REPALIGN: A NEW REGULARIZER FOR ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks are known to be highly vulnerable to adversarial examples, making robust training a practical necessity. However, robust training often leads to severe accuracy-robustness trade-offs. Existing methods like TRADES and Adversarial Logit Pairing (ALP) align softmax or logit outputs of adversarial inputs with those of clean samples, but are limited to layers at the end of the network. We propose RepAlign, a framework which uses representation similarity (RS) metrics such as Centered Kernel Alignment (CKA) to optimize for alignment between arbitrary features of clean and adversarially perturbed data. The use of general RS metrics allows RepAlign to align features having different properties. Evaluations across CIFAR-10, CIFAR-100, and Imagenette demonstrate that RepAlign consistently pushes the accuracy-robustness Pareto frontier forward. Soft-kNN on CIFAR-10 achieves 86.90% clean accuracy (a $+4.05\%$ gain over TRADES) while maintaining 49.37% robust accuracy ($+3.06\%$ over PGD-AT). On CIFAR-100, Soft-kNN strictly outperforms TRADES on both fronts, achieving 60.42% clean and 26.10% robust accuracy (gains of $+4.63\%$ and $+0.77\%$ over TRADES, respectively). On Imagenette, CKNNA closes the gap to clean training (91.82%) more than any other defense, reaching 88.46% clean accuracy ($+1.81\%$ over PGD-AT and $+3.77\%$ over TRADES) while sustaining 61.89% robust accuracy. CKA heatmaps show that RepAlign mitigates representation drift and restores the layer specializaion disrupted by standard adversarial training.

1 INTRODUCTION

The last two decades of research in machine learning (ML) have established the weakness of neural networks and many other types of ML models to adversarial perturbations of input data (Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2017). The resultant samples called adversarial examples are input samples that are imperceptibly close to benign samples that the classifier correctly predicts with high confidence, yet are themselves misclassified with high confidence. A vast array of various adversarial attacks with variation in optimization formulation and constraints have been developed (§2.1). This has led to a continuous attack-defense arms race where many proposed defenses have been found to be vulnerable to adaptive attacks (Athalye et al., 2018) . Benchmarks like RobustBench (Croce et al., 2021) now exist which rigorously evaluate defenses against diverse white-box and black-box threat models.

To defend against such attacks, adversarial training (AT) (Madry et al., 2018) and its variants Wang et al. (2020) have been the only primary effective defense (§2.2). While provable defenses exist, (Cohen et al., 2019) they are often restricted to impractically small perturbation radii and significantly reduce accuracy on clean samples ('benign accuracy'). The tradeoff between benign and robust accuracy is well known (Tsipras et al., 2019; Su et al., 2018). Approaches to mitigating this trade-off include incorporating both benign and adversarial data during training (Kurakin et al., 2017; Zhang et al., 2019), and regularizing prediction differences via Adversarial Logit Pairing (ALP) (Kannan et al., 2018) and TRADES (Zhang et al., 2019). TRADES penalizes the KL-divergence between softmax output distributions and is the current baseline for balancing clean and robust accuracy without adding any synthetic data in training (Gowal et al., 2021a; Pang et al., 2021).

However, TRADES only matches the model's final outputs, ignoring the features learned in earlier layers. Given the emergence of representation similarity (RS) metrics that enable quantitative com-

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.

parisons of arbitrary features Raghu et al. (2017); Kornblith et al. (2019); Huh et al. (2024), a natural question arises:

Can representation similarity metrics be used instead of KL-divergence to achieve a better clean-robust accuracy trade-offs?

In this paper, we answer this by proposing RepAlign (§3), a framework which regularizes adversarial training using RS metrics. Instead of restricting alignment to the final softmax distribution via KL-divergence as in TRADES, RepAlign directly quantifies and maximizes the similarity between the intermediate latent representations of benign and adversarial inputs. We adapt Centered Kernel Alignment (CKA), Centered Kernel Nearest Neighbor Alignment (CKNNA), the Hilbert-Schmidt Independence Criterion (HSIC), and Soft k-Nearest Neighbors (Soft-kNN) (Huh et al., 2024) to penalize divergence in feature representations between benign and perturbed inputs during training.

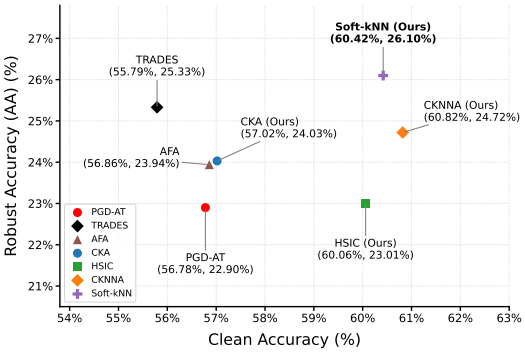


Figure 1: **Clean vs. Robust accuracy Pareto frontier on CIFAR-100.**

We evaluate RepAlign across three benchmark datasets, Imagenette (Howard, 2019), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) against standard attacks like the Autoattack (AA) suite (APGD-CE, APGD-DLR, FAB-T, Square) (Croce & Hein, 2020), ART-PGD (Nicolae et al., 2018), and the gradient-free SPSA attack (Uesato et al., 2018) [1]. Our key findings are:

1. **Improved clean-robust Pareto frontier:** Across all benchmarks, RepAlign consistently pushes the Pareto frontier over standard PGD-AT (Madry et al., 2018), TRADES(Zhang et al., 2019) and adversarial feature alignment (AFA) (Park et al., 2024):
   - On **CIFAR-10** (Figure 1), Soft-kNN reaches $\mathbf{86.90}\%$ clean accuracy ($+4.05\%$ over TRADES, $+3.75\%$ over PGD-AT) while maintaining strong AutoAttack robustness (49.37%) and higher black-box accuracy on SPSA ($\mathbf{70.19}\%$ vs. 67.63%).
   - On **CIFAR-100**, Soft-kNN achieves $\mathbf{60.42}\%$ clean accuracy ($+4.63\%$ over TRADES) while beating TRADES on both AutoAttack ($\mathbf{26.10}\%$ vs. 25.33%) and SPSA ($\mathbf{42.97}\%$ vs. 40.39%).
   - On **Imagenette**, RepAlign recovers up to $+3.77\%$ over TRADES in clean accuracy (reaching $\mathbf{88.46}\%$) while outperforming it on both AutoAttack (62.50% vs. 61.66%) and SPSA ($\mathbf{84.33}\%$ vs. 80.36%).
2. **Feature representation analysis:** Using CKA to analyse the learned features after robust training shows that RepAlign preserves the block-wise structure and layer specialization typical of clean models (Cianfarani et al., 2022). PGD-AT and TRADES, can degrade these representations while RepAlign avoids this collapse, providing evidence for how it recovers clean accuracy without sacrificing robustness.

2 Background and Notation

2.1 Adversarial Attacks

Let $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ be a classifier trained on some dataset D, with samples (x, y) drawn from an underlying distribution $\mathcal{D}$. We typically assume that $x \in \mathbb{R}^d$ and $y \in [C]$ This classifier is parameterized by θ and trained using a loss function $\mathcal{L}(f_\theta(x), y)$.

Although deep neural networks achieve high accuracy on benign data, they are vulnerable to test-time adversarial inputs (Szegedy et al., 2014; Carlini & Wagner, 2017). The Fast Gradient Sign

[1] Code is available at: `https://anonymous.4open.science/r/RepAlign`

Method (FGSM) (Goodfellow et al., 2015) generates perturbations in a single gradient step. The Projected Gradient Descent (PGD) attack (Madry et al., 2018; Kurakin et al., 2017) is perhaps the most commonly used attack that maximizes the loss function using multiple gradient steps given step size α and the set of allowed perturbations within a ball of radius ϵ around x:

$$x^{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}\big(x^t + \alpha \operatorname{sign}(\nabla_{x^t}\mathcal{L}(f_\theta(x^t), y))\big) \tag{1}$$

This iterative process is more powerful than FGSM, but is slower as multiple ascent steps are needed. Furthermore, studies have shown that PGD can still fail to yield optimal results, which is often attributed to its use of a fixed step size (Mosbach et al., 2019), inability to capture the optimization trend (Croce et al., 2020) and its reliance on cross-entropy loss formulation. These factors can cause the loss to plateau after a few iterations, meaning the attack fails to fully exploit the given perturbation budget (Maini et al., 2020; Mosbach et al., 2019; Croce et al., 2020).

AutoAttack (Croce & Hein, 2020) addresses these issues without requiring manual parameter tuning by dynamically adapting step sizes and using the Difference-of-Logits-Ratio (DLR) loss to avoid gradient vanishing. While computationally heavy, it reliably evaluates robustness using an ensemble of four attacks: APGD-CE, targeted APGD-DLR, targeted FAB-T, and the black-box Square Attack.

2.2 Adversarial Defenses

Defenses against adversarial attacks aim to train models that are robust to small input perturbations. While various methods were proposed to improve robustness (Papernot et al., 2016; Xie et al., 2019; Carmon et al., 2019; Zhang et al., 2020), many of these proved ineffective due to their reliance on obfuscated gradients (Athalye et al., 2018) or gradient masking (Papernot et al., 2017). Standard Adversarial training (PGD-AT) (Madry et al., 2018; Pang et al., 2021) remains the benchmark empirical defense, formulating robustness as a saddle point problem:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\max_{x'\in\mathcal{B}(x,\epsilon)} \mathcal{L}(f_\theta(x'), y)\big] \tag{2}$$

where $x' \in \mathcal{B}(x,\epsilon)$ is the worst-case adversarial perturbation found within the ϵ-ball around the benign input x. The inner maximization is often approximated using methods mentioned in §2.1. Single-step methods like FGSM are faster but have poor generalization and suffer from "catastrophic overfitting" (Wong et al., 2020), which is explained and remedied by zeroing out the gradients (Golgooni et al., 2021).

To address the degradation in clean accuracy caused by standard AT, regularized training approaches incorporate both clean and adversarial inputs. Early efforts like Adversarial Logit Pairing (ALP) (Kannan et al., 2018) penalize the ℓ_2 distance between clean and adversarial logits. The TRADES framework (Zhang et al., 2019) provides a principled information-theoretic alternative that balances benign and robust accuracy:

$$\min_\theta \ \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathcal{L}(f_\theta(x), y) + \beta \cdot \max_{x'\in\mathcal{B}(x,\epsilon)} \underbrace{D_{\mathrm{KL}}\big(f_\theta(x') \,\|\, f_\theta(x)\big)}_{\mathcal{L}_{\text{div}}}\Big] \tag{3}$$

where β is a regularizing hyperparameter and $\mathcal{L}_{\text{div}}$ represents the KL-divergence between the softmax outputs of clean and perturbed inputs. [2] Our work builds directly upon this foundation, but shifts the alignment objective $\mathcal{L}_{\text{div}}$ from the output probability simplex to arbitary intermediate feature space.

This line of work has since been extended in various directions, including the use of misclassified examples (Wang et al., 2020; Kurakin et al., 2017), early stopping to prevent robust overfitting (Rice et al., 2020), weight ensembling (Chen et al., 2021; Izmailov et al., 2018; Wang & Wang, 2022), data augmentation (Rebuffi et al., 2021), generative modeling (Gowal et al., 2021b; Sehwag et al., 2022), double-perturbation mechanisms that regularize the flatness of the loss landscape (Wu et al., 2020), and feature alignment using a contrastive loss (Park et al., 2024).

