalization through label smoothing, consistency regularization, weight averaging, and adversarial weight perturbation (Pang et al., 2020; Tack et al., 2022; Rice et al., 2020; Wu et al., 2020). ADR replaces one-hot labels with targets produced by an annealed momentum teacher (Wu et al., 2024). These approaches motivate our study of how the supervisory target affects the clean–robust trade-off.

Predictive targets in adversarial distillation. ARD transfers a robust teacher's predictions during adversarial training (Goldblum et al., 2020). RSLAD emphasizes robust soft labels (Zi et al., 2021), IAD discounts unreliable teacher guidance (Zhu et al., 2021), and AdaAD uses the teacher in both adversarial-example generation and student optimization (Huang et al., 2023). IGDM adds the teacher's local input geometry to logit-based distillation (Lee et al., 2025). These methods are designed to inherit robustness from an adversarially trained teacher. Our target instead comes from a natural teacher at the clean input and remains fixed during the student's inner maximization.

Guidance from naturally trained models. BGAT is the closest direct-regression precedent: it matches adversarial logits to a fixed natural teacher's clean logits, and LBGAT extends this to a jointly trained natural branch (Cui et al., 2021). ARREST starts from a naturally pretrained model and regularizes latent representations while retaining adversarial cross-entropy and noisy replay (Suzuki et al., 2023). HAT uses a natural helper to label examples outside the training ball (Rade & Moosavi-Dezfooli, 2022), and B-MTARD combines clean and robust teachers with adaptive temperature and loss balancing (Zhao et al., 2024). CFA similarly uses a naturally trained model as a clean reference, but treats the self-teacher and its transferability as part of the design: our analysis examines what the fixed clean anchor controls, how teacher properties affect transfer, and how its perturbation neighborhood is allocated. In particular, teacher clean accuracy alone is insufficient to characterize transferability (Section 3.3).

Sample-dependent perturbation budgets. IAAT and CAT adapt the training radius to sample difficulty, while MMA directly increases an input-space margin (Balaji et al., 2019; Cheng et al., 2022; Ding et al., 2020). Our allocation has a different source: it redistributes a fixed batch budget according to the input sensitivity of the loss being optimized. The comparison is therefore not between adaptive and uniform radii alone, but between the signals used to assign the same budget (Table 6).

3 Analysis

Adversarial training often improves robustness at the cost of clean accuracy. To mitigate this trade-off, we first learn a clean self-teacher from natural images and then use it as a fixed reference while adversarially training the student. In this section, we motivate this fixed clean target from the perspective of knowledge distillation, characterize what controlling the target implies for the student, and finally examine which properties make a clean self-teacher a useful reference.

3.1 From Knowledge Distillation to a Fixed Clean Target

With a natural teacher, predictive distillation provides a natural starting point for transferring clean knowledge to an adversarially trained student. We therefore first ask how such supervision behaves when the teacher is kept fixed and no separate label objective is introduced. Let $f = h \circ \Phi$, where Φ is a feature map and $h(z) = Wz + b$ is a linear classifier producing logits $f(x) \in \mathbb{R}^C$ for C classes. Subscripts t and s denote the frozen clean self-teacher and student, respectively. For a clean input x and a perturbed input $x' \in B(x, \epsilon)$, standard knowledge distillation with a shared temperature τ minimizes $\tau^2 D_{\mathrm{KL}}(q_t^\tau(x) \| p_s^\tau(x'))$, where $q_t^\tau(x) = \mathrm{softmax}(f_t(x)/\tau)$ and $p_s^\tau(x') = \mathrm{softmax}(f_s(x')/\tau)$. As the temperature increases, this objective approaches direct matching to a fixed clean teacher target. In particular, with $\bar{f} = f - \frac{1}{C}\mathbf{1}\mathbf{1}^\top f$ (Hinton et al., 2015),

$$\lim_{\tau \to \infty} \tau^2 D_{\mathrm{KL}}(q_t^\tau(x) \| p_s^\tau(x')) = \frac{1}{2C} \left\| \bar{f}_t(x) - \bar{f}_s(x') \right\|_2^2 . \tag{1}$$

Thus, as the predictive distribution is softened, distillation increasingly resembles direct regression onto a fixed clean teacher target. The empirical comparison in Figure 2 follows the same trend: on CIFAR-100, shared-temperature KD improves from 30.70 NRR at $\tau = 1$ to 36.11 at $\tau = 32$, while direct regression reaches comparable performance. This motivates using the teacher's clean representation directly as a fixed target rather than relying on a softened predictive distribution. CFA adopts this fixed clean representation as its anchor (Section 4.1). We next examine what constraining this anchor implies for the student.

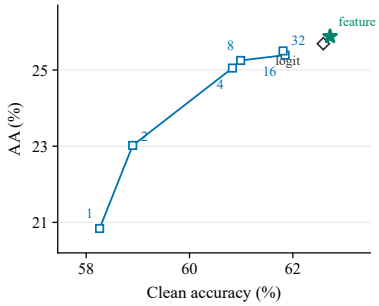

Figure 2: Shared-temperature KD and direct-regression controls on CIFAR-100. Curve labels mark τ; exact values are in Table 8.

3.2 What the Anchor Controls

The fixed clean anchor provides a common reference for the student's representations throughout the adversarial neighborhood. This raises a basic question: what quantities are constrained when the worst-case discrepancy from this reference is kept small? For $B = B(x, \epsilon)$, define

$$L = \max_{x' \in B} \|\Phi_s(x') - \Phi_t(x)\|_2, \quad F = \|\Phi_s(x) - \Phi_t(x)\|_2, \quad O = \max_{x', x'' \in B} \|\Phi_s(x') - \Phi_s(x'')\|_2. \tag{2}$$

Here, F measures the discrepancy between the student and teacher on the clean input, whereas O measures variation of the student's representation within the adversarial neighborhood.

Proposition 1 (Fixed-anchor decomposition)**.** *For every x and $\epsilon \geq 0$,*

$$L \leq F + O \leq 3L. \tag{3}$$

The triangle inequality gives $F \leq L$, $O \leq 2L$, and $L \leq F + O$; the full proof is given in Appendix A. Consequently, a single fixed-anchor objective constrains two quantities simultaneously: fidelity to the teacher on the clean input (F) and variation of the student's representation within the perturbation neighborhood (O). Importantly, this statement requires access to the teacher only at the clean input; the teacher itself need not be robust anywhere in the neighborhood. This decomposition characterizes what the objective controls, rather than providing an explanation for the measured robustness. In particular, it does not imply that an anchored student must exhibit less representation variation than a label-trained student, and our measurements do not show smaller variation (Appendix D.2). The result instead establishes that keeping the worst-case anchor discrepancy small necessarily constrains both clean fidelity and local variation relative to the same fixed reference.

Does anchoring inherit fragile features? The decomposition above concerns the student's representation relative to the teacher, but it leaves open a different concern: if the clean teacher itself relies on fragile features, does matching its representation force the student to inherit the same fragile mechanism? We study this question in a restricted version of the two-feature model of Tsipras et al. (2018). A robust feature x_1 agrees with the label with probability p and cannot be attacked, while d features $z_j \sim \mathcal{N}(\eta y, 1)$ are individually weak, jointly informative through their mean $m(z) = d^{-1} \sum_{j=1}^{d} z_j$, and fully attackable. The teacher uses only this fragile route, $\Phi_t = m(z)$, whereas the student is parameterized as $\Phi_s = a x_1 + b\, m(z)$, where $a, b \in \mathbb{R}$ are trainable coefficients for the robust and non-robust features, respectively. Thus, b measures how strongly the student retains the teacher's fragile route.

Proposition 2 (The fragile route is deleted)**.** *With the inner maximum exact, the anchor objective is $J(a, b) = \mathbb{E}[R^2] + 2\epsilon|b|\,\mathbb{E}|R| + \epsilon^2 b^2$ with $R = a x_1 + (b-1)m$. Above a threshold radius ϵ_0, every minimizer has $b^\star = 0$.*

The adversary introduces a penalty that is linear in $|b|$. As in the lasso, this term can eliminate the fragile coefficient rather than merely shrink it; under a quadratic penalty, by contrast, $b^\star > 0$ at every

radius. Thus, even though the student is trained to reproduce the teacher's clean representation, it need not reproduce the same fragile route used to compute it. The full proof, the threshold condition, and a numerical verification are provided in Appendix C.

3.3 Does Better Clean Teacher Performance Transfer to the Student?

The preceding analysis treats the clean self-teacher as a fixed reference. This naturally raises a second question: how should that reference be chosen? In particular, does a more accurate clean teacher necessarily provide a better target for the adversarially trained student?