Representation-Level Robustness: Prior work shows that adversarial perturbations distort not only final outputs but also intermediate feature representations (Huang et al., 2019; Cianfarani et al.,

[2] While Zhang et al. (2019) formulate a general ϕ-divergence theoretically, their empirical implementation uses the KL-divergence.

2022). To protect hidden layers, several defenses have explored feature-space regularization, including feature denoising (Xie et al., 2019), penalizing large eigenvalues in activation covariance (Cheng et al., 2024), manifold smoothing (Jin & Rinard, 2020), and contrastive feature alignment (Park et al., 2024). However, these methods typically impose rigid point-wise constraints (such as ℓ_2 penalties) or force representations toward fixed class centers, which can disrupt natural feature specialization and degrade clean accuracy. In contrast, RS metrics compare the relational geometry between samples rather than absolute coordinates. Being invariant to orthogonal transformations and scaling, RS metrics can provide a principled way to align clean and adversarial representations without distorting the underlying feature space.

2.3 Representation Similarity Metrics

Quantifying the similarity between representations is important for understanding deep neural net models. Metrics such as Euclidean (ℓ_2) distance are not invariant to invertible linear transformations, orthogonal transformation or isotropic scaling, while metrics such as cosine similarity are instance specific. Representation Similarity (RS) metrics provide one or more of these properties by comparing the relational geometry of a batch of samples rather than individual activations.

HSIC: The Hilbert–Schmidt Independence Criterion (Gretton et al., 2007) is a kernel-based method to measure the dependence between two representations. Let $\phi_i, \psi_i \in \mathbb{R}^p$ denote the vectorized feature representations of benign and adversarial inputs, respectively. Let $K, L \in \mathbb{R}^{n \times n}$ be the corresponding kernel matrices with entries $K_{ij} = \kappa(\phi_i, \phi_j)$ and $L_{ij} = \kappa(\psi_i, \psi_j)$. Given the centering matrix $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, the empirical estimator for HSIC is:

$$\text{HSIC}(K, L) = \frac{1}{(n-1)^2}\text{tr}(KHLH) \tag{4}$$

We note that in reality κ can represent any positive semi-definite kernel, with the Radial Basis Function (RBF) and linear being the popular choices. Both perform comparably for representation analysis (Kornblith et al., 2019; Huh et al., 2024), we use linear biased kernels ($K = XX^\top, L = X_{\text{adv}}X_{\text{adv}}^\top$) for differentiability and computational efficiency. Additionally, a key limitation of raw HSIC is that it is unbounded and scales with the norm of the representations, making it sensitive to arbitrary feature scaling.

CKA: Centered Kernel Alignment (Kornblith et al., 2019) normalizes HSIC to derive a bounded value invariant to isotropic scaling:

$$\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K)\,\text{HSIC}(L, L)}} \tag{5}$$

CKNNA: Centered Kernel Nearest Neighbor Alignment (Huh et al., 2024) is a variant of CKA that focuses on local structure. CKNNA modifies the definition of alignment to restrict the measurement to samples considered to be nearest neighbors of the current sample i.

$$\text{Align}_{\text{knn}}(K, L) = \sum_i \sum_j \alpha(i, j) \cdot (\langle \phi_i, \phi_j \rangle - \mathbb{E}_l[\langle \phi_i, \phi_l \rangle]) \times (\langle \psi_i, \psi_j \rangle - \mathbb{E}_l[\langle \psi_i, \psi_l \rangle]) \tag{6}$$

where $\alpha(i, j) = \mathbf{1}\big[\phi_j \in \text{knn}(\phi_i) \wedge \psi_j \in \text{knn}(\psi_i) \wedge i \neq j\big]$.

Here $\alpha(i, j)$ is a scalar weighting that assigns 1 if j is a mutual nearest neighbor to both ϕ_i and ψ_i, and 0 otherwise. We obtain CKNNA as:

$$\text{CKNNA}(K, L) = \frac{\text{Align}_{\text{knn}}(K, L)}{\sqrt{\text{Align}_{\text{knn}}(K, K)\,\text{Align}_{\text{knn}}(L, L)}} \tag{7}$$

SoftKNN: Mutual-kNN is a stricter measure for local representation similarity where only points which belong to neighborhoods of each other are considered. This uses a sort operation and is thus non-differentiable. The Soft Nearest Neighbor Loss (Salakhutdinov & Hinton, 2007; Frosst et al., 2019) provides a differentiable alternative. We formulate a differentiable Soft k-Nearest Neighbor

(Soft-kNN) alignment objective for training. This provides continuous, temperature-scaled similarities between representations. Given feature representations ϕ_i and ψ_i, we construct the soft-neighborhood matrices K and L. For K, the entries are computed as:

$$K_{ij} = \begin{cases} \frac{\exp\left(\phi_i^\top \phi_j / \tau\right)}{\sum_{l \neq i} \exp\left(\phi_i^\top \phi_l / \tau\right)} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The matrix L is constructed similarly. The overall alignment metric is then defined as $\mathcal{S} = \frac{1}{n}\mathrm{Tr}(KL^\top)$.

3 Robust Training via Representation Alignment

3.1 The RepAlign Objective

Our central idea is to train robust models by explicitly penalizing representation drift between benign and adversarial examples. Let $f_\theta(x)$ denote the classifier predictions and $g_\theta(x) \in \mathbb{R}^p$ denote the intermediate feature representations extracted from a chosen layer of the network. For a benign input x with ground truth label y and its corresponding adversarial input x', the proposed **RepAlign** training objective is:

$$\mathcal{L}_{\text{RA}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathcal{L}_{\text{CLF}}(f_\theta(x'), y) - \lambda \cdot \mathcal{L}_{\text{RS}}\big(g_\theta(x), g_\theta(x')\big)\Big] \tag{9}$$

Here, $\mathcal{L}_{\text{CLF}}$ is the robust classification loss (typically cross-entropy loss) evaluated on adversarial sample x', which makes sure that the model maintains prediction accuracy even after perturbation. The second term is the proposed **RepAlign** regularizer, where $\mathcal{L}_{\text{RS}} \in [0, 1]$ is a representation similarity metric (such as CKA, CKNNA, HSIC, SoftKNN as mentioned in §2.3), and $\lambda > 0$ is a hyperparameter to balance robust classification and representation alignment. The $\mathcal{L}_{\text{RS}}$ regularizer has a negative sign because it measures similarity so minimizing $\mathcal{L}_{\text{RA}}$ simultaneously minimizes classification error on perturbed inputs while maximizing feature alignment between clean $g_\theta(x)$ and adversarial $g_\theta(x')$ representations.

3.2 Training Algorithm

To solve Equation (9), we alternate between generating adversarial perturbations in the inner loop and updating model parameters θ in outer loop. For an input x, adversarial perturbation x' is generated using multi-step PGD. The gradient ascent update is:

$$x'_{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}\Big(x'_t + \alpha \,\mathrm{sign}\left(\nabla_{x'_t}\mathcal{L}_{\text{CLF}}(f_\theta(x'_t), y)\right)\Big) \tag{10}$$

where Π is the projection operator onto $\mathcal{B}(x, \epsilon)$ and α is the step size.

Design Choice for the Inner Maximization. We optimize the inner PGD attack on $\mathcal{L}_{\text{CLF}}$ instead of the full objective[3] for two reasons: i) realistic threat model: Test-time adversaries aim to induce label misclassification using $\mathcal{L}_{\text{CLF}}$, which directly targets the classification boundary and are not concerned with moving arbitary representations apart; ii) computational efficiency: Differentiating the $O(b^2)$-sized batch similarity matrices through multiple inner PGD steps is computationally expensive. Decoupling the inner attack keeps training complexity throughput comparable to standard PGD-AT. The complete training procedure is detailed in Algorithm 1. **Choice of Representation Layer:** A key design choice is the layer from which to extract the representations $g_\theta(x)$. Different layers capture features at different levels of abstraction: i)final layer (pre-logit): Applying the regularizer to the features just before the final classification layer is the most direct analogue to TRADES. It encourages stability in the space that directly feeds into the classifier while preserving the global geometry learned across the entire model; ii) earlier layers: These learn generic features like edges and textures, while middle layers learn more complex shapes and parts. Enforcing stability in these layers could prevent the model from relying on non-robust, high-frequency features that are easily

[3]Note the difference from TRADES here, where the KL-objective can be directly maximised against to cause misclassification.

Algorithm 1 Adversarial Training with Representation Alignment (RepAlign)

Require: Model f_θ with feature extractor g_θ, data distribution $\mathcal{D}$, trade-off parameter λ, perturbation budget ϵ, PGD steps K, step size α, learning rate η.
for each training batch $\{x_i, y_i\}_{i=1}^N \sim \mathcal{D}$ **do**
  Extract benign representations $\Phi_x = g_\theta(x)$.
  Initialize adversarial examples $x'_0 \leftarrow x + \mathcal{U}(-\epsilon, \epsilon)$ and clip to data bounds.
  for $t = 0, \ldots, K-1$ **do**
    Update adversarial examples $x'_{t+1} \leftarrow \Pi_{\mathcal{B}(x,\epsilon)}\Big(x'_t + \alpha \operatorname{sign}\left(\nabla_{x'_t} \mathcal{L}_{\text{CLF}}(f_\theta(x'_t), y)\right)\Big)$.
  end for
  Compute adversarial representations $\Phi_{x'} = g_\theta(x'_K)$ and logits $f_\theta(x'_K)$.
  Compute robust classification loss $\mathcal{L}_{\text{CLF}} = \frac{1}{N}\sum_i \mathcal{L}_{\text{CLF}}(f_\theta(x'_{K,i}), y_i)$.
  Compute alignment similarity $\mathcal{L}_{\text{RS}} = \mathcal{L}_{\text{RS}}(\Phi_x, \Phi_{x'})$.
  Compute total loss $\mathcal{L}_{\text{RA}} = \mathcal{L}_{\text{CLF}} - \lambda \cdot \mathcal{L}_{\text{RS}}$.
  Update model parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{RA}}$.
end for

manipulated by an adversary. Empirically we find that applying regularization on **pre-logits layer** gives the the best clean-robust Pareto frontier across all benchmarks. Constraining on earlier intermediate blocks seems to restrict feature flexibility and degrades clean accuracy. Comprehensive ablation comparisons with middle-layer regularization are reported in §C.3.

4 Experimental Setup

Datasets and Architectures: We conduct our experiments across three standard vision benchmark datasets: CIFAR-10 (Krizhevsky, 2009) (50,000 training and 10,000 test images distributed across 10 classes), CIFAR-100 (Krizhevsky, 2009) (50,000 training and 10,000 test images distributed across 100 classes) and Imagenette (Howard, 2019) (a high-resolution (320 px) 10-class subset of ImageNet comprising 9,469 training and 3,925 validation images). Consistent with prior work on adversarial robustness, we use the WideResNet-28-10 architecture for CIFAR-10, and CIFAR-100. For the higher-resolution Imagenette dataset, we instead adopt a standard ResNet-50, as scaling a WideResNet-28-10 to 320px inputs is computationally expensive and incurs massive memory costs.