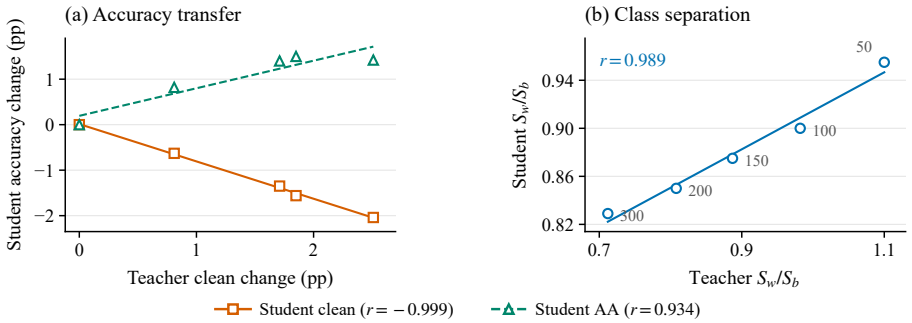

Figure 3: Transfer across clean self-teacher training stages on CIFAR-100 (ResNet-18). (a) Student clean and AutoAttack (AA) changes versus teacher clean accuracy, relative to the 50-epoch teacher. (b) Teacher and student within-to-between class scatter ratios on unit-normalized features; lower S_w/S_b indicates tighter class concentration. Labels show teacher epochs; lines are least-squares fits. Values are in Table 9.

We first vary the teacher along a single natural-training trajectory. As the teacher's clean accuracy improves, the resulting student does not exhibit a corresponding improvement in clean accuracy (Figure 3). Instead, different teacher checkpoints lead to different clean–robust operating points. The same qualitative trend is observed for an independently trained teacher trajectory (Table 17). To examine what changes beyond teacher accuracy, we measure the within-to-between scatter ratio S_w/S_b on unit-normalized clean features. Its evolution across teacher checkpoints is reflected in the corresponding students, indicating that the transferred representation changes in ways that are not captured by teacher clean accuracy alone. We treat this relationship as descriptive rather than causal. We next test whether the same conclusion holds when the teacher training procedure, rather than only its training duration, is changed. Starting from the standard 200-epoch recipe, we train additional self-teachers using label smoothing (0.1) (Szegedy et al., 2016) and mixup ($\alpha = 1.0$) (Zhang et al., 2018), using the common default for each. Both methods improve teacher clean accuracy by approximately one point and finish within 0.11 points of one another, yet their effects on the resulting students differ substantially: student clean accuracy moves in opposite directions while robustness remains nearly unchanged (Table 1). The standard teacher is the least accurate of the three, yet it transfers better than the label-smoothed teacher: its student is 3.11 points more accurate on clean inputs and 0.70 points higher in NRR. Their feature geometry also differs substantially. The scatter ratio orders the three students' clean accuracy in the same way as in the checkpoint study: label smoothing produces the tightest teacher classes ($S_w/S_b = 0.68$), whereas mixup produces substantially looser classes (1.70). Label smoothing is known to pull same-class features toward tight clusters (Müller et al., 2019), while the mixup teacher is also markedly less confident, with a mean logit margin of 2.17 compared with 4.55 for the standard teacher. These observations are consistent with differences in the learned representations, but we do not interpret them as establishing a causal mechanism for transfer. Together, the checkpoint and training-scheme comparisons show that teacher clean accuracy alone is an incomplete descriptor of transferability. Based on these transfer results, we use the mixup-trained self-teacher as the fixed reference in the remaining experiments.

Table 1: Transfer from clean self-teachers trained with different schemes on CIFAR-100 (ResNet-18; 50-epoch students).

	Teacher		Student			
Natural training	Clean	S_w/S_b	Clean	AA	Avg	NRR
Label smoothing	78.81	0.68	59.61	25.72	42.67	35.94
Standard	77.66	0.81	62.72	25.88	44.30	36.64
Mixup	78.92	1.70	63.80	25.70	44.75	36.64

4 METHOD

The analysis above motivates a fixed clean target. This section defines the objective and adds a per-sample perturbation budget.

4.1 CLEAN FEATURE ANCHORING

Let Φ_t, h_t denote the frozen clean self-teacher. The student is initialized from the teacher, $\Phi_s \leftarrow \Phi_t$ and $h_s = h_t$, after which only Φ_s is optimized. For each clean example,

$$x_{\mathrm{adv}} = \arg\max_{x' \in B(x,\epsilon)} \|\Phi_s(x') - \Phi_t(x)\|_2^2, \qquad \mathcal{L}_{\mathrm{CFA}} = \mathbb{E}_x \|\Phi_s(x_{\mathrm{adv}}) - \Phi_t(x)\|_2^2. \tag{4}$$

We approximate the inner maximization with randomly initialized PGD. Inference uses the inherited classifier $h_t \circ \Phi_s$. The full procedure is given in Algorithm 1.

4.2 SENSITIVITY-MATCHED ϵ

The fixed clean target specifies what should be preserved, but the same input-space radius need not have the same effect on the anchoring objective for every training sample. For $\ell_{\mathrm{anc}}(x'; x_i) = \|\Phi_s(x') - \Phi_t(x_i)\|_2^2$, the first-order worst-case change under an ℓ_∞ perturbation of radius e is

$$\max_{\|\delta\|_\infty \le e} \langle \nabla_{x'} \ell_{\mathrm{anc}}, \delta \rangle = e \|\nabla_{x'} \ell_{\mathrm{anc}}\|_1. \tag{5}$$

Thus, a common input-space radius can induce different changes in the anchoring objective when samples have different local sensitivities. Motivated by this observation, we assign each sample a sensitivity score $g_i > 0$ based on the gradient magnitude of the anchoring objective. In our implementation, we use the ℓ_2 norm of the input gradient as the sensitivity score. More sensitive samples receive smaller radii, whereas less sensitive samples receive larger ones. Specifically, we set

$$\epsilon_i = N\epsilon \frac{g_i^{-p}}{\sum_{j=1}^N g_j^{-p}}, \tag{6}$$

where N is the batch size, ϵ is the target mean radius, and p controls the strength of sensitivity matching. When $p = 0$, all samples receive the same radius ϵ; for $p > 0$, the radius decreases with sensitivity. The normalization ensures $\frac{1}{N}\sum_{i=1}^N \epsilon_i = \epsilon$, so the average perturbation budget is unchanged. For $p = 1$, the allocation is inversely proportional to the sensitivity score, so samples with larger local gradients receive proportionally smaller perturbation radii. The first-order analysis above motivates this inverse allocation, while the gradient norm used in practice serves as a local sensitivity proxy rather than an exact equalization of the nonlinear anchoring loss. We evaluate the effect of sensitivity matching and the importance of assigning the resulting radii to their corresponding samples in Tables 5 and 6. Details on sensitivity estimation, radius clipping, and sensitivity-norm choices are provided in Appendix F.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

We evaluate CFA on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and Tiny-ImageNet-200 (Le & Yang, 2015). We use ResNet-18 as the default architecture and additionally report results with

WideResNet-34-10. Unless stated otherwise, the clean self-teacher is trained on natural images and then frozen. Following the transfer analysis in Section 3.3, we select the teacher based on its transfer to the robust student rather than its clean accuracy alone, and use the mixup-trained self-teacher in the main experiments. The student is initialized from the teacher, with the inherited classifier kept fixed during adversarial training. The main CIFAR experiments use 150 training epochs, AdamW with a one-cycle learning-rate schedule, 10-step projected gradient descent (PGD-10), $\epsilon_{\text{train}} = 8/255$, sensitivity-matched perturbation radii with $p = 1$, weight averaging (WA), and adversarial weight perturbation (AWP). For comparison, we include standard adversarial training baselines (PGD-AT, TRADES, and MART), adversarial distillation methods (ARD, RSLAD, AdaAD, and AdaAD+IGDM), and methods that are particularly relevant to CFA's clean–robust setting. The latter include LBGAT and ARREST, which use natural teachers; HAT and RPAT++, which target the clean–robust trade-off; and ADR, which uses self-distillation. We use the respective training recipes of each baseline, with the same natural teacher as CFA for distillation methods where applicable. We report clean accuracy and AutoAttack (AA) accuracy on the full test set at $\epsilon = 8/255$, together with their arithmetic mean (Avg) and harmonic mean (NRR). Dataset-specific settings, teacher statistics, attack diagnostics, and additional controls are provided in Appendices G and J.

5.2 Main Results

Table 2: ResNet-18 results on CIFAR-10 and CIFAR-100 under ℓ_∞ attacks at $\epsilon = 8/255$. NRR is the harmonic mean of clean and AutoAttack (AA) accuracy. The CIFAR-100 CFA row reports the mean of three student seeds; all other rows are individual runs.

	CIFAR-10				CIFAR-100			
Method	Clean	AA	Avg	NRR	Clean	AA	Avg	NRR
PGD-AT	84.54	43.57	64.06	57.50	57.36	20.34	38.85	30.03
TRADES	81.71	49.16	65.44	61.39	55.28	23.55	39.42	33.03
MART	80.41	45.56	62.99	58.16	53.86	22.72	38.29	31.96
ARD	84.58	43.75	64.16	57.67	57.61	20.24	38.93	29.96
RSLAD	85.15	42.91	64.03	57.06	59.68	21.30	40.49	31.40
AdaAD	85.39	46.22	65.81	59.98	59.79	23.19	41.49	33.42
AdaAD + IGDM	85.01	45.50	65.26	59.27	48.25	19.48	33.87	27.75
LBGAT	84.52	47.69	66.10	60.98	57.29	26.22	41.75	35.98
ARREST	85.06	46.42	65.74	60.06	66.97	22.78	44.88	34.00
HAT	84.59	48.81	66.70	61.90	60.67	24.45	42.56	34.85
ADR	83.36	49.72	66.54	62.29	59.93	27.20	43.57	37.42
RPAT++	82.41	50.75	66.58	62.82	55.93	27.36	41.64	36.74
CFA (ours)	87.37	51.33	**69.35**	**64.67**	64.82	28.04	**46.43**	**39.15**

On CIFAR-100, CFA achieves 64.82% clean accuracy and 28.04% AA, with an NRR of 39.15. Compared with ADR, the closest baseline in NRR, CFA improves clean accuracy by 4.89 points, AA by 0.84 points, and NRR by 1.73 points. On CIFAR-10, CFA achieves 87.37% clean accuracy and 51.33% AA, with an NRR of 64.67, 1.85 points above the closest baseline (Table 2). We additionally evaluate CFA on Tiny-ImageNet-200 and with WideResNet-34-10 to examine whether the results persist beyond the default CIFAR/ResNet-18 setting. On Tiny-ImageNet-200 with ResNet-18, CFA achieves 55.16% clean accuracy and 20.54% AA, yielding an NRR of 29.93. With WideResNet-34-10 on CIFAR-10, CFA achieves 88.67% clean accuracy and 55.29% AA, with an NRR of 68.11 (Table 3). On CIFAR-100 with the same architecture, CFA obtains $65.04/30.14$ clean/AA, compared with $64.50/29.04$ for ADR (Table 14).

5.3 Ablation and Component Analysis

Clean feature anchoring. We first isolate the effect of the anchor objective by removing sensitivity matching, weight averaging, and AWP. With the remaining training recipe held fixed, CFA improves both clean accuracy and AA over label-based adversarial training (Table 18). The improvement remains when the cross-entropy baseline uses its own optimizer and training schedule. We further examine the feature geometry induced by the two objectives. We measure the within-to-

Table 3: Results on Tiny-ImageNet-200 with ResNet-18 and CIFAR-10 with WideResNet-34-10, measured in our framework.

	Tiny-ImageNet, ResNet-18				CIFAR-10, WRN-34-10			
Method	Clean	AA	Avg	NRR	Clean	AA	Avg	NRR
PGD-AT	49.72	14.45	32.09	22.39	86.00	45.71	65.86	59.69
TRADES	46.98	15.95	31.46	23.81	85.13	48.99	67.06	62.19
MART	46.01	16.90	31.46	24.73	85.29	47.06	66.18	60.65
ARD	49.74	14.48	32.11	22.43	86.35	45.52	65.94	59.61
RSLAD	51.26	17.02	34.14	25.56	86.41	45.80	66.10	59.87
AdaAD	51.59	16.79	34.19	25.33	86.80	46.84	66.82	60.85
AdaAD + IGDM	50.87	16.21	33.54	24.59	84.92	45.93	65.42	59.62
LBGAT	51.04	18.47	34.75	27.13	81.56	51.79	66.68	63.35
ARREST	59.38	16.06	37.72	25.28	87.54	48.83	68.19	62.69
HAT	52.57	17.28	34.93	26.01	87.15	53.21	70.18	66.07
ADR	47.66	20.45	34.06	28.62	88.64	52.34	70.49	65.82
RPAT++	46.90	20.10	33.50	28.14	87.60	54.45	71.03	67.16
CFA (ours)	55.16	20.54	**37.85**	**29.93**	88.67	55.29	**71.98**	**68.11**

Table 4: Feature geometry of CFA and label-based adversarial training on CIFAR-100. *Own* is the mean sample-to-own-centroid angle, and *Gap* is the mean angle from each class centroid to its nearest other-class centroid. Lower S_w/S_b and *Own* indicate tighter within-class concentration, whereas larger *Gap* indicates greater separation between class centroids. Angles are measured on unit-normalized features.

	Clean			Adversarial		
	S_w/S_b	Own	Gap	S_w/S_b	Own	Gap
CFA	0.709	26.3°	29.2°	1.969	33.5°	19.2°
Label CE	3.744	52.2°	33.5°	5.611	55.0°	28.2°

between class scatter ratio S_w/S_b, the mean angle of each feature to its own class centroid (*Own*), and the mean angle from each class centroid to its nearest other-class centroid (*Gap*). As shown in Table 4, CFA produces substantially tighter within-class feature concentration on both clean and adversarial inputs, while its centroid gaps are not larger than those of the label-trained model. The geometric difference is therefore dominated by tighter within-class structure rather than greater angular separation between classes. To verify that the gain is not explained by teacher initialization, weight averaging, or AWP alone, we additionally apply these components to the baseline methods. Although they improve most baselines, none reaches CFA's NRR (Table 10).

Sensitivity-matched radius allocation. The sensitivity-matched radius is intended to account for the fact that the same input-space radius can affect the anchoring objective differently across samples. We first isolate its effect by comparing a uniform radius ($p = 0$) with sensitivity matching ($p = 1$), while holding the teacher, training schedule, mean radius, WA, and AWP fixed. Sensitivity

Table 5: Uniform versus sensitivity-matched perturbation radii on CIFAR-100 with ResNet-18 (150 epochs; training radius $8/255$). Values are mean $\pm$ sample standard deviation over three student seeds.

Radius allocation	Clean	AA	NRR
Uniform ($p = 0$)	62.63 ± 0.22	28.01 ± 0.20	38.71 ± 0.21
Sensitivity-matched ($p = 1$)	64.82 ± 0.07	28.04 ± 0.09	$\mathbf{39.15 \pm 0.09}$

matching increases clean accuracy by 2.19 points while essentially preserving AA (28.01 vs. 28.04), increasing NRR from 38.71 to 39.15. Thus, the adaptive allocation improves clean retention without sacrificing adversarial accuracy under the same average perturbation budget.

Does the allocation signal matter? The preceding comparison changes both the radii assigned to individual samples and their correspondence with local sensitivity. We therefore ask whether the gain comes simply from using heterogeneous radii or from assigning them according to the sensitivity signal. The "same multipliers, difficulty order" control preserves the exact set of sensitivity-

Table 6: Comparison of perturbation-radius allocation signals on CIFAR-100 with ResNet-18 (100 epochs; weight averaging and adversarial weight perturbation). $\mathrm{sd}(w)$ measures the dispersion of radius multipliers. Alternative signals are evaluated within the same allocation framework and are not reimplementations of prior methods.

Allocation	$\mathrm{sd}(w)$	Clean	AA	NRR
Uniform	0	60.42	28.42	38.66
Sensitivity	0.32	62.17	28.86	**39.42**
Same multipliers, difficulty order	0.32	60.90	28.06	38.42
Clean CE difficulty	0.39	61.27	28.12	38.55
Logit margin	0.42	61.07	27.79	38.20

derived radii and changes only which samples receive them. This reassignment reduces AA from 28.86 to 28.06 and NRR from 39.42 to 38.42. Alternative allocation signals also fail to recover the sensitivity-matched result. These comparisons indicate that the benefit does not arise from radius heterogeneity alone; the correspondence between local sensitivity and the assigned perturbation neighborhood is important. Additional recipe, radius, and WA/AWP controls are reported in Tables 10, 13 and 18.