Training Details and Computational Resources: All models are trained for 200 epochs using Stochastic Gradient Descent (SGD) (Bottou, 2012) with a Nesterov momentum of 0.9 and a weight decay of 5×10^{-4}. The learning rate is initialized at 0.1 and decayed by a factor of 10 at epochs 100 and 150. All adversarial training methods use both 10- and 20-step PGD inner maximization (PGD_{20}) bounded by an ℓ_∞ constraint of $\epsilon = 8/255$ with a step size fixed to $\alpha = 2/255$. All experiments were conducted using NVIDIA RTX PRO 6000 Blackwell Max-Q GPU (96 GB VRAM). Adversarial training requires approximately 23-25 hours for CIFAR-10/100, and 10-11 hours for Imagenette. RepAlign methods require almost the same training time as standard PGD-AT (sometimes even less time) while running $\sim 25\%$–30% faster than TRADES (see §F for detailed run-times).

Baselines and Initialization Strategy: We benchmark our proposed alignment regularizers against four standard training paradigms: i) **Clean Training:** Models trained purely on benign data using standard cross-entropy loss; ii) **PGD-AT:** Standard min-max adversarial training (Madry et al., 2018); iii) **TRADES:** Regularized adversarial training (Zhang et al., 2019) with the trade-off hyperparameter fixed to $\beta = 6$ based on standard literature (and our finding), and iv) **AFA:** Adversarial Feature Alignment (Park et al., 2024), a recent contrastive feature-level defense (using their Track B joint training with pre-trained initialization).

To thoroughly evaluate the impact of initialization, we trained the PGD-AT and TRADES baselines using two distinct strategies: *i) Training from scratch* (random initialization), and *ii) Pretrained initialization* (loading the weights from a pretrained clean checkpoint trained for 200 epochs). Both baselines were trained utilizing both 10-step and 20-step PGD inner maximizations.

Proposed Methods: We evaluate four variants for $\mathcal{L}_{\text{RS}}$ in the RepAlign framework: i) **CKA:** Centered Kernel Alignment (Global alignment, bounded); ii) **HSIC:** Hilbert-Schmidt Independence Cri-

Table 1: Clean and Robust Accuracy (%) across CIFAR-10, CIFAR-100, and Imagenette under an ℓ_∞ budget of $\epsilon = 8/255$. Results reflect the best configuration for each method with standard deviations across three seeds where available. APGD-DLR and FAB-T denote the targeted variants used in AutoAttack. For AFA, we report Track B (joint training) with pretrained initialization. **Bold** and underlined values indicate the best and second-best results among defense models, respectively. Green values indicate the worst-case (minimum) robust accuracy across the AutoAttack suite components (APGD-CE, APGD-DLR, FAB-T, Square) for each method, representing the overall AutoAttack robustness.

			White-Box Attacks				Black-Box Attacks	
Dataset	**Method**	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CIFAR-10	Clean Training	96.17	0.00	0.00	3.59	0.02	1.04	0.25
	PGD-AT	83.15	49.18 ± 0.05	46.31 ± 0.01	47.21	49.87	54.94 ± 0.05	66.88 ± 0.01
	TRADES	82.85	**52.60 ± 0.03**	**50.01 ± 0.01**	**51.42**	**53.10**	56.61 ± 0.06	67.63 ± 0.06
	AFA	83.80	51.24	48.79	50.47	51.96	55.87	68.33
	CKA (Ours)	84.33	50.92 ± 0.03	48.43 ± 0.01	49.68	51.61	56.83 ± 0.03	68.20 ± 0.07
	CKNNA (Ours)	86.01	50.57 ± 0.01	48.70 ± 0.03	50.99	51.49	57.90 ± 0.06	69.71 ± 0.04
	HSIC (Ours)	85.92	51.57 ± 0.04	49.17	51.21	52.17	57.95 ± 0.06	69.63 ± 0.09
	Soft-kNN (Ours)	**86.90**	51.19 ± 0.04	49.37 ± 0.01	51.30	51.82	**58.19 ± 0.08**	**70.19 ± 0.02**
CIFAR-100	Clean Training	80.98	0.00	0.00	5.92	0.02	0.49	0.19
	PGD-AT	56.78	26.23 ± 0.02	22.90 ± 0.01	23.69	26.78	28.84 ± 0.06	39.87 ± 0.02
	TRADES	55.79	28.86 ± 0.06	25.33 ± 0.01	26.57	29.20	29.53 ± 0.09	40.39 ± 0.04
	AFA	56.86	27.28	23.94	25.80	27.68	28.40	39.95
	CKA (Ours)	57.02	27.62 ± 0.02	24.03 ± 0.01	24.76	28.30	29.85 ± 0.11	40.88 ± 0.05
	CKNNA (Ours)	**60.82**	27.47 ± 0.04	24.72 ± 0.01	26.50	28.11	30.17 ± 0.04	41.42 ± 0.04
	HSIC (Ours)	60.06	25.16 ± 0.03	23.01 ± 0.01	25.03 ± 0.03	25.72 ± 0.01	28.52 ± 0.07	39.36 ± 0.04
	Soft-kNN (Ours)	60.42	**29.41 ± 0.03**	**26.10 ± 0.01**	**27.77 ± 0.20**	**30.36 ± 0.02**	**31.94 ± 0.04**	**42.97 ± 0.04**
Imagenette	Clean Training	91.82	0.00	0.00	7.18	0.00	5.78	39.57
	PGD-AT	86.65	64.55 ± 0.04	**62.52 ± 0.04**	63.56 ± 0.05	64.87	72.13 ± 0.09	82.79 ± 0.03
	TRADES	84.69	**65.06 ± 0.01**	61.66	63.37 ± 0.15	**65.43**	70.28 ± 0.17	80.36 ± 0.07
	AFA	77.73	52.10	48.15	50.24	52.48	56.54	74.19
	CKA (Ours)	87.57	63.75 ± 0.01	61.91	63.51 ± 0.15	64.43	73.14 ± 0.04	83.52 ± 0.03
	CKNNA (Ours)	**88.46**	63.61 ± 0.01	61.89 ± 0.01	63.52 ± 0.33	64.48	74.01 ± 0.09	84.31
	HSIC (Ours)	87.75	63.37 ± 0.04	61.65 ± 0.01	63.57 ± 0.12	63.90	72.87 ± 0.08	83.92 ± 0.01
	Soft-kNN (Ours)	87.87	64.54 ± 0.01	62.50	**64.21 ± 0.06**	65.17	**74.11 ± 0.09**	**84.33 ± 0.04**

terion (Global alignment, unbounded); iii) **CKNNA:** k-Nearest Neighbor Alignment (Local alignment); iv) **Soft-kNN:** Soft k-Nearest Neighbor Alignment (Local soft-neighborhood alignment).

To isolate the benefit of representation alignment and ensure stable convergence, all four of our proposed methods are initialized from the pretrained clean checkpoints and trained for 200 epochs. We train variants using both 10-step and 20-step inner PGD maximizations, sweeping the alignment strength $\lambda \in \{0.1, 1, 5, 10\}$ across all three datasets (CIFAR-10, CIFAR-100, and Imagenette). For computational efficiency, online mini-batch estimation of RS (Nguyen et al., 2021) is used during training. Metrics such as HSIC and CKA require operations involving Gram matrices, scaling quadratically with dataset size ($\mathcal{O}(N^2)$). However, in our method we use batch-wise for backpropagation. This means we have an additional time complexity of $\mathcal{O}(b^2 d)$ that translates to an overall linear dependence on batch size and feature dimension $\mathcal{O}(Nbd)$ in training. For all primary results reported in Table 1, representations are extracted from the *pre-logits layer* (immediately preceding the final classifier). A comprehensive layer ablation comparing pre-logits with intermediate residual blocks across architectures is provided in §C.3.

Evaluation Protocol: After training, we freeze the network weights and conduct a rigorous, standalone evaluation of all checkpoints. To ensure our attacks do not suffer from gradient masking and provide true adversarial robustness, we evaluate the networks across both white-box and black-box threat models using an ℓ_∞ budget of $\epsilon = 8/255$ and step size $\alpha = 2/255$:

- **White-Box Attacks:** We evaluate against the white-box components of the AutoAttack suite (Croce & Hein, 2020)—specifically, untargeted Auto-PGD with cross-entropy loss (**APGD-CE**), targeted Auto-PGD with Difference-of-Logits-Ratio loss (**APGD-DLR**), and targeted Fast Adaptive Boundary attack (**FAB-T**)—alongside a standardized 20-step PGD attack implemented via the Adversarial Robustness Toolbox (**ART-PGD**) (Nicolae et al., 2018).
- **Black-Box Attacks:** To ensure defenses do not rely on obfuscated gradients, we evaluate against the query-based **Square Attack** (Andriushchenko et al., 2020) (the black-box component of AutoAttack) and the finite-difference gradient-free **SPSA** attack (Uesato et al., 2018).

5 RESULTS AND ANALYSIS

In this section, we evaluate the proposed RepAlign framework across standard vision benchmarks. We first analyze the classical robustness-accuracy trade-off, demonstrating that our methods push the Pareto frontier forward. We then analyze the internal latent representations of the networks, using layer-wise representation similarity heatmaps. Finally, we provide ablations on regularizer strength, neighborhood temperature, and layer depth.

5.1 ROBUSTNESS EVALUATION AND THE PARETO FRONTIER

Table 1 summarizes the primary results across CIFAR-10, CIFAR-100, and Imagenette under ℓ_∞ perturbations of budget $\epsilon = 8/255$. Standard PGD-AT suffers from a severe trade-off, losing significant clean accuracy. While TRADES mitigates some of these issues by smoothing the output logits, our *alignment-based methods provide a strictly superior defense advancing the clean-versus-robust Pareto frontier across all three datasets* (see §A for full Pareto curves). For an exhaustive breakdown of all baseline configurations including the impact of random versus pretrained initialization and 10-step versus 20-step inner maximization, please refer to §B.

CIFAR-10: Soft-kNN and CKNNA substantially outperform all baselines in clean accuracy, reaching 86.90% and 86.01% respectively (a $+4.05\%$ and $+3.16\%$ gain over TRADES). Simultaneously, they maintain highly competitive white-box robustness (51.19% and 50.57% APGD-CE) and achieve the strongest black-box robustness across the board (58.19% on Square Attack and 70.19% on SPSA for Soft-kNN). **CIFAR-100:** Soft-kNN dominates across both accuracy and robustness, simultaneously improving clean accuracy to 60.42% (vs. 55.79% for TRADES and 56.78% for PGD-AT) while outperforming all defenses across every evaluated attack, including targeted APGD-DLR (26.10%), FAB-T (27.77%), ART-PGD (30.36%), Square Attack (31.94%), and SPSA (42.97%). CKNNA achieves the overall highest clean accuracy (60.82%).
Imagenette: CKNNA achieves the highest clean accuracy of any defense (88.46%, $+3.77\%$ over TRADES and $+1.81\%$ over PGD-AT) while remaining strong against APGD-DLR (61.89%). Soft-kNN secures the highest robust accuracy under FAB-T (64.21%), Square Attack (74.11%), and SPSA (84.33%)

5.2 PROBING THE LATENT GEOMETRY: RESTORING FEATURE SPECIALIZATION

To study the influence of our methods on the learned representation and compare them with established regimes, we compare their similarity. Cianfarani et al. (2022) establishes CKA as a metric for quantifying these influences and shows that adversarial training severely distorts a network's feature manifold. Specifically, robust networks lose the hierarchical "block structure" typical of the ResNet family of classifiers (indicating poor feature specialization) and suffer from extreme representation drift in deeper layers. A comprehensive study can be seen in Figure 2. *An exhaustive atlas of representation similarity heatmaps detailing cross-method comparisons (e.g., PGD vs. Soft-kNN) and internal similarities across all metrics and datasets is provided in Appendix D.*

5.3 ABLATIONS

Sensitivity to Alignment Strength (λ): We thoroughly ablated the impact of the alignment strength λ. *The complete numerical results for $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ across both 10-step and 20-step PGD training for all datasets are provided in Appendix C.* Excessively high values of λ lead to *feature collapse*—where the model ignores the primary classification task to just satisfy the alignment constraint, resulting in severely degraded clean accuracy. Conversely, very low values of λ cause the model to degenerate back into standard PGD-AT behavior.