6 Conclusion

We have introduced CFA as a way to mitigate the clean–robust trade-off by separating the learning of clean knowledge from the learning of robustness. A naturally trained self-teacher provides a fixed clean reference, and the student learns robustness by preserving this reference over its adversarial neighborhood. Our analysis characterizes the joint control of clean fidelity and local representation variation induced by the anchor, and shows that teacher clean accuracy alone does not determine how well its knowledge transfers to the robust student. Sensitivity-matched radii complement the fixed anchor by redistributing a fixed average perturbation budget according to the local response of the training objective. Across datasets and architectures, CFA consistently retains high clean accuracy while maintaining strong adversarial robustness, and the ablations show that both the fixed clean anchor and the correspondence between local sensitivity and perturbation radius contribute to this trade-off. These results support clean self-teaching as a practical approach to robust training without requiring an adversarial teacher or a separate label objective during the robust stage. At the same time, teacher accuracy alone does not identify the most transferable clean representation. Understanding which properties of clean representations determine their transferability under adversarial training remains an important direction for future work.

A Proof and Scope of the Fixed-Anchor Decomposition

Proof of Proposition 1. For any $x' \in B(x, \epsilon)$, the triangle inequality gives

$$\|\Phi_s(x') - \Phi_t(x)\|_2 \le \|\Phi_s(x') - \Phi_s(x)\|_2 + \|\Phi_s(x) - \Phi_t(x)\|_2 \le O + F,$$

so $L \le F + O$. Since $x \in B(x, \epsilon)$, $F \le L$. For any $x', x'' \in B(x, \epsilon)$,

$$\|\Phi_s(x') - \Phi_s(x'')\|_2 \le \|\Phi_s(x') - \Phi_t(x)\|_2 + \|\Phi_s(x'') - \Phi_t(x)\|_2 \le 2L,$$

so $O \le 2L$ and therefore $F + O \le 3L$. □

The result concerns exact maxima over the perturbation set; finite-step PGD does not certify them. It applies to any fixed target in a normed space, including logits, and does not imply that features are superior to logits or that the bound alone guarantees robust classification.

B Algorithm

Algorithm 1 Clean Feature Anchoring (CFA)

Input: dataset $\mathcal{D}$, radius ϵ, attack steps K, step size a, exponent p, clipping thresholds $[w_{\mathrm{lo}}, w_{\mathrm{hi}}]$

Output: robust student $h_t \circ \Phi_s$

{*Phase 1 — natural self-teacher.* The only stage that uses labels.}
Train $f_t = h_t \circ \Phi_t$ on clean data with mixup and cross-entropy; freeze it

{*Phase 2 — anchored adversarial training.* No labels, no head update.}
$\Phi_s \leftarrow \Phi_t$ {student initialized at the teacher}
for epoch $= 1$ **to** T **do**
  for minibatch $\{x_i\}_{i=1}^N \subset \mathcal{D}$ **do**
    $\phi_i \leftarrow \Phi_t(x_i)$ {teacher read once, at the clean input}
    $g_i \leftarrow \max(\|\nabla_x \|\Phi_s(x_i) - \phi_i\|^2\|_2, 10^{-12})$ {one backward pass}
    $v_i \leftarrow \mathrm{clip}((\bar{g}/g_i)^p, w_{\mathrm{lo}}, w_{\mathrm{hi}})$; $w_i \leftarrow N v_i / \sum_j v_j$
    $\epsilon_i \leftarrow \epsilon w_i$; $a_i \leftarrow a w_i$ $\{\sum_i \epsilon_i = N\epsilon\}$
    $x_i^0 \leftarrow x_i + \eta_i$
    for $k = 0$ **to** $K - 1$ **do**
      $x_i^{k+1} \leftarrow \Pi_{B(x_i, \epsilon_i)}\big(x_i^k + a_i \operatorname{sign} \nabla_x \|\Phi_s(x_i^k) - \phi_i\|^2\big)$
    end for
    $\mathcal{L} \leftarrow \frac{1}{N} \sum_i \|\Phi_s(x_i^K) - \phi_i\|^2$
    $\Phi_s \leftarrow \Phi_s - \mu \nabla_{\Phi_s} \mathcal{L}$ {h_t is never updated}
  end for
end for

C Two Partial Results on the Robustness Side

Natural-teacher guidance is shared with prior work; our question is whether a fixed clean target can support adversarial training without an additional label objective. In CFA, the teacher specifies the target while the student's inner maximization enforces matching under perturbations. Teacher vulnerability alone therefore does not determine the student's response: matching a clean target is different from reproducing the teacher at perturbed inputs.

The two results below examine this distinction in restricted settings. The first studies robust and non-robust features in a two-feature model; the second relates anchor error to a classification margin. They supplement the fixed-anchor decomposition, but do not predict the measured accuracy or establish superiority over other regression targets.

C.1 In the Two-Feature Model, the Non-Robust Coefficient Is Zeroed Rather Than Shrunk

We use a restricted version of the feature model of Tsipras et al. (2018). One feature x_1 agrees with the label with probability $p = 0.95$; here, we assume it cannot be attacked. A further d features $z_j \sim \mathcal{N}(\eta y, 1)$ are individually almost uninformative, are jointly close to perfectly informative through their mean $m(z)$, and can all be moved by an ℓ_∞ adversary. We take $\Phi_t = m(z)$ as a natural teacher with near-perfect clean accuracy and poor robustness.

Write the student's feature as $\Phi_s = ax_1 + b\,m(z)$. Then b is exactly the quantity the objection is about: how much of the teacher's fragile route the student takes over. If the objection is right, minimizing our objective should leave b large.

Proposition 2 in Section 3.2 states the result: the one-sided slopes of J at $b = 0$ differ by $4\epsilon\,\mathbb{E}|R(a,0)|$, and above a threshold ϵ_0 every minimizer has $b^\star = 0$.

Proof of Proposition 2. The adversary reaches Φ_s only through m, and only through the mean perturbation $\bar\delta = \frac{1}{d}\sum_j \delta_j$, which ranges over $[-\epsilon, \epsilon]$ as $\|\delta\|_\infty \le \epsilon$. The inner objective is therefore $\max_{|\bar\delta|\le\epsilon}(R + b\bar\delta)^2 = (|R| + \epsilon|b|)^2$, since a scalar quadratic is maximized at an endpoint and the sign of $\bar\delta$ is free. Taking expectations and expanding the square gives the display.

Fix a and differentiate in b. The term $\mathbb{E}[R^2]$ and $\epsilon^2 b^2$ are smooth, while $2\epsilon|b|\,\mathbb{E}|R|$ contributes $\pm 2\epsilon\,\mathbb{E}|R|$ as $b \to 0^\pm$. The one-sided derivatives at $b = 0$ are thus $2\mathbb{E}[R(a,0)\,m] \pm 2\epsilon\,\mathbb{E}|R(a,0)|$, so $b = 0$ is a minimum along b exactly when

$$\big|\mathbb{E}[R(a,0)\,m]\big| \;\le\; \epsilon\,\mathbb{E}|R(a,0)|. \tag{7}$$

At $b = 0$ the objective in a is $\mathbb{E}(ax_1 - m)^2$. Writing $q = 2p - 1$ for the mean label-alignment of x_1, it is minimized at $a^\star = \mathbb{E}[x_1 m]/\mathbb{E}[x_1^2] = q\eta$, and there $\mathbb{E}[R(a^\star,0)\,m] = q^2\eta^2 - (\eta^2 + \frac{1}{d}) = -c$ with

$$c \;=\; \eta^2 v + \tfrac{1}{d}, \qquad v = 1 - q^2. \tag{8}$$

Equation (7) therefore holds for every

$$\epsilon \;\ge\; \epsilon_0 \;=\; \frac{c}{\mathbb{E}|q\eta x_1 - m|}, \tag{9}$$

which is the stated threshold. For fixed x_1, m and $\bar\delta \in [-\epsilon, \epsilon]$, $(ax_1 + (b-1)m + b\bar\delta)^2$ is convex in (a, b). Its maximum over $\bar\delta$ and then its expectation are convex, so J is jointly convex. At $(a^\star, 0)$ the a-derivative vanishes, and Equation (7) puts zero in the b-subgradient; hence this point is a global minimizer. If $\epsilon > \epsilon_0$, the directional derivative from $(a^\star, 0)$ is strictly positive in every direction with nonzero b. Along $b = 0$, $\mathbb{E}(ax_1 - m)^2$ has a unique minimizer. Thus every global minimizer has $b = 0$ above the threshold. □

Why the coefficient is deleted rather than shrunk. The adversary contributes $2\epsilon|b|\,\mathbb{E}|R|$ to the loss — linear in $|b|$, not quadratic in b. A quadratic penalty pushes a coefficient toward zero and never arrives; a penalty on $|b|$ is kinked at the origin and sets the coefficient to zero outright once its slope dominates, which is the same distinction that makes the lasso sparse and the ridge not. What produces the kink here is that the student is asked to reproduce a *value* rather than a route: the fragile features carry that value on clean data but not inside the ball, so past a certain radius carrying any of them costs more than it pays. The teacher supplies what to match, and the inner maximization discards how the teacher got there.