Sensitivity to Soft-kNN Temperature (τ): For Soft-kNN, tuning the temperature parameter τ was equally critical; an intermediate temperature of $\tau = 0.1$ provided a sufficiently sharp gradient signal without causing saturation in the neighborhood softmax operation. The temperature parameter τ in Soft-kNN controls the sharpness of the neighborhood similarity. We ablated $\tau \in \{0.05, 0.1, 0.2, 0.3, 0.5\}$ and found that lower values ($\tau \leq 0.1$) are required to maintain high robust accuracy. Lower temperatures strictly penalize feature drift by enforcing alignment with

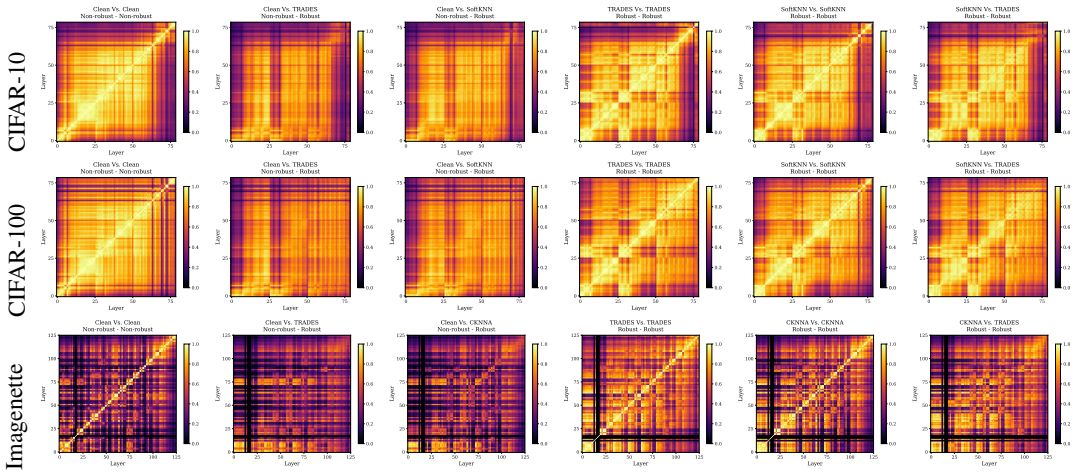


Figure 2: A layer-wise CKA similarity heatmap matrix computed on representations across CIFAR-10, CIFAR-100 and Imagenette. The sequence of plots demonstrates: **Col 1 (Ideal Benign Structure):** Standard models exhibit healthy, specialized block structures. **Cols 2 & 3 (Representation Drift):** TRADES drifts strongly from the Clean representations in deep layers, whereas **RepAlign** framework anchors features much closer to the clean space. **Cols 4 & 5 (Structural Collapse & Restoration):** TRADES loses its internal hierarchical block structure, suffering from extreme cross-layer similarity. Our methods successfully restore this specialized hierarchy. **Col 6 (Divergent Geometries):** Direct comparison reveals our methods learn deep-layer features that also bear structural similarity to TRADES explaining the robust accuracy. (*Note: "Ours" refers to Soft-kNN for CIFAR-10 and CIFAR-100 and CKNNA for SVHN and Imagenette*).

the exact nearest neighbors. In contrast, higher values smooth the distribution, washing out local structure and reducing robustness. Results for CIFAR-10 dataset are provided in §C.1.

Impact of Alignment Layer Depth: Our primary results apply **RepAlign** to the penultimate layer. To test if earlier alignment improves performance, we also evaluated our methods on the intermediate (middle) layers. We found that enforcing representation alignment too early over-constrains the network, limiting its ability to learn useful features for the primary classification task. Consequently, applying **RepAlign** to the middle layers results in lower clean and robust accuracy compared to the penultimate layer. The complete results for these experiments are provided in §C.3.

6 Discussion and Future Work

Summary: Representation alignment provides a useful regularization mechanism to improve trade-offs in adversarial training. Among all methods, CKNNA and Soft-kNN provide the best trade-offs between clean and robust accuracy.

Limitations: The effectiveness of our methods relies on the selection of the adversary, choice of regularization layer and the kernel bandwidth, adding to the hyperparameter search space. In our experiments, we have only trained against an (Madry et al., 2018) ℓ_∞ based adversary at a budget of $8/255$, which we plan to extend in future work. There might also be multiple variations of this loss formulation using different clean and robust losses for classification along with the **RepAlign** based alignment at multiple layers.

Future Work: This framework opens several exciting avenues for future work. Investigating the impact of local similarity metrics like CKNNA and Soft-kNN could reveal the importance of preserving local neighborhood structure for robustness. A deeper theoretical analysis, analogous to the derivation of TRADES, could provide formal bounds on the robust error based on the perturbation in the feature space, similar to what TRADES does in the probability space. Finally, extending this principle to other domains, such as multi-modal learning, is a promising direction, where representation similarity between models for different modalities may exhibit greater robustness to adversarial perturbations that transfer through shared semantic structure.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks with required disclosure (such as theoretical modeling, proof writing, method implementation, synthetic data generation, or result interpretation). Additionally, we used generative AI tools (LLMs) for tasks with recommended disclosure: identifying relevant literature to expand our related work survey, generating boilerplate visualization scripts (such as plots and heatmaps), assisting with debugging code errors, and editing our original text to improve clarity and readability.

We have thoroughly reviewed all AI-assisted work. All papers suggested by LLMs were manually verified, downloaded, and read by the authors to confirm their relevance and avoid hallucinated citations. All visualization and debugging code was manually verified and tested for accuracy before running experiments. All original text remained authored by us, with LLM suggestions used solely for grammatical refinement. We take full responsibility for the final content of this work, including all text, code, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, all loss functions and algorithmic details for RepAlign are provided in §3. Complete training hyperparameters, architectures, and evaluation protocols are described in §4 and the Appendix. For main results, we report mean and standard deviation across multiple random seeds. Complete, anonymous source code to reproduce all models, evaluations, and heatmaps is provided in the supplementary material and available at `https://anonymous.4open.science/r/RepAlign`.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, pp. 484–501, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58591-4. doi: 10.1007/978-3-030-58592-1_29. URL https://doi.org/10.1007/978-3-030-58592-1_29.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/athalye18a.html.

Léon Bottou. *Stochastic Gradient Descent Tricks*, pp. 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_25. URL https://doi.org/10.1007/978-3-642-35289-8_25.

Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, Los Alamitos, CA, USA, May 2017. IEEE Computer Society. doi: 10.1109/SP.2017.49. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.49.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/32e0bd1497aa43e02a42f47d9d6515ad-Paper.pdf.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qZzy5urZw9.

Zhen Cheng, Fei Zhu, Xu-Yao Zhang, and Cheng-Lin Liu. Delving into feature space: Improving adversarial robustness by feature spectral regularization. In *Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part XXVI*, pp.

423–439, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-78394-4. doi: 10.1007/978-3-031-78395-1_28. URL https://doi.org/10.1007/978-3-031-78395-1_28.

Christian Cianfarani, Arjun Nitin Bhagoji, Vikash Sehwag, Ben Zhao, Heather Zheng, and Prateek Mittal. Understanding robust learning through the lens of representation similarities. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34912–34925. Curran Associates, Inc., 2022. doi: 10.52202/068431-2530. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/e1fa017a312368906411501bbd27a1d6-Paper-Conference.pdf.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2206–2216. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/croce20b.html.

Francesco Croce, Jonas Rauber, and Matthias Hein. Scaling up the randomized gradient-free adversarial attack reveals overestimation of robustness using established attacks. *International Journal of Computer Vision*, 128(4):1028–1046, 2020. doi: 10.1007/s11263-019-01213-0. URL https://doi.org/10.1007/s11263-019-01213-0.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=SSKZPJCt7B.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2012–2020. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/frosst19a.html.

Zeinab Golgooni, Mehrdad Saberi, Masih Eskandar, and Mohammad Hossein Rohban. Zerograd : Mitigating and explaining catastrophic overfitting in FGSM adversarial training. *CoRR*, abs/2103.15476, 2021. URL https://arxiv.org/abs/2103.15476.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021a. URL https://arxiv.org/abs/2010.03593.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4218–4233. Curran Associates, Inc., 2021b. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/21ca6d0cf2f25c4dbb35d8dc0b679c3f-Paper.pdf.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. 20, 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/d5cfead94f5350c12c322b5b664544c1-Paper.pdf.

Jeremy Howard. imagenette. https://github.com/fastai/imagenette, 2019.

Qian Huang, Horace Isola, Phillip andeh, and Kevin Zhang. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4733–4742, 2019.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. URL https://auai.org/uai2018/proceedings/papers/313.pdf.

Charles Jin and Martin Rinard. Manifold regularization for locally stable deep neural networks, 2020. URL https://arxiv.org/abs/2003.04286.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing, 2018. URL https://arxiv.org/abs/1803.06373.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJm4T4Kgx.

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. URL https://openreview.net/forum?id=rJzIBfZAb.

Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6640–6650. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/maini20a.html.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks, 2019. URL https://arxiv.org/abs/1810.12042.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, et al. Adversarial robustness toolbox v1.0.0. *arXiv preprint arXiv:1807.01069*, 2018. URL https://arxiv.org/abs/1807.01069.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Xb8xvrtB8Ce.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016. doi: 10.1109/SP.2016.41.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349444. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

Leo Hyun Park, Jaeuk Kim, Myung Gyo Oh, Jaewoo Park, and Taekyoung Kwon. Adversarial feature alignment: Balancing robustness and accuracy in deep learning via adversarial training. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, AISec '24, pp. 101–112, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400712289. doi: 10.1145/3689932.3694765. URL https://doi.org/10.1145/3689932.3694765.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29935–29948. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fb4c48608ce8825b558ccf07169a3421-Paper.pdf.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pp. 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL https://proceedings.mlr.press/v2/salakhutdinov07a.html.

Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WVX0NNVBBkV.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? – a comprehensive study on the robustness of 18 deep image classification models. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XII*, pp. 644–661, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01257-1. doi: 10.1007/978-3-030-01258-8_39. URL https://doi.org/10.1007/978-3-030-01258-8_39.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5025–5034. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/uesato18a.html.

Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oU3aTsmeRQV.