At $d = 512$ the model's constants give $c = 7.89 \times 10^{-3}$ and $\mathbb{E}|q\eta x_1 - m| = 0.0529$, hence $\epsilon_0 = 0.843\,\eta$ — inside the bracket $(0.5\eta,\ 1.0\eta)$ that the Monte-Carlo minimization returns, and far below the 2η at which the model is usually stated.

The model assumes x_1 is unperturbable and z fully perturbable, so some suppression of z is built in. The proposition establishes the exact threshold behavior within this restricted model; it does not describe the geometry of real image classifiers.

That the exactness is the kink's doing, and not a general consequence of adding an adversary, is visible from dropping it. With the $|b|$ term removed, $J_{\text{quad}} = \mathbb{E}[R^2] + \epsilon^2 b^2$; minimizing out a and

writing $u = 1 - b$ collapses it to $u^2 c + \epsilon^2 (1-u)^2$, so

$$b^\star_{\text{quad}} \;=\; \frac{c}{c + \epsilon^2}, \tag{10}$$

which is strictly positive at every radius. Against the exact minimizer ($d = 512$, Monte-Carlo, $n = 2 \times 10^5$):

ϵ	$b^\star$ (exact)	$b^\star_{\text{quad}}$ (kink dropped)
0.5η	0.510	0.502
1.0η	**0.000**	0.202
2.0η	**0.000**	0.060
4.0η	**0.000**	0.016

The two columns agree below ϵ_0 and separate above it: the quadratic surrogate shrinks the non-robust coefficient and never deletes it, which is the ordinary picture, while the kinked objective sets it to zero and keeps it there.

Since every ϵ in J multiplies b, at $b^\star = 0$ the objective reduces to $\mathbb{E}(ax_1 - m)^2$ and no longer depends on the radius. This saturation is a property of the two-feature model, not a prediction of the empirical radius sweep in Table 13.

The two-feature model cannot explain the measured clean–robust trade-off: it has only a robust and a fragile direction. It shows the narrower point that a robust student need not copy the fragile route by which its natural teacher computes the clean target.

C.2 What Robust Accuracy Tracks Is the Loss Measured Against a Margin

Writing $\gamma_t(x)$ for the teacher's margin at x and D_y for the relevant decision-boundary constant, Cauchy–Schwarz gives that $\gamma_t(x) > D_y \cdot L(x)$ implies the student is robustly correct at x. The certificate is vacuous at the scales we train at, but the ratio DL/γ_t tracks both axes across the teacher ladder of Figure 3. What it does not do is predict the size of the effect, and it saturates before AutoAttack does.

D Supplementary Analysis of the Clean-Feature Anchor

D.1 Teacher Evaluation on Perturbed Inputs

The fixed-target comparison in Section 3.1 evaluates the teacher only at the clean input. This choice is important because the natural teacher is not locally robust: under $\epsilon = 8/255$, its feature rotates by 63.8° and its norm grows by a factor of 2.45. The main-paper control in Table 7 shows the consequence of replacing the fixed clean target with the teacher's perturbed representation. Matching $\Phi_t(x_{\text{adv}})$ makes an exact copy of the vulnerable teacher a zero-loss solution and yields zero AutoAttack accuracy, whereas matching the fixed clean target retains nonzero robustness. These diagnostics reinforce that the teacher supplies a clean reference value rather than a robust perturbed response.

Table 7: Teacher-target comparison on CIFAR-100 with ResNet-18. Both variants use the same 50-epoch setting; the teacher is evaluated on either the clean or adversarial input.

Teacher target	Clean	AA	NRR
$\Phi_t(x)$	61.33	26.19	**36.71**
$\Phi_t(x_{\text{adv}})$	76.11	0.00	0.00

D.2 Additional Evidence for the Anchor Objective

We measure feature geometry on the final 150-epoch CFA model and a label-trained student with the same natural initialization, weight averaging, and AWP. Both models are evaluated on 5,000 test images under the same 10-step true-label PGD attack at $8/255$; all angular statistics use unit-normalized features.

Proposition 1 bounds variation within an anchored model, but does not predict a smaller variation than label training. Under the matched true-label PGD protocol, the anchor and label-trained students move by 0.312 and 0.281 of their clean feature norms. These displacements are not the worst-case diameter O and provide no evidence of smaller relative displacement under anchoring. The reproducible difference is instead within-class concentration: CFA has a $5.3\times$ smaller clean S_w/S_b and a $2.85\times$ smaller ratio under attack (Table 4). Its class centroids are not farther apart than those learned with labels, so the result supports tighter clusters rather than greater inter-class separation.

The classification benefit is evaluated in Section 5.3. Without weight averaging, AWP, or the sensitivity-matched radius, the anchor improves clean and AutoAttack accuracy over label cross-entropy by 6.69 and 5.91 points under the same training conditions. The gains are 4.63 and 4.45 points when cross-entropy uses its own optimizer and schedule (Table 18). Under a matched attack, CFA has slightly larger relative feature displacement and smaller centroid gaps, but substantially tighter class-conditional features. This association does not establish concentration as the cause of the robustness gain.

The checkpoint ladder in Figure 3 further separates within-class concentration from between-class separation on unit-normalized features. From the 50-epoch to the 300-epoch teacher, the mean angle from a sample to its class centroid decreases from 34.6° to 30.9°, whereas the nearest-other-centroid angle between class means increases from 35.7° to 44.9°. The corresponding students reproduce the latter change—their nearest-centroid separation increases from 25.4° to 29.8°—while their mean within-class angle remains nearly unchanged (27.9° to 28.2°). These measurements motivate interpreting the ladder through between-class geometry, but they remain observational because the teacher's accuracy and geometry change together. The numerical values underlying Figure 3 and the target comparison of Section 3.1 are collected below.

The target comparison uses the 50-epoch base regime at $\epsilon = 8/255$, without weight averaging or AWP.

Table 8: Distillation and regression targets on CIFAR-100 with ResNet-18. $H(q_t^\tau)$ is teacher-target entropy in nats; NRR is the harmonic mean of clean and AutoAttack accuracy.

Target	τ	$H(q_t^\tau)$	Clean	AA	NRR
Teacher-only temperature					
KL	1	0.76	58.26	20.84	30.70
KL	2	2.40	59.33	22.79	32.93
KL	4	4.20	59.39	24.48	34.67
KL	8	4.56	59.47	25.19	35.39
KL	16	4.60	57.78	24.00	33.91
Shared-temperature KD					
KL	1	0.76	58.26	20.84	30.70
KL	2	2.40	58.90	23.02	33.10
KL	4	4.20	60.83	25.05	35.49
KL	8	4.56	60.99	25.25	35.71
KL	16	4.60	61.85	25.39	36.00
KL	32	4.60	61.81	25.50	36.11
Logit regression, fixed head	—	—	62.59	25.69	36.43
Feature anchor, fixed head	—	—	62.72	25.88	**36.64**

Table 9: Clean self-teacher checkpoints and resulting students on CIFAR-100 with ResNet-18. Values correspond to Figure 3.

	Teacher		Student		
Checkpoint	Clean	S_w/S_b	Clean	S_w/S_b	AA
50 epochs	75.81	1.100	64.28	0.955	24.38
100 epochs	76.62	0.982	63.65	0.900	25.20
150 epochs	77.52	0.887	62.93	0.875	25.78
200 epochs	77.66	0.808	62.72	0.850	25.88
300 epochs	78.32	0.712	62.24	0.829	25.80

E Additional Results

Baseline training-stack control. To test whether CFA's final performance can be explained by teacher initialization, weight averaging, and AWP alone, we add the same components to the baseline recipes. Most baselines improve, but none reaches CFA's NRR.

Table 10: Baseline results on CIFAR-100 with ResNet-18 before and after adding teacher initialization, weight averaging, and AWP. ‡RPAT++ already includes weight averaging; its right-hand result adds the other two components and uses the final checkpoint.

	Own recipe			+ teacher init + WA + AWP		
Method	Clean	AA	NRR	Clean	AA	NRR
PGD-AT	57.36	20.34	30.03	59.85	25.77	36.03
TRADES	55.28	23.55	33.03	56.16	24.56	34.17
MART	53.86	22.72	31.96	50.97	25.13	33.66
ARD	57.61	20.24	29.96	60.19	26.10	36.41
RSLAD	59.68	21.30	31.40	63.44	25.55	36.43
AdaAD	59.79	23.19	33.42	57.21	27.10	36.78
AdaAD + IGDM	48.25	19.48	27.75	54.24	25.56	34.75
Consistency-AT	57.53	20.01	29.69	60.74	25.38	35.80
LBGAT	57.29	26.22	35.98	55.75	25.57	35.06
ADR	59.93	27.20	37.42	59.84	27.84	38.00
HAT	60.67	24.45	34.85	57.63	24.74	34.62
ARREST	66.97	22.78	34.00	64.95	23.02	33.99
RPAT++‡	55.93	27.36	36.74	57.83	27.47	37.25
CFA (ours)		—		64.82	28.04	**39.15**

The following tables provide additional results supporting the main experiments and supplementary analysis.