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. URL https://openreview.net/forum?id=rklOg6EFwS.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2958–2969. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1ef91c212e30e14bf125e9374262401f-Paper.pdf.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

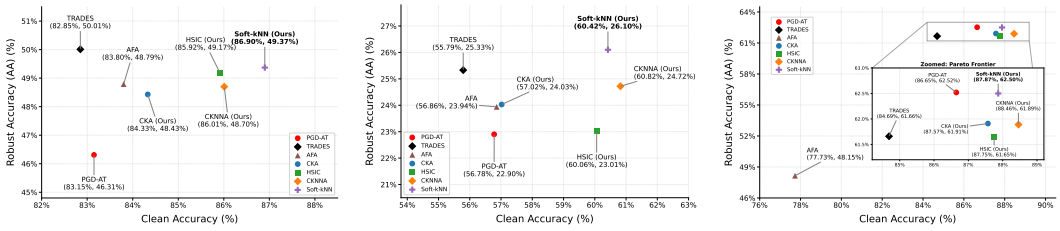
Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11278–11287. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20z.html.

This Appendix provides additional information corresponding to each section of the main paper as follows:

- **§A: Extended Pareto Frontiers.** Clean vs. AutoAttack robust accuracy trade-offs for all methods across CIFAR-10, CIFAR-100, and Imagenette.
- **§B: Extended Baseline Evaluations.** Full PGD-AT and TRADES results across initialization (scratch vs. pretrained) and inner-maximization steps (10 vs. 20).
- **§C: Hyperparameter Ablations.** Alignment-strength (λ) sweeps for all four RepAlign regularizers, the Soft-kNN temperature (τ), the TRADES β trade-off, and applying RepAlign at the middle layer.
- **§D: Representation Similarity Atlas.** Complete layer-wise CKA heatmaps: internal geometry, drift from the clean model, baselines vs. RepAlign, and RepAlign variants against each other.
- **§E: AFA Baseline.** Our reimplementation of Adversarial Feature Alignment, both training protocols, configurations and multi-attack results.
- **§F: Training Cost.** Per-epoch and end-to-end training time for all methods and datasets.

A Extended Pareto Frontiers

Figure 3 presents the Pareto frontiers of Clean Accuracy versus Robust Accuracy (evaluated under the AutoAttack suite) across CIFAR-10, CIFAR-100, and Imagenette. Standard adversarial training methods like PGD-AT and TRADES suffer from an acute trade-off, where gaining robustness incurs a steep drop in clean performance. By directly regularizing deep representations against clean anchors, RepAlign breaks this barrier. Across all three datasets, our representation alignment regularizers (most notably **Soft-kNN** and **CKNNA**) push the Pareto frontier up and to the right, consistently delivering a strictly superior trade-off compared to PGD-AT and TRADES.



(a) CIFAR-10 (b) CIFAR-100 (c) Imagenette

Figure 3: Clean vs. Robust Accuracy Pareto frontiers across CIFAR-10, CIFAR-100, and Imagenette evaluated under AutoAttack. Our proposed representation alignment methods (e.g., Soft-kNN, CKNNA) consistently push the frontier up and to the right, achieving a strictly superior trade-off compared to PGD-AT and TRADES across all datasets.

Note: For Imagenette, the plot zooms into the pareto frontier as AFA falls far off-scale (77.73% clean, 48.15% robust).

B Extended Baseline Evaluations: Initialization and Attack Steps

In §5, we reported the best-performing baseline models to provide a fair and rigorous comparison against our proposed methods. Here, we present the exhaustive baseline evaluations. Table 2 details the impact of two critical training hyperparameters for PGD-AT and TRADES:

1. **Initialization Strategy:** We compare training from scratch (random initialization) versus utilizing a pretrained clean checkpoint.

Table 2: Comprehensive baseline evaluations across CIFAR-10, CIFAR-100, and Imagenette comparing initialization (Scratch vs. Pretrained) and training PGD steps (10 vs. 20) across white-box and black-box attacks under an ℓ_∞ budget of $\epsilon = 8/255$. APGD-DLR and FAB-T denote the targeted variants used in AutoAttack. The overall best value per dataset and method is bolded.

					White-Box Attacks				Black-Box Attacks	
Dataset	**Method**	**Init**	**Steps**	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CIFAR-10	PGD-AT	Scratch	10	79.54	47.93	44.64	45.39	48.52	51.60	63.78
		Scratch	20	79.10	48.69	45.19	45.90	49.29	52.45	63.91
		Pretrained	10	82.59	**49.37**	46.20	47.20	**50.33**	54.37	66.10
		Pretrained	20	**83.15**	49.13	**46.32**	**47.21**	49.87	**54.87**	**66.89**
	TRADES ($\beta = 6$)	Scratch	10	83.34	51.40	49.56	51.00	52.06	55.59	67.54
		Scratch	20	**83.43**	52.02	49.76	51.05	52.46	56.09	67.80
		Pretrained	10	83.18	52.19	**50.04**	**51.57**	**53.42**	56.39	**67.95**
		Pretrained	20	82.85	**52.63**	50.00	51.42	53.10	**56.59**	67.57
CIFAR-100	PGD-AT	Scratch	10	50.04	25.73	21.85	22.10	26.12	26.29	36.66
		Scratch	20	49.50	25.38	21.64	21.86	25.85	26.52	36.62
		Pretrained	10	53.80	**26.18**	**22.45**	**22.65**	**26.63**	27.63	38.20
		Pretrained	20	**54.77**	25.91	21.96	22.13	26.52	**27.83**	**38.93**
	TRADES ($\beta = 6$)	Scratch	10	**57.09**	28.66	25.28	25.56	29.00	**29.50**	**40.76**
		Scratch	20	56.12	28.55	**25.44**	**25.72**	28.84	29.30	39.58
		Pretrained	10	56.25	28.71	25.20	25.34	29.06	29.14	39.91
		Pretrained	20	56.07	**28.83**	25.31	25.62	**29.18**	29.25	39.93
Imagenette	PGD-AT	Scratch	10	84.92	63.08	60.97	62.27	63.36	70.01	80.48
		Scratch	20	86.75	64.46	62.22	63.54	64.89	71.95	82.57
		Pretrained	10	**88.15**	**65.81**	**63.95**	**64.99**	**66.55**	**74.17**	**84.31**
		Pretrained	20	86.70	64.51	62.50	63.62	64.87	72.03	82.75
	TRADES ($\beta = 6$)	Scratch	10	9.86	9.86	9.86	9.86	9.86	9.86	9.86
		Scratch	20	9.86	9.86	9.86	9.86	9.86	9.86	9.86
		Pretrained	10	**85.45**	**65.61**	**63.24**	**65.20**	**66.14**	70.22	**81.10**
		Pretrained	20	84.69	65.07	61.66	63.26	65.43	**70.29**	80.31

2. **Inner Maximization Steps:** We compare the robustness yielded by 10-step versus 20-step PGD during the training phase.

Pretrained Initialization Prevents Training Collapse: On Imagenette, TRADES trained from scratch completely collapses to trivial majority-class predictions (yielding only 9.86% accuracy under both 10-step and 20-step inner maximization). Initializing from a clean pretrained checkpoint completely resolves this failure mode, achieving over 84.6% clean accuracy and strong robust generalization ($> 65\%$ APGD-CE).

Pretrained Models Yield Superior Robustness and Clean Trade-offs: Across CIFAR-10 and CIFAR-100, pretrained initialization consistently achieves higher clean accuracy (e.g., $+4.05\%$ on CIFAR-10 PGD-AT and $+5.27\%$ on CIFAR-100 PGD-AT) while simultaneously improving robust accuracy against white-box and black-box attacks (AutoAttack, Square, and SPSA).

Impact of Inner Maximization Steps (10 vs. 20 Steps): Increasing inner maximization steps from 10 to 20 during training provides marginal robustness gains on CIFAR-10 and CIFAR-100 (e.g., $+0.44\%$ APGD-CE for TRADES on CIFAR-10), but 10-step training remains highly competitive while requiring half the inner-loop computation and often preserving slightly higher clean accuracy.

C Comprehensive Hyperparameter Ablations for **RepAlign** framework

In this section, we present the exhaustive hyperparameter sweeps for our Representation Alignment Regularizers (CKA, HSIC, CKNNA, and Soft-kNN). As discussed in §5.3, the alignment strength λ balances standard classification accuracy against how strictly we align the clean and adversarial features.

To provide a standardized comparison across all datasets and methods, we report evaluations for $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ and on PGD steps 10 and 20. All models in these sweeps were initialized from a clean pretrained checkpoint to isolate the regularization effect. The results for CIFAR-10 are detailed in Table 3, Table 4, for CIFAR-100 are detailed in Table 5, Table 6 and for Imagenette are detailed in Table 7, Table 8 respectively.

Table 3: **CIFAR-10 Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 10-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	82.20	47.50	45.41	46.76	48.31	53.57	65.20
	1.0	**86.50**	48.36	47.21	**49.71**	49.25	55.88	68.30
	5.0	84.98	**50.12**	**47.69**	49.43	**51.74**	**56.38**	**68.44**
	10.0	82.18	47.27	45.14	47.08	49.15	53.88	65.05
CKNNA	0.1	86.32	48.09	46.93	49.47	48.87	55.77	68.10
	1.0	86.42	47.92	46.61	49.43	48.82	55.46	68.02
	5.0	**86.51**	50.46	48.84	**51.15**	52.39	57.36	**69.44**
	10.0	85.10	**51.47**	**49.00**	50.63	**53.28**	**57.65**	69.43
HSIC	0.1	**86.78**	47.66	46.65	49.55	48.44	55.28	67.76
	1.0	86.58	48.07	46.74	49.56	48.86	55.64	67.93
	5.0	85.06	**50.90**	**48.51**	**50.31**	**52.46**	**57.43**	**68.59**
	10.0	82.10	48.57	45.81	47.70	50.17	54.08	65.48
Soft-kNN ($\tau = 0.1$)	0.1	86.12	47.47	46.37	48.85	48.42	55.39	67.47
	1.0	81.91	48.63	45.72	46.67	49.19	54.12	66.21
	5.0	**86.91**	**51.55**	**49.71**	**51.87**	**53.41**	**58.70**	**70.14**
	10.0	84.87	51.47	49.24	50.88	**53.41**	57.40	69.23

Table 4: **CIFAR-10 Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 20-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	83.34	48.41	45.46	46.56	49.29	54.43	66.82
	1.0	**85.96**	48.70	47.52	**49.75**	49.41	56.25	68.09
	5.0	84.33	**50.92**	**48.42**	49.68	**51.61**	**56.80**	**68.23**
	10.0	81.92	48.35	45.73	47.77	49.77	53.82	65.25
CKNNA	0.1	86.13	47.89	46.81	49.30	48.69	55.33	67.35
	1.0	**86.16**	48.39	47.15	49.84	49.18	55.66	67.96
	5.0	86.01	50.56	48.73	**51.00**	51.49	**57.88**	**69.76**
	10.0	85.07	**51.42**	**48.90**	50.85	**53.29**	57.73	69.12
HSIC	0.1	85.86	47.38	46.37	48.93	48.01	55.22	67.69
	1.0	82.45	48.38	45.65	46.88	48.89	53.63	66.09
	5.0	**85.92**	**51.52**	**49.17**	**51.21**	**52.17**	**57.91**	**69.54**
	10.0	83.38	48.97	46.50	48.27	50.67	55.00	66.22
Soft-kNN ($\tau = 0.1$)	0.1	86.09	47.81	46.61	49.35	48.58	55.33	68.06
	1.0	82.40	48.25	45.28	46.34	48.94	54.05	65.26
	5.0	**86.90**	51.23	49.37	**51.30**	51.82	**58.09**	**70.17**
	10.0	84.20	**52.22**	**49.42**	50.84	**53.81**	57.95	69.17

Table 5: **CIFAR-100 Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 10-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	57.52	24.82	22.13	22.90	25.49	28.43	39.42
	1.0	**59.91**	24.51	21.89	**26.14**	25.30	26.83	38.29
	5.0	57.18	25.22	22.13	22.79	25.72	28.35	39.62
	10.0	58.49	**27.72**	**23.65**	24.67	**28.45**	**29.92**	**41.50**
CKNNA	0.1	57.98	25.50	22.43	23.25	26.08	28.51	39.83
	1.0	59.25	24.75	21.81	26.04	25.41	26.49	38.09
	5.0	58.93	25.21	22.39	26.36	25.81	26.89	38.13
	10.0	**61.09**	**27.28**	**24.47**	**26.62**	**28.05**	**30.54**	**41.55**
HSIC	0.1	56.12	24.84	21.76	22.64	25.29	27.81	38.81
	1.0	57.32	**25.05**	22.16	23.04	**25.56**	**28.56**	39.42
	5.0	59.09	24.36	21.84	**25.62**	25.18	26.92	37.63
	10.0	**59.72**	24.99	**23.10**	24.76	25.46	28.45	**39.43**
Soft-kNN ($\tau = 0.1$)	0.1	59.92	23.88	22.11	25.46	24.57	27.49	37.70
	1.0	59.56	24.98	22.01	25.67	25.78	27.19	38.63
	5.0	60.13	26.54	23.32	25.65	27.23	29.13	40.51
	10.0	**60.23**	**29.83**	**26.32**	**27.74**	**30.58**	**31.79**	**43.05**

Table 6: **CIFAR-100 Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 20-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	55.95	24.81	22.13	23.08	25.39	27.99	38.57
	1.0	59.58	24.86	22.46	**26.35**	25.82	27.37	38.31
	5.0	**60.79**	25.38	23.55	25.21	26.07	28.99	40.40
	10.0	57.02	**27.62**	**24.02**	24.76	**28.30**	**29.94**	**40.82**
CKNNA	0.1	55.42	25.30	21.46	22.19	25.83	28.05	38.92
	1.0	55.57	25.10	22.21	23.12	25.50	28.14	38.94
	5.0	58.82	25.59	22.58	**26.59**	26.16	26.77	38.31
	10.0	**60.82**	**27.43**	**24.72**	26.50	**28.11**	**30.16**	**41.46**
HSIC	0.1	56.99	25.15	21.69	22.51	25.64	27.83	39.22
	1.0	55.74	25.36	21.69	22.48	25.92	27.81	38.86
	5.0	54.57	**25.74**	22.07	22.91	**26.28**	27.96	38.44
	10.0	**60.05**	25.13	**23.00**	**25.00**	25.71	**28.60**	**39.37**
Soft-kNN ($\tau = 0.1$)	0.1	55.79	24.99	21.94	22.87	25.54	27.60	38.84
	1.0	58.09	25.06	22.55	23.58	25.67	28.65	39.54
	5.0	59.83	26.40	23.25	25.48	27.09	28.91	40.62
	10.0	**60.42**	**29.43**	**26.10**	**27.54**	**30.34**	**31.96**	**43.00**

Table 7: **Imagenette Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 10-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	87.49	63.06	61.30	**63.41**	64.56	72.89	83.69
	1.0	**87.80**	**63.82**	**61.61**	63.18	**65.04**	**72.92**	**83.87**
	5.0	79.24	56.89	53.83	56.38	57.30	62.32	74.24
	10.0	32.03	24.28	22.37	26.09	24.82	24.15	28.99
CKNNA	0.1	87.67	63.54	62.01	63.72	65.07	73.45	84.51
	1.0	**88.38**	**64.10**	**62.55**	**64.48**	**65.25**	**73.83**	**85.45**
	5.0	85.71	63.01	60.56	62.47	63.46	71.59	82.62
	10.0	83.39	60.74	57.94	60.38	61.32	67.82	80.56
HSIC	0.1	**88.00**	63.31	61.15	63.24	64.64	**73.20**	**83.90**
	1.0	87.36	**63.87**	**61.94**	**64.08**	**64.89**	73.12	83.69
	5.0	19.21	17.35	16.92	17.17	18.29	17.10	18.90
	10.0	20.08	16.84	16.00	16.61	19.26	16.33	19.97
Soft-kNN ($\tau = 0.1$)	0.1	**87.62**	63.39	**61.58**	**63.31**	64.59	**73.02**	**83.90**
	1.0	**87.62**	**63.52**	61.55	63.21	**64.84**	72.87	83.54
	5.0	76.64	54.68	51.97	53.63	54.98	59.77	72.87
	10.0	54.68	40.13	37.27	40.28	40.56	41.32	50.78

Table 8: **Imagenette Ablations:** Hyperparameter sweeps for alignment strength $\lambda \in \{0.1, 1.0, 5.0, 10.0\}$ on 20-step PGD. Best values within each method are bolded.

			White-Box Attacks				Black-Box Attacks	
Method	λ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CKA	0.1	**88.03**	**64.20**	**62.47**	**64.23**	**65.02**	**73.71**	**84.82**
	1.0	87.57	63.75	61.91	63.49	64.43	73.17	83.52
	5.0	81.43	57.76	55.08	58.52	58.27	64.05	76.76
	10.0	31.03	24.23	21.89	25.43	24.84	22.93	28.66
CKNNA	0.1	87.21	63.46	61.43	**63.72**	64.05	73.10	**84.33**
	1.0	**88.46**	**63.62**	**61.89**	63.18	**64.48**	**74.11**	84.30
	5.0	85.50	63.24	60.48	62.88	63.85	71.44	82.19
	10.0	82.45	60.87	57.76	59.85	61.45	67.64	79.34
HSIC	0.1	**87.97**	**64.03**	**62.45**	**64.46**	**64.28**	**72.89**	**84.03**
	1.0	87.75	63.36	61.66	63.46	63.90	**72.89**	83.92
	5.0	26.14	20.10	18.04	18.57	20.48	19.80	24.79
	10.0	19.06	17.25	16.64	16.76	17.83	16.71	18.80
Soft-kNN ($\tau = 0.1$)	0.1	87.41	63.29	61.32	62.65	64.10	73.20	84.05
	1.0	**87.87**	**64.56**	**62.50**	**64.20**	**65.17**	**74.09**	**84.38**
	5.0	76.82	55.52	52.43	53.91	55.92	60.56	72.92
	10.0	41.94	30.45	27.67	28.66	30.73	30.29	39.24

C.1 Soft-kNN Temperature (τ) Ablation

This section provides the full evaluation of the temperature parameter (τ) used in the Soft-kNN alignment metric, supporting the observations made in §5.3. The temperature controls how strictly the adversarial example must match the local neighborhood of its clean counterpart. The results in Table 9 show that a sharper distribution (lower τ) preserves local neighborhood structure much better under adversarial attack. When τ is set too high ($\tau \geq 0.2$), the alignment penalty becomes too diffuse, and the robust accuracy across both white-box attacks (APGD-CE, APGD-DLR, FAB-T, ART-PGD) and black-box/gradient-free attacks (Square, SPSA) drops compared to the $\tau = 0.1$ results reported in the main text.

Table 9: Ablation of the Soft-kNN temperature parameter (τ) on CIFAR-10 across PGD steps (10, 20) and regularizer weights $\lambda \in \{5.0, 10.0\}$. Lower temperatures consistently yield better robust accuracy by strictly enforcing local neighborhood alignment. Best values within each block are bolded.

				White-Box Attacks				Black-Box Attacks	
PGD Steps	λ	τ	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
10	5.0	0.05	**86.59**	**51.37**	**49.52**	**51.44**	**52.41**	**58.50**	**70.20**
		0.2	86.51	47.34	46.12	49.09	48.07	55.20	67.70
		0.3	85.95	47.38	46.43	49.04	48.07	55.05	67.34
		0.5	82.69	47.42	44.87	45.88	48.28	53.47	65.65
	10.0	0.05	83.41	**50.83**	**48.62**	**50.35**	**51.77**	**56.45**	67.62
		0.2	86.13	47.60	46.59	49.13	48.41	55.57	67.36
		0.3	**86.33**	47.70	46.49	49.27	48.58	55.54	**67.67**
		0.5	85.86	46.97	46.01	49.23	47.97	54.99	67.03
20	5.0	0.05	**86.05**	**51.99**	**49.99**	**52.02**	**53.02**	**58.37**	**69.97**
		0.2	82.54	47.93	45.03	46.03	48.43	54.44	66.18
		0.3	83.80	48.61	45.78	47.02	49.23	54.88	66.93
		0.5	81.39	47.62	44.44	45.59	48.47	52.85	64.72
	10.0	0.05	81.47	**51.18**	**48.06**	49.67	**51.97**	**56.05**	66.40
		0.2	85.68	48.66	47.34	**49.84**	49.53	55.87	67.73
		0.3	86.19	48.29	46.92	49.39	49.06	55.88	**68.10**
		0.5	**86.23**	47.47	46.24	49.11	48.21	54.91	67.31

C.2 TRADES Beta (β) Ablation

In TRADES, the regularizer weight β explicitly controls the trade-off between natural classification error and adversarial boundary invariance via $\mathcal{D}_{\text{KL}}(f(x) \,\|\, f(x_{\text{adv}}))$. As shown in Table 10, increasing β from 0.5 to 20.0 on CIFAR-10 raises 20-step PGD robustness from $\sim 47.1\%$ up to $\sim 54.4\%$, while natural clean accuracy steadily declines from $\sim 88.6\%$ to $\sim 78.6\%$. Empirically, the most favorable trade-off is achieved around $\beta \in [5.0, 6.0]$, as larger values begin to significantly degrade clean accuracy (dropping below 80% at $\beta = 20.0$ and causing numerical divergence with NaNs for warm-started models at $\beta \geq 50.0$). Therefore, consistent with the original TRADES recommendation (Zhang et al., 2019) and our empirical balance between clean and robust performance, we adopt $\beta = 6.0$ across all our TRADES experiments. Furthermore, across all stable regimes ($\beta \leq 20.0$), warm-started models closely trace the same trade-off curve as training from scratch, verifying that warm-starting from benign representations preserves optimal robust capacity.

C.3 Middle Layer

This section provides the full evaluation of the **RepAlign** framework when applied to the intermediate layers (here done one middle layer) of the networks. Specifically, `block2.layer.3.add` for WideResNet-28-10 and `layer2.3.add` for ResNet-50. As discussed in §5.3, forcing the representations to align this early over-constrains the network. While it successfully reduces feature distortion, it prevents the remaining layers from learning good classification features. Consequently, both clean and robust accuracy drop slightly across all three datasets compared to our penultimate-layer models. Full numerical results are detailed in Table 11.

Table 10: **Effect of TRADES Regularization Strength β on CIFAR-10.** Evaluation of clean accuracy and 20-step PGD robustness ($\epsilon = 8/255$) across varying β, comparing training from scratch against warm-starting from a clean checkpoint. The default setting $\beta = 6.0$ used across all our experiments is bolded.