Regression under weight averaging and AWP. Table 11 compares logit and feature regression with the same natural teacher, inherited frozen classifier, AdamW optimizer, one-cycle schedule, WA, and AWP. Both use a uniform radius: neither run includes sensitivity-based allocation. Each objective is maximized to generate training perturbations and minimized to update the backbone, so the comparison includes its effect on both stages. Logit regression remains competitive with the training stack; the single-seed difference does not establish a general advantage for feature regression.

Table 11: Clean-logit and clean-feature regression on CIFAR-100 with ResNet-18 (100 epochs, weight averaging, AWP, and a uniform training radius of $8.8/255$). Both use the inherited frozen classifier; AutoAttack is evaluated at $8/255$.

Target	Clean	AA
Clean logits	60.36	28.02
Clean features	60.42	28.42

Table 12: Classifier-head choices for clean feature anchoring on CIFAR-100 with ResNet-18 (50 epochs, $\epsilon = 8/255$). The comparison uses raw features without weight averaging or AWP.

	Clean	PGD-20	CW	AA	NRR
Logit target only, $\tau = 1$	58.26	22.62	22.79	20.84	30.70
Logit target only, $\tau = 4$	59.39	30.30	26.84	24.48	34.67
Logit target only, $\tau = 16$	57.78	31.57	26.11	24.00	33.91
Feature anchor + head KD, $\tau = 1$	62.34	26.34	26.11	23.93	34.58
Feature anchor + head KD, $\tau = 16$	62.61	32.47	27.32	25.61	36.35
Feature anchor, head untouched	62.72	28.69	27.80	25.88	**36.64**
head refit, adversarial CE	62.77	29.93	27.66	25.65	36.42
head refit, adv. CE + smoothing 0.1	62.57	30.43	27.55	25.66	36.39
head refit, clean CE	62.70	28.90	27.23	25.15	35.90

Table 13: Training-radius comparison on CIFAR-10 and CIFAR-100 with ResNet-18. All results are evaluated at $\epsilon = 8/255$. The $8.8/255$ rows use the corrected AWP implementation; the other rows predate the correction (Appendix L).

	Clean	PGD-20	CW	AA	NRR
CIFAR-100					
$6/255$	68.07	28.40	28.16	25.82	37.44
$7/255$	65.61	30.34	29.64	27.44	38.70
$8/255$	63.89	31.41	29.83	27.78	38.72
$8.8/255^{\dagger}$	62.17	32.37	30.93	28.86	**39.42**
$10/255^{\dagger}$	59.99	32.82	30.53	28.77	38.89
CIFAR-10					
$6/255$	89.64	49.29	50.20	47.88	62.42
$7/255$	88.30	51.43	52.64	50.27	64.07
$8/255^{\dagger}$	87.22	52.43	53.55	51.15	**64.48**
$8.8/255^{\dagger}$	84.96	52.98	53.94	51.74	64.31
$10/255^{\dagger}$	83.29	53.25	53.88	51.79	63.87

F The Allocation Rule and Its Budget Solve

F.1 Where the rule comes from

As in Section 4.2, the fixed clean anchor is optimized over an input-space ℓ_∞ ball, but a common radius need not change the anchoring loss by the same amount for every sample. The first-order worst-case change in a differentiable loss $\mathcal{L}$ within a radius e is

$$\max_{\|\delta\|_\infty \le e} \langle \nabla_x \mathcal{L},\, \delta \rangle \;=\; e\, \|\nabla_x \mathcal{L}\|_1, \tag{11}$$

because the dual norm of ℓ_∞ is ℓ_1. If this local first-order change were equalized across a batch under $\sum_i \epsilon_i = N\epsilon$, the radii would satisfy $\epsilon_i \propto 1/g_i$ for $g_i = \|\nabla_x \mathcal{L}(x_i)\|_1$. This gives the theoretical motivation for inverse sensitivity matching. The reported implementation instead uses the ℓ_2 norm of the input gradient of the raw squared-feature anchoring loss as a proxy for g_i. It applies $\epsilon_i \propto g_i^{-p}$, with $p = 0$ giving uniform radii and $p = 1$ giving inverse allocation. Consequently, the implemented rule does not exactly equalize the first-order bound in Equation (11).

The proxy is measured once at the clean input before each attack, so it does not capture sensitivity along the PGD trajectory. The derivation itself is also first order. These limits make the allocation a local heuristic, rather than an exact solution of the nonlinear inner maximization. Its signal comes from the training objective, whereas the alternative signals in Table 6 use clean cross-entropy difficulty or logit margin.

The separation is not one of dispersion, which is worth stating because a rule that simply spread the budget more aggressively would be the simpler explanation of any gain. Measured over training on the reported configuration, the coefficient of variation of our sensitivity signal has median 0.56, against 0.75 for the difficulty score IAAT and CAT allocate from and 1.33 for the logit margin of MMA. The box of this section compresses all three before they reach the attack, leaving per-sample multipliers of spread 0.32, 0.39 and 0.42 respectively, and the ordering survives the compression. Ours is the least dispersed of the three signals either way, so the comparison in Table 6 is a comparison of which geometry the budget follows.

The choice of gradient norm also changes the radius distribution. Measured over training on the reported configuration, the ℓ_1 signal's coefficient of variation has median 0.68 against the ℓ_2 signal's 0.56, and the per-sample multiplier the attack receives has spread 0.34 against 0.32 — so the norms really do distribute the budget differently.

They nonetheless land close together. Training the reported recipe with the ℓ_1 gradient norm in place of ℓ_2, and nothing else changed, gives $62.16/31.97/30.56/28.57$ (NRR 39.15) against $62.17/32.37/30.93/28.86$ (NRR 39.42): clean accuracy is identical to a hundredth and AutoAttack differs by 0.29, in favour of the norm we ship.

The difference is small but non-negligible relative to the observed seed variation. It is twice the seed-to-seed spread of Appendix G, and it is the same order of magnitude as what the allocation buys in the first place: against the uniform radius, $p = 1$ gains $+1.75$ clean and $+0.44$ AutoAttack at ℓ_2 and $+1.74$ and $+0.15$ at ℓ_1. So the rule's benefit survives the norm in direction and almost entirely on the clean axis, while most of its AutoAttack half does not. Thus the sensitivity norm is a genuine implementation choice: the reported ℓ_2 variant performs slightly better in AA, while the clean-accuracy gain persists with the theoretically matched ℓ_1 norm.

The permuted control in Table 6 isolates the assignment signal from the radius distribution. It takes the multiset of weights the gradient rule produces — the same values, the same clip $[0.5, 1.5]$, the same restored mean — and only permutes which example receives which, so that the larger radii go to the easier examples, which is the direction IAAT and CAT assign them. Reordering costs 0.80 AutoAttack against the rule it reorders and lands below even the uniform row on AutoAttack, CW and NRR alike.

F.2 Clipping and restoring the mean radius

The reported configuration floors the gradient norm at 10^{-12} and computes $v_i = \mathrm{clip}((\bar{g}/g_i)^p, 0.5, 1.5)$, followed by $w_i = N v_i / \sum_j v_j$ and $\epsilon_i = \epsilon w_i$. PGD uses step size $a_i = a w_i$. This preserves the batch mean radius exactly. The clipping thresholds apply before normalization, so they are not hard bounds on the final multipliers. When all gradients vanish, the floor makes all g_i equal and the allocation uniform.

Optional exact-box variant. The implementation also supports an alternative that constrains the final multipliers. It is not enabled in the reported default configuration.

With $r_i = (\bar{g}/g_i)^p$ and box $[w_{\mathrm{lo}}, w_{\mathrm{hi}}] = [0.5, 1.5]$, the radii that satisfy both the box and the budget exactly are $\epsilon_i = \epsilon \cdot \mathrm{clip}(t r_i, w_{\mathrm{lo}}, w_{\mathrm{hi}})$ with t the solution of

$$\sum_{i=1}^{N} \mathrm{clip}\big(t\, r_i,\ w_{\mathrm{lo}},\ w_{\mathrm{hi}}\big) \ =\ N. \tag{12}$$

The left side is continuous and nondecreasing in t from $N w_{\mathrm{lo}}$ to $N w_{\mathrm{hi}}$, so a root exists and bisection finds it; when no clip is active it reduces to $t = N / \sum_i r_i$, that is, to restoring the mean. Clipping first and rescaling afterwards, the obvious alternative, does not preserve the box—the rescaling moves the clipped entries back out of it.

Table 14: Results on CIFAR-100 with WideResNet-34-10 at $\epsilon = 8/255$, measured in our framework.

Method	Clean	AA	Avg	NRR
PGD-AT	58.97	22.84	40.91	32.93
TRADES	57.35	25.60	41.48	35.40
MART	57.16	22.43	39.80	32.22
ARD	59.52	23.05	41.29	33.23
RSLAD	60.48	21.63	41.06	31.87
AdaAD	61.66	25.62	43.64	36.20
AdaAD + IGDM	60.84	25.03	42.94	35.47
LBGAT	59.63	28.55	44.09	38.61
ARREST	67.69	24.69	46.19	36.18
HAT	62.81	26.91	44.86	37.68
ADR	64.50	29.04	46.77	40.05
RPAT++	63.10	28.72	45.91	39.47
CFA (ours)	**65.04**	**30.14**	**47.59**	**41.19**

Table 15: Standard-training clean teacher checkpoints for CIFAR-10, CIFAR-100, and Tiny-ImageNet-200 with ResNet-18. These are comparison controls; the main results use mixup-trained teachers.

Dataset	Architecture	Epochs	Clean
CIFAR-10	ResNet-18	200	95.32
CIFAR-100	ResNet-18	200	77.66
Tiny-ImageNet	ResNet-18	200	66.29

G Experimental Details

Our measured CFA results use the final epoch rather than a checkpoint selected on a robustness metric. Published methods may use different checkpoint-selection rules, so their reported results are not directly comparable on that dimension.

Teacher. The main experiments use the mixup-trained clean self-teacher selected in Section 3.3; its training uses natural images only. The student is warm-started from that teacher, and its classifier is then frozen. For the CIFAR-100 teacher comparison, mixup uses $\alpha = 1.0$. The standard-training teacher checkpoints in Table 15 are comparison controls, not the selected main-experiment teachers. They use ResNet-18, SGD at learning rate 0.1 under a one-cycle schedule, batch size 128, standard crop and flip, and 200 epochs. Figure 3 varies a standard teacher's training duration.

Student. The main ResNet-18 configurations are listed below. Unless noted otherwise, CIFAR-100 ablations use 100 epochs at $8.8/255$, with a 10-epoch AWP warmup.

	CIFAR-10	CIFAR-100	Tiny-ImageNet-200
Backbone	ResNet-18	ResNet-18	ResNet-18
Optimizer	AdamW	AdamW	AdamW
Learning rate / schedule	0.021, one-cycle	0.021, one-cycle	0.021, one-cycle
Epochs	150	150	100
Batch size	128	128	128
Training radius ϵ_{train}	$8/255$	$8/255$	$8.8/255$
Inner attack	PGD-10, step $2/255$	PGD-10, step $2/255$	PGD-10, step $2/255$
Allocation exponent p	1	1	1
Radius box $[w_{\text{lo}}, w_{\text{hi}}]$	$[0.5, 1.5]$	$[0.5, 1.5]$	$[0.5, 1.5]$
Classifier head	frozen	frozen	frozen
Weight averaging (start, decay)	0.2, 0.999	0.2, 0.999	0.2, 0.999
AWP (γ, warmup)	0.005, 15 ep	0.005, 15 ep	0.005, 10 ep
Teacher checkpoint	mixup, clean, 200 ep	mixup, clean, 200 ep	mixup, clean, 200 ep

We keep the inherited classifier fixed and omit the head-KD term, removing its temperature τ and weight β from the objective. p is not tuned. The training-radius sweep is reported in Table 13; evaluation is at $8/255$ in every cell.

Evaluation. We report AutoAttack accuracy at ℓ_∞ radius $8/255$ on the full test set; NRR is its harmonic mean with clean accuracy. Additional attacks are in Appendix J. Ported baselines retain their published recipes unless stated otherwise.

Repeated training. We repeat both arms of the final radius-allocation comparison with three student seeds. Uniform radii give 62.63 ± 0.22 clean and 28.01 ± 0.20 AA; sensitivity matching gives 64.82 ± 0.07 clean and 28.04 ± 0.09 AA (mean $\pm$ sample standard deviation). The paired clean gains are 2.35, 2.24, and 1.97 points, whereas the paired AA changes are 0.14, -0.23, and 0.18. Thus the clean gain repeats, while the data support no AA change. Other rows remain individual runs, so small differences between regression targets or nearby operating points should not be interpreted as significant.

Compute, teacher included. A single RTX 5080. One student cell is about 2.2 hours on CIFAR-100 at 100 epochs, of which the AutoAttack evaluation is roughly 7 minutes; Tiny-ImageNet is longer in proportion to its test set.

The teacher has to be counted, since calling it free would otherwise hide a training run. It is not free, but it is cheap: a clean epoch has no inner maximization, so it costs 6.8 seconds against 74.5 for an adversarial one, and 200 clean epochs come to 0.39 hours. The teacher is therefore 19% of one student run and is amortized over every cell that reads it, the teacher ladder of Figure 3 included. Table 16 puts the total against the baselines measured on the same machine.

Table 16: Training time on CIFAR-100 with ResNet-18 using one RTX 5080. Evaluation time is excluded; adversarial teacher costs for distillation baselines are not included.

Method	Epochs	Teacher (h)	Student (h)	Total (h)
Natural teacher (ours, reused)	200 clean	–	0.39	0.39
CFA (ours)	100 adv.	0.39	2.07	2.46
Label CE at our regime	100 adv.	0.39	3.22	3.61
PGD-AT, initialized at the teacher	100 adv.	0.39	1.72	2.11
ARD	100 adv.	0.39	1.52	1.91
RSLAD	100 adv.	0.39	1.71	2.10
AdaAD	100 adv.	0.39	2.79	3.18
LBGAT	100 adv.	joint	2.00	2.00
HAT	50 adv.	0.39	1.10	1.49
ADR	200 adv.	self	3.52	3.52

Two readings follow. The anchor is not bought with compute, since it costs less end to end than label cross-entropy at the same regime and about what PGD-AT costs once the teacher is included. The adversarial-distillation rows additionally rely on robust teachers with different training requirements, so their total teacher–student cost is not directly comparable to CFA's natural self-teacher setup.

H Additional Teacher-Checkpoint Results

Figure 3 shows one natural-training trajectory; Table 17 reports an independently trained trajectory. Each listed teacher was trained to completion under its own one-cycle schedule, then used to initialize a 50-epoch student with the same training procedure. These experiments vary the teacher checkpoint, not just the target features.

Table 17: Independent clean-teacher checkpoint comparison on CIFAR-100 with ResNet-18. Teacher seed 1 is transferred to 50-epoch students with seed 0.

	Teacher		Student		
Epoch	Clean	S_w/S_b	Clean	AA	NRR
20	73.87	1.118	63.82	23.22	34.05
40	75.57	1.112	64.36	23.86	34.81
100	77.27	0.990	63.68	25.61	36.53
150	77.68	0.888	63.37	25.49	36.36
200	77.90	0.817	63.02	26.16	**36.97**
300	78.27	0.708	62.34	25.94	36.64

The second trajectory again moves from higher student clean accuracy at earlier checkpoints to higher AA at later ones. The strongest AA occurs at 200 rather than 300 epochs, although the difference is small. The teacher's scatter ratio again orders student clean accuracy: its rank correlation with student clean accuracy is 0.94 on these six checkpoints, against 0.83 on the same grid of the first trajectory, and the teachers of Table 1, trained with different schemes rather than for different lengths, follow the same order. This is an association on one dataset and architecture, not a validated teacher-only selection rule.

I THE RECIPE CONTROLS

Unless explicitly ablated, CFA warm-starts from the natural teacher and trains with AdamW at 0.021 under a one-cycle schedule, while the baselines retain their authors' recipes. Table 18 asks whether that recipe is what wins: it runs our own objective, and label cross-entropy for reference, under both recipes and both initializations. Every row is measured here at the base regime, which carries no weight averaging, no AWP and a uniform radius, on CIFAR-100 with ResNet-18 for 100 epochs.

Table 18: Anchor and label-training objectives under two optimizer schedules on CIFAR-100 with ResNet-18 (100 epochs). Each pair holds the training radius and other settings fixed.