	From Scratch		Pretrained (Warm-Start)	
β	**Clean (%)**	**PGD-20 (%)**	**Clean (%)**	**PGD-20 (%)**
0.5	88.18	47.17	88.57	47.62
1.0	87.02	47.00	87.46	48.50
2.0	85.86	49.04	86.59	48.86
5.0	84.44	52.00	83.98	52.28
6.0	**83.34**	**52.00**	**83.19**	**52.81**
10.0	82.18	53.31	81.58	53.74
20.0	79.71	54.38	78.58	54.16
50.0	76.44	53.82	*Diverged (NaN)*	
100.0	73.07	52.66	*Diverged (NaN)*	

Table 11: Comprehensive evaluation across white-box and black-box attacks when applying regularizers at the **middle layer** under an ℓ_∞ budget of $\epsilon = 8/255$. For RepAlign, $\lambda = 5.0$ (CIFAR-10 and CIFAR-100), $\lambda = 1.0$ (Imagenette), and $\tau = 0.1$ for Soft-kNN. Baselines use $\beta = 6$ (TRADES) and PGD_{20}. APGD-DLR and FAB-T denote targeted variants in AutoAttack. Best values among completed defense runs are bolded.

			White-Box Attacks				Black-Box Attacks	
Dataset	**Method**	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CIFAR-10	Clean Training	96.17	0.00	0.00	3.59	0.02	1.04	0.25
	PGD-AT	83.15	49.13	46.32	47.21	49.87	54.87	66.89
	TRADES	82.85	**52.63**	**50.00**	**51.42**	**53.10**	**56.59**	**67.57**
	CKA (Ours)	80.28	45.82	42.42	43.30	46.40	51.27	63.25
	CKNNA (Ours)	**85.37**	45.90	44.39	47.54	46.79	54.15	66.16
	HSIC (Ours)	73.80	41.95	38.61	39.90	42.81	45.85	58.31
	Soft-kNN (Ours)	84.05	45.52	43.79	46.33	46.27	52.63	65.46
CIFAR-100	Clean Training	80.98	0.00	0.00	5.92	0.02	0.49	0.19
	PGD-AT	54.77	25.91	21.96	22.13	26.52	27.83	38.93
	TRADES	**56.07**	**28.83**	**25.31**	**25.62**	**29.18**	**29.25**	**39.93**
	CKA (Ours)	55.61	24.61	21.22	21.88	25.24	27.58	39.12
	CKNNA (Ours)	56.05	24.42	21.07	21.79	24.90	27.26	38.80
	HSIC (Ours)	52.84	24.18	20.71	21.57	24.54	25.70	37.13
	Soft-kNN (Ours)	54.82	24.29	21.01	21.81	24.66	26.89	38.01
Imagenette	Clean Training	91.82	0.00	0.00	7.18	0.00	5.78	39.57
	PGD-AT	86.70	64.51	**62.50**	**63.62**	64.87	72.03	82.75
	TRADES	84.69	**65.07**	61.66	63.26	**65.43**	70.29	80.31
	CKA (Ours)	**87.67**	62.70	60.84	63.54	63.34	72.28	83.52
	CKNNA (Ours)	87.44	63.16	61.40	63.44	63.80	**72.84**	**83.57**
	HSIC (Ours)	86.65	63.01	61.12	63.21	63.52	71.87	82.90
	Soft-kNN (Ours)	86.88	63.52	61.38	63.59	64.08	72.13	83.52

D Complete Atlas of Representation Similarity Heatmaps

In §5.2 we provided a condensed grid of CKA similarity heatmaps to demonstrate representation drift and structural restoration. Here, we present the complete atlas of heatmaps generated during our analysis. These plots exhaustively cover the internal and cross-layer representation similarities across all proposed methods, baselines, and datasets, acting as a visual proof of the geometric modifications induced by our regularizers. All heatmaps are computed using benign feature representations.

D.1 Internal (Self) Representation Geometries

Figure 4 displays the internal similarity (comparing a model against itself) for the baselines and the best-performing proposed method for each dataset. This isolates whether a method successfully learns hierarchical feature specialization (indicated by a block-diagonal structure) or suffers from structural collapse (indicated by high off-diagonal similarity). As expected, standard adversarial training baselines (PGD, TRADES) suffer from internal structural collapse, whereas our proposed alignment methods restore the healthy block-diagonal structure seen in the Clean models.

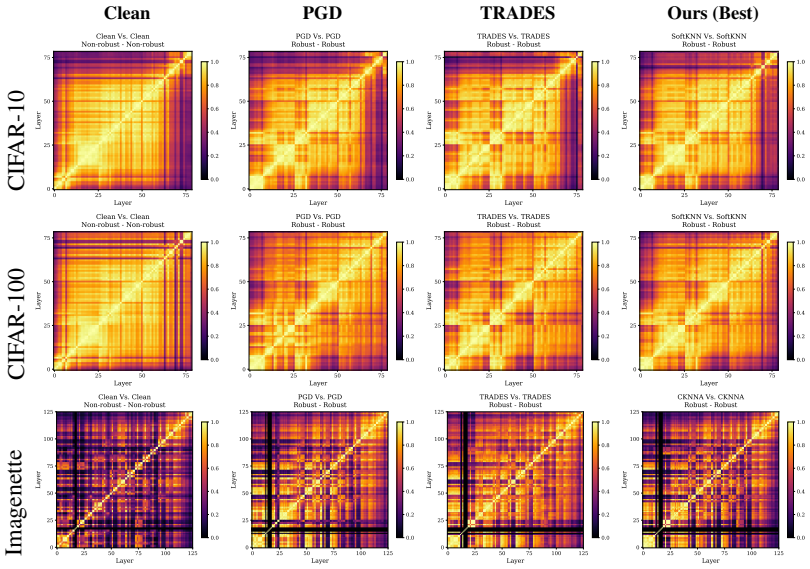


Figure 4: **Internal Similarities (Method vs. Itself):** Comparison of internal latent geometries. "Ours (Best)" refers to Soft-kNN for CIFAR-10 and CIFAR-100, and CKNNA for Imagenette.

D.2 Representation Drift: Clean vs. Robust Models

Figure 5 visualizes the deviation of each robustly trained model from the natural, specialized feature space of the Clean model. PGD and TRADES exhibit severe drift (dark regions in deep layers), whereas our methods successfully anchor the terminal representations closer to the clean manifold.

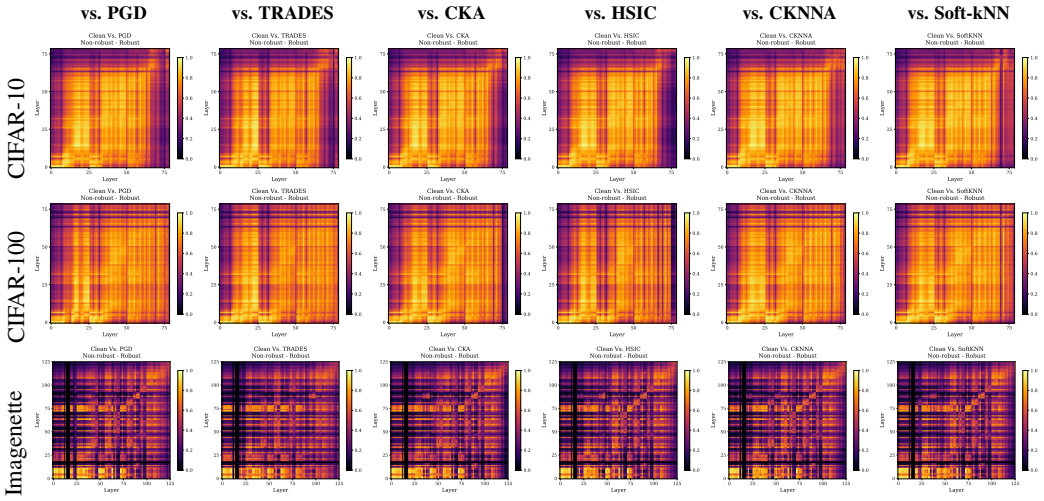


Figure 5: **Representation Drift:** Cross-model CKA similarity comparing the Clean network against all robustly trained models.

D.3 Baselines vs. Proposed Alignment Methods

Figure 6 directly compares the latent geometries of standard adversarial training baselines (PGD, TRADES) against our **RepAlign** methods. The lack of deep-layer similarity highlights that our regularizers do not merely mimic the baselines, but learn fundamentally distinct internal features.

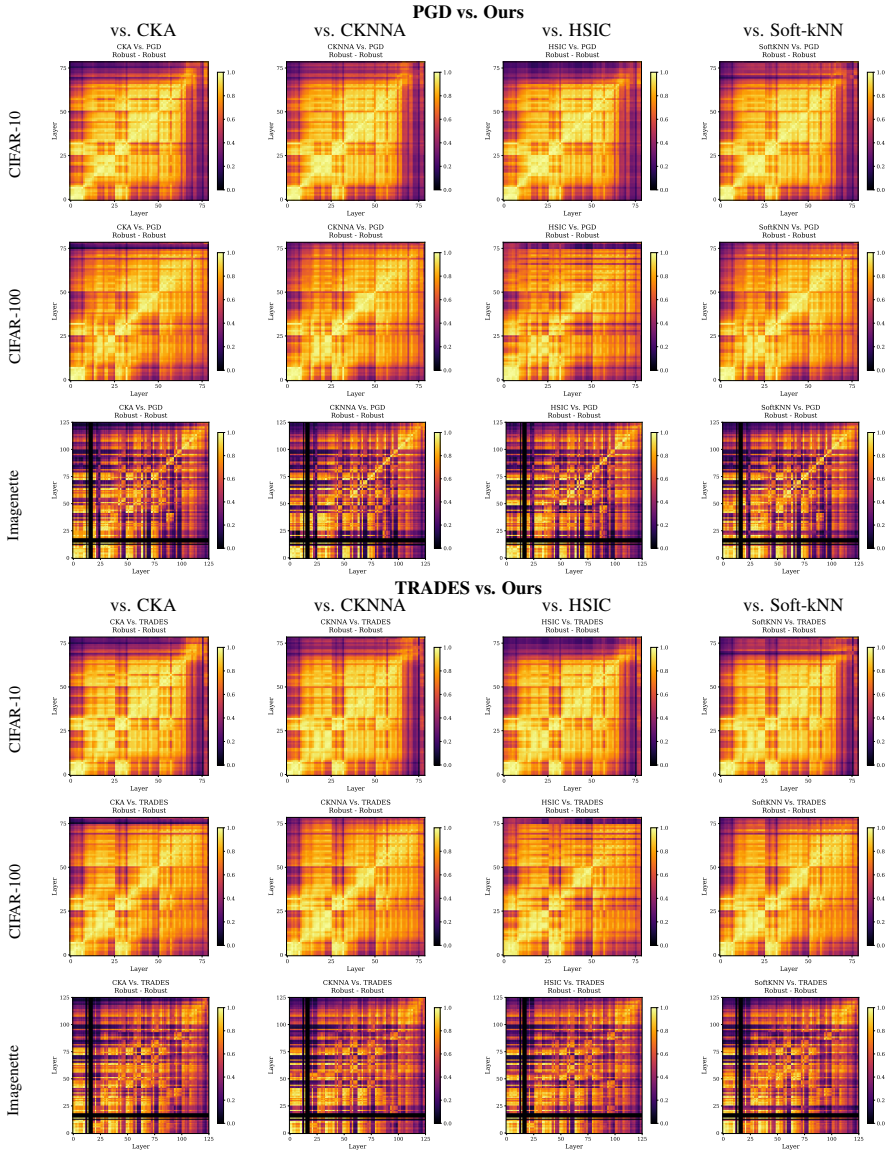


Figure 6: **Divergent Latent Geometries:** Cross-model CKA similarity between the standard baselines (PGD, TRADES) and our proposed representation alignment methods.