Objective	Recipe	Clean	AA	Avg	NRR
Anchor	ours, teacher init	61.29	25.36	**43.33**	**35.88**
Anchor	ours, random init	60.81	25.21	43.01	35.64
Anchor	theirs, teacher init	59.65	25.54	42.60	35.77
Label cross-entropy	ours, teacher init	54.60	19.45	37.03	28.68
Label cross-entropy	theirs, teacher init	56.66	20.91	38.79	30.55

Changing initialization or schedule changes the anchor's NRR by 0.22 and 0.11, respectively. Cross-entropy improves by 1.87 NRR under its own schedule, and that stronger baseline is the one carried into the comparisons in the main text. The anchor remains ahead under both tested schedules. These comparisons support an objective-level contribution within the tested recipes, without establishing significance from single runs or claiming exhaustive tuning of the alternatives.

J GRADIENT MASKING

We test for gradient masking with AutoAttack, independent attacks, and PGD sweeps (Athalye et al., 2018). Table 19 reports these checks for a 100-epoch CIFAR-100 repeat with 62.00% clean accuracy.

Table 19: Attack checks on all 10,000 CIFAR-100 test images at $\epsilon = 8/255$. Individual attacks are evaluated independently.

Check	Setting	Accuracy
white box against black box		
APGD-CE	white box, untargeted	31.80
APGD-T	white box, targeted	28.83
FAB-T	white box, boundary	29.16
Square	black box, 5000 queries	34.82
AutoAttack	standard evaluation	28.73
radius sweep, PGD-20		
$\epsilon = 2/255$		54.62
$\epsilon = 4/255$		47.03
$\epsilon = 8/255$		32.46
$\epsilon = 12/255$		20.88
$\epsilon = 16/255$		11.94
$\epsilon = 32/255$		1.05
$\epsilon = 64/255$		0.00
step sweep, $\epsilon = 8/255$		
PGD-10 / 20 / 50 / 100		33.17 / 32.57 / 32.30 / 32.24
transfer		
Natural teacher $\rightarrow$ student	PGD-20 built on the teacher	60.42

AutoAttack yields 28.73% accuracy. PGD accuracy falls monotonically with attack radius, reaching zero at $64/255$, and decreases with additional steps. These checks show no indication of gradient masking under the evaluated attacks.

The transfer row is worth reading separately. The student inherits the teacher's classifier and is initialized at the teacher's features, so adversarial examples built on the teacher might be expected

to carry over; they cost the student 1.58 points against its clean accuracy. Sharing a head is therefore not sharing a vulnerability, which is the same conclusion Section 5.3 reaches from class separation: what the anchored student changes is the position of its features, and the teacher's own adversarial directions do not point at the student's boundaries.

K Notes on the Baselines

Several rows of Table 2 run a specific variant of the method they name, or are grouped in a way that needs saying. This section is those decisions; the table carries none of them.

ARREST uses its published objective: adversarial finetuning from the natural model with SGD at 0.025, the angular representation term at $\lambda = 50$, and Noisy Replay over the first half at $\phi = 30°$. Table 2 reports its 150-epoch run for the main comparison, whereas Table 10 uses its published 20-epoch schedule. We read its representation after global pooling, where our anchor and geometry measurements are defined.

LBGAT trains its natural branch jointly with the robust one, so unlike the distillation rows its teacher is not obtained for free, and we group it accordingly. The two datasets run different variants because the authors publish different ones: CIFAR-100 is LBGAT6 and CIFAR-10 is LBGAT0, their release lists CIFAR-10 checkpoints under LBGAT0 alone, and $\beta = 6$ is their CIFAR-100 setting. We report the configuration each dataset has a published counterpart for rather than forcing one β across both.

The released LBGAT schedule was corrected in our CIFAR runs after an epoch-indexing error. The CIFAR-100 clean/AA result changed from $56.46/26.02$ to $57.29/26.22$; CIFAR-10 then trained normally. The Tiny-ImageNet and WideResNet results were measured before that correction, so those LBGAT cells use a slightly different schedule from the CIFAR ResNet-18 cells.

HAT uses a naturally trained network only to *label* helper points beyond the ball, never as a target, so it belongs with the trade-off methods and not the distillation ones. It is run on its own published schedule, because our one-cycle schedule collapses it to chance accuracy on both datasets — the same allowance we make for LBGAT. On CIFAR-10 the port reproduces HAT's published no-extra-data figure to within 0.31 clean and 0.27 AutoAttack ($84.90/49.08$ reported, ResNet-18), which is what licenses reporting its CIFAR-100 row at all: HAT publishes no CIFAR-100 result without additional data, its $61.50/28.88$ being trained with DDPM-generated samples, so that column has no published figure to be read against.

ADR reports two configurations — AT + ADR at $56.10/26.87$ on CIFAR-100 and AT + WA + AWP + ADR at $57.36/28.50$ — of which the second is the one component-matched to our recipe, since it carries both weight averaging and AWP. ADR is also self-distillation: its teacher is an exponential moving average of the student, no external network is read, and the student trains from scratch, so its row inherits nothing of our natural teacher.

Teacher-only KL with the training stack. This diagnostic combines probability-target distillation with WA and AWP; it is not a reproduction of a published method. Temperature is applied only to the teacher, and the best tested stacked setting, $\tau = 4$, gives $56.97/26.96$ clean/AA; $\tau = 16$ gives $42.45/20.10$. The comparison concerns this KL training design, not all logit-based supervision. In particular, fixed-head logit MSE is evaluated separately in Table 11, where it remains competitive with feature regression.

L AWP Implementation Check

AWP uses an auxiliary ascent objective before the backbone update. We corrected an implementation mismatch: the ascent closure retained a head-distillation term that the descent objective omitted, and a detach affected the wrong feature path. Re-running the reported recipe with matching closures gives $62.17/28.86$ clean/AutoAttack on CIFAR-100 and $84.96/51.74$ on CIFAR-10, compared with

$62.65/28.77$ and $85.58/51.79$ before correction. The main results use the corrected implementation. In the training-radius sweep of Table 13, the $8.8/255$ rows are these corrected runs, whereas the other radii predate the correction. At $8.8/255$ the correction changed clean accuracy by -0.48 (CIFAR-100) and -0.62 (CIFAR-10) and AutoAttack by $+0.09$ and -0.05, so differences between radii smaller than this should not be read from that table.

M Limitations and Discussion

The teacher comparisons show that clean accuracy alone is an incomplete measure of transferability. We select the mixup-trained teacher based on observed transfer, while the measured associations with feature geometry provide directions for further study. A reliable selection criterion that can be evaluated before training the student would make this approach more practical. The competitive clean-logit regression results also suggest that the value of a fixed clean reference extends beyond the particular feature metric used by CFA.

The analysis characterizes the quantities controlled by anchoring and illustrates how a student can avoid a teacher's fragile features in a restricted model. Connecting these results quantitatively to the behavior of trained image classifiers remains open. Similarly, sensitivity matching uses a local gradient estimate to allocate the perturbation budget, and its benefit is established empirically. Studying how this estimate evolves along the adversarial trajectory may help refine the allocation while retaining its simple implementation.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Customized adversarial training for improved robustness. In *IJCAI*, 2022.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *ICCV*, 2021.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3996–4003, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Bo Huang, Mingyang Chen, Yi Wang, Junda Lu, Minhao Cheng, and Wei Wang. Boosting accuracy and robustness of student models via adaptive adversarial distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24668–24677, 2023.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. CS231n course project report, Stanford University, 2015.

Hongsin Lee, Seungju Cho, and Changick Kim. Indirect gradient matching for adversarial robust distillation. In *ICLR*, 2025.

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.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019.

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

Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*, 2022.

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

Satoshi Suzuki, Shin'ya Yamaguchi, Shoichiro Takeda, Sekitoshi Kanai, Naoki Makishima, Atsushi Ando, and Ryo Masumura. ARREST: Adversarial finetuning with latent representation constraint to mitigate the accuracy-robustness tradeoff. In *ICCV*, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jihoon Tack, Sihyun Yu, Jongheon Jeong, Minseon Kim, Sung Ju Hwang, and Jinwoo Shin. Consistency regularization for adversarial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8414–8422, 2022.

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

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Yu-Yu Wu, Hung-Jui Wang, and Shang-Tse Chen. Annealing self-distillation rectification improves adversarial training. In *ICLR*, 2024.

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.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Shiji Zhao, Xizhe Wang, and Xingxing Wei. Mitigating accuracy-robustness trade-off via balanced multi-teacher adversarial distillation. *IEEE TPAMI*, 2024.

Jianing Zhu, Jiangchao Yao, Bo Han, Jingfeng Zhang, Tongliang Liu, Gang Niu, Jingren Zhou, Jianliang Xu, and Hongxia Yang. Reliable adversarial distillation with unreliable teachers. *arXiv preprint arXiv:2106.04928*, 2021.

Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16443–16452, 2021.