D.4 Intra-Alignment Geometries: Method vs. Method

Figure 7 compares our local and global alignment methods against each other, demonstrating the subtle structural differences induced by varying the geometric constraints (i.e., Global topology preservation vs. Local neighborhood preservation).

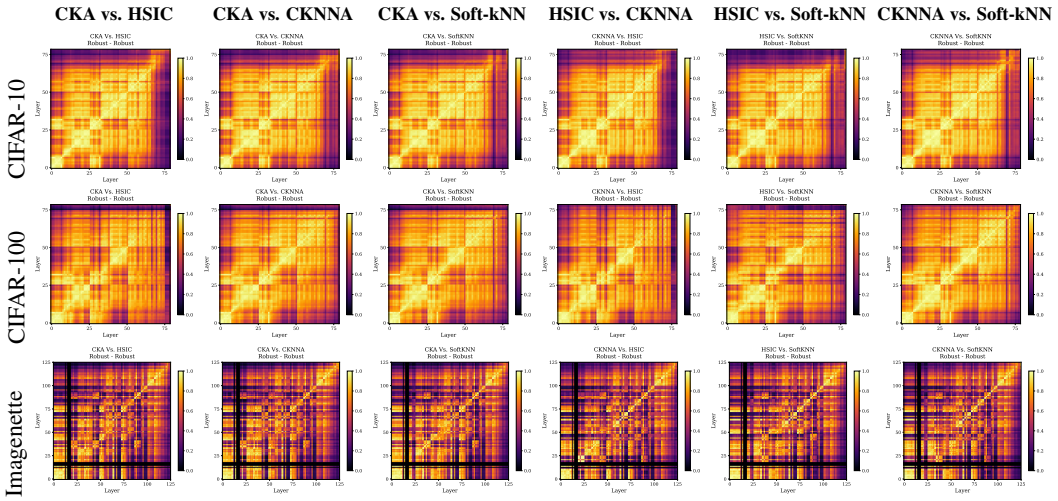


Figure 7: **Intra-Alignment Similarities:** Cross-model CKA directly comparing our proposed global and local representation alignment methods against one another.

E AFA Baseline: Training Protocol and Multi-Attack Evaluation

As an additional point of comparison we reimplement Adversarial Feature Alignment (AFA) (Park et al., 2024) and evaluate the resulting models under the same attack suite used throughout this paper. This section documents the training protocol, the exact configurations used, the resulting robustness numbers, and the evaluation cost.

E.1 Experimental Protocol

AFA uses two distinct training recipes which we reproduce:

- **Track A (pre-train → ALF).** Contrastive AFA/SupCon pre-training of the encoder (Eq. 7 of the original paper), followed by *Adversarial Linear Fine-tuning*: the encoder is frozen and only a freshly initialized linear head is trained adversarially.
- **Track B (joint).** The joint AFA + TRADES objective (Algorithm 1 of the original paper), $\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{TRADES}} + 0.1\,\mathcal{L}_{\text{AFA}}$ with $\beta = 6$, $\lambda_1 = 1.0$, $\lambda_2 = 2.0$, optimized end-to-end. Note that the two tracks are independent runs.

We run both tracks across a 3×2 matrix: three datasets (CIFAR-10, CIFAR-100, Imagenette) $\times$ two initializations (random, and a clean pretrained checkpoint), giving six experiments and twelve checkpoints in total. Inner maximization during training is PGD-10 with $\epsilon = 8/255$ and $\alpha = 2/255$, with the AFA views $\{x, T_1(x)\}$ perturbed and $T_2(x)$ kept clean; inputs are kept in $[0, 1]$ with no normalization, matching the training-time preprocessing. WideResNet-28-10 is used at 32×32 (CIFAR-10, CIFAR-100) and ResNet-50 at 224×224 (Imagenette).

Evaluation. Every checkpoint is evaluated on the *full* test set against the same suite used for our own methods: the four standard AutoAttack components (APGD-CE, APGD-DLR, FAB-T, Square), each run *individually* against the full clean set, plus the gradient-free SPSA attack ($T = 50$ steps, $n = 128$ Rademacher samples per step, $\delta = 0.01$, i.e. 12,800 queries per image).

E.2 Configurations

For CIFAR-10 and CIFAR-100, we train WideResNet-28-10 (32×32), and for Imagenette, ResNet-50 (224×224), using SGD with momentum 0.9 and weight decay 1×10^{-4}. For joint training (Track B, reported in the main paper), models are trained for 76 epochs at batch size 512 (128 for Imagenette), decaying the initial learning rate of 0.1 (0.05 for Imagenette) by $\times 0.1$ at epoch 60; gradient clipping (norm 1.0) and a 20-step warmup were added to stabilize Imagenette against numerical divergence. For the two-stage protocol (Track A), contrastive pretraining runs for 200 epochs with a 10-epoch warmup followed by cosine decay with $\times 0.1$ milestones at epochs 150 and 175 (100 and 150 on Imagenette), using learning rates of 0.25 (CIFAR-10), 1.0 (CIFAR-100), and 0.1 (Imagenette). Adversarial linear fine-tuning (ALF) on frozen representations follows the original paper lengths: 5 epochs for CIFAR-10 and Imagenette, and 25 epochs for CIFAR-100.

E.3 Multi-Attack Evaluation Results for AFA

Table 12: **AFA baseline evaluation** across white-box, boundary, black-box, and gradient-free attacks on the test set of each dataset ($\epsilon = 8/255$). Evaluated across Track A (Adversarial Linear Fine-tuning, ALF) and Track B (Joint training) under both random and clean-pretrained initializations. WideResNet-28-10 is used for CIFAR-10/100, and ResNet-50 for Imagenette.

				White-Box Attacks				Black-Box Attacks	
Dataset	**Track**	**Init**	**Clean**	**APGD-CE**	**APGD-DLR**	**FAB-T**	**ART-PGD**	**Square**	**SPSA**
CIFAR-10	ALF (Track A)	Random	93.10	35.92	25.15	49.27	50.96	26.67	68.81
	ALF (Track A)	Pretrained	87.13	45.94	44.77	49.44	47.10	53.53	66.46
	Joint (Track B)	Random	76.57	45.32	41.75	42.43	45.75	49.22	60.80
	Joint (Track B)	Pretrained	83.80	51.24	48.79	50.47	51.96	55.87	68.33
CIFAR-100	ALF (Track A)	Random	62.72	15.74	8.15	21.83	31.38	8.28	43.23
	ALF (Track A)	Pretrained	51.10	26.45	19.27	22.50	28.22	25.30	38.12
	Joint (Track B)	Random	55.60	23.75	20.74	22.42	24.42	26.16	38.03
	Joint (Track B)	Pretrained	56.86	27.28	23.94	25.80	27.68	28.40	39.95
Imagenette	ALF (Track A)	Random	89.32	59.11	57.48	59.08	59.85	73.30	85.58
	ALF (Track A)	Pretrained	87.90	54.39	53.10	55.72	55.36	67.85	84.41
	Joint (Track B)	Random	79.16	52.23	49.22	51.31	52.64	58.93	75.97
	Joint (Track B)	Pretrained	77.73	52.10	48.15	50.24	52.48	56.54	74.19

Two key observations emerge from Table 12. First, the relative ranking of AFA's two training methods reverses across evaluation protocols. Under fixed-step ART-PGD, Track A (ALF) appears to be the stronger method in five of six settings. However, under APGD-DLR (targeted)—which serves as the most stringent AutoAttack component across all checkpoints—Track B (joint training) achieves superior robustness across all WideResNet benchmarks (CIFAR-10 and CIFAR-100), while Track A retains an advantage only on Imagenette/ResNet-50.

Second, this discrepancy arises from the ALF setup: two of the six ALF runs exhibit severe gaps of more than 20% between their ART-PGD and APGD-DLR accuracies (25.8% on CIFAR-10 and 23.2% on CIFAR-100), whereas all six joint-training checkpoints differ by at most 4.3%. Because ALF keeps the contrastive pretrained encoder frozen and only retrains a linear head on small-input WideResNets, the inflated ART-PGD performance reflects a failure of fixed-step gradient attacks (i.e., gradient masking) rather than genuine adversarial robustness. In contrast, Track B (joint training) updates the full network end-to-end and exhibits consistent robustness across white-box and black-box attacks. Accordingly, for the comparisons in the main text (Table 1), we report Track B (joint training with clean-pretrained initialization), which provides an end-to-end full-network defense directly comparable to PGD-AT, TRADES, and RepAlign.

F Training Time Analysis

All experiments were conducted on a single NVIDIA RTX PRO 6000 Blackwell Max-Q GPU. Models across all datasets and methods were trained for 200 epochs using identical batch sizes (512 for CIFAR-10 and CIFAR-100; 128 for Imagenette) and standard adversarial training hyperparameters (20-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$). Table 13 reports the empirical average time per epoch and the full end-to-end training duration across all methods and datasets.

Table 13: **Training time comparison across datasets (200 epochs each).**

Dataset	Method	Per epoch	Full Training
CIFAR-10	Clean	25.1 s	1 h 24 m
	PGD	7 m 31 s	25 h 02 m
	TRADES	10 m 02 s	33 h 26 m
	CKA	7 m 02 s	23 h 25 m
	CKNNA	6 m 59 s	23 h 16 m
	HSIC	7 m 00 s	23 h 19 m
	Soft-kNN	6 m 59 s	23 h 15 m
CIFAR-100	Clean	24.3 s	1 h 21 m
	PGD	7 m 15 s	24 h 11 m
	TRADES	10 m 04 s	33 h 32 m
	CKA	7 m 00 s	23 h 21 m
	CKNNA	7 m 17 s	24 h 18 m
	HSIC	6 m 55 s	23 h 04 m
	Soft-kNN	7 m 00 s	23 h 20 m
Imagenette	Clean	25.2 s	1 h 24 m
	PGD	3 m 15 s	10 h 51 m
	TRADES	3 m 52 s	12 h 54 m
	CKA	3 m 09 s	10 h 30 m
	CKNNA	3 m 01 s	10 h 04 m
	HSIC	3 m 00 s	10 h 02 m
	Soft-kNN	3 m 04 s	10 h 14 m

Key Observations:

- **Efficiency vs. PGD-AT:** Our representation alignment methods (CKA, CKNNA, HSIC, Soft-kNN) maintain training throughput comparable to or slightly faster than standard PGD adversarial training (e.g., $\approx$ 7 m/epoch vs. 7 m 31 s/epoch on CIFAR-10).
- **Speedup vs. TRADES:** Compared to TRADES, our alignment formulations deliver a consistent $\sim$ 20%–30% training speedup across all three datasets (saving $\approx$ 10 hours on CIFAR-10/100 and $\approx$ 2.5–3 hours on Imagenette).