EMPIRICAL AND CERTIFIED ROBUSTNESS ACROSS PRE-TRAINING REGIMES IN TRANSFER LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transfer learning is a natural fit for safety-critical applications, where task-specific data is scarce but models have to be robust against adversarial perturbations. Robustness can be instilled during pre-training, during fine-tuning, or both, where robust pre-training is by far the most expensive of these. Yet it remains unclear when this expensive step is actually needed: prior work reaches opposing conclusions, and each study fixes a single adaptation mechanism while letting downstream data scale vary together with the target domain. In this paper, we take on a new comparative perspective to investigate empirical and certified robustness across pre-training regimes, fine-tuning objectives, adaptation mechanisms, and downstream data scales, under different adversary norms, also exposing trade-offs. Our findings show that the adaptation mechanism decides the answer, also reconciling prior work's settings and opposing conclusions: With a frozen backbone, robustness can only be inherited from pre-training, and with an updateable backbone, robust fine-tuning instils robustness while standard fine-tuning largely erases it. We further show that robust pre-training nonetheless gains value for empirical robustness as downstream data becomes scarce. Finally, a layer-wise representational analysis reveals that comparable robustness can arise from distinct internal representations.

1 INTRODUCTION

Deploying deep learning in safety-critical settings imposes two requirements at once: first, models must be trained on limited task-specific data, and second, they must be robust to adversarial perturbations. Transfer learning meets the first requirement, as it improves downstream accuracy (Huh et al., 2016), accelerates convergence (He et al., 2019), and is most valuable when downstream data is scarce (Kornblith et al., 2019b; Newell & Deng, 2020). The second requirement refers to model input changes that are deliberately chosen to degrade performance. These adversarial input changes have been demonstrated effective, e.g., against vision systems (Eykholt et al., 2018), malware detectors (Biggio et al., 2013), and language models (Zou et al., 2023). Whether transfer learning can also meet robustness against adversarial attacks is less clear. Adversarial robustness (hereafter, robustness) can be established during pre-training, fine-tuning (of a new head on the frozen backbone, i.e. linear probing, or of the full network), or both, at very different costs: standard pre-trained backbones are freely available, while robust pre-training costs several-fold to over an order of magnitude more compute (Mądry et al., 2017; Shi et al., 2021), and robust checkpoints exist for only a handful of threat models (Croce et al., 2020).

The choice is therefore where in the transfer-learning pipeline to incur the costs of robustness. Previous studies directly addressing this question yield different findings (Table 1, top row). Vaishnavi et al. (2024) find that robust fine-tuning of a standard or self-supervised ResNet-50 backbone comes close to a fully robust pipeline, and attribute this to the general utility of pre-training rather than the transfer of robust features. Kopetzki et al. (2025), sweeping eleven source-domain training procedures and three fine-tuning methods, find instead that the source regime dominates downstream robustness. The two studies overlap in task, architecture, and threat model, yet reach opposite conclusions. The same split runs through robust fine-tuning methods, most of which presuppose a robust backbone (Liu et al., 2023; Xu et al., 2024; Hua et al., 2024), while recent work targets standard ones (Ngnawé et al., 2026). We review earlier work on the transfer of robustness and on adaptation methods between linear probing and full fine-tuning in Appendix A.

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.

Table 1: Is robust pre-training necessary for downstream robustness? Columns give the adaptation mechanism crossed with the robustness type; rows vary the threat model and, in case of ℓ_∞, the downstream data scale. ✓ necessary, ✗ not necessary, ? not addressed by prior work. An arrow (? → ✓/✗) marks a cell this work resolves.

		Linear probing *(backbone frozen)*		**Full fine-tuning** *(backbone updated)*	
		Empirical	**Certified**	**Empirical**	**Certified**
ℓ_2		✓ *Kopetzki et al.*	? *not addressed*	✗ *Vaishnavi et al.*	✗ *Vaishnavi et al.*
ℓ_∞	High data	? → ✓	? → ✓	? → ✗	? → ✗
	Low data	? → ✓	? → ✓	? → ✓	? → ✗

Prior work cannot settle whether robust pre-training is necessary because four aspects remain uncontrolled or unexamined. (1) Each fixes a single adaptation mechanism: Vaishnavi et al. (2024) update the whole network, whereas Kopetzki et al. (2025) retrain only on top of frozen features, so neither can tell whether its conclusion holds under the other; Nern et al. (2023) compare mechanisms, but only for robust backbones fine-tuned with a standard objective. (2) Both use an ℓ_2 adversary with $\epsilon \leq 0.5$, while ℓ_∞ is the more prevalent threat model (Mądry et al., 2017; Li et al., 2023), and only Vaishnavi et al. (2024) address certification, via randomised smoothing. (3) Downstream data scale covaries with domain shift across target datasets, so whether robust fine-tuning can substitute for robust pre-training as data becomes scarce cannot be attributed to data availability alone. (4) Robust accuracy is the only outcome assessed, although robustness also leaves a recognisable signature in internal representations (Cianfarani et al., 2022) that allows attributing it to specific layers. These gaps leave open whether *robust pre-training is necessary for downstream robustness*, which we address through three research questions:

RQ1. Adaptation Mechanisms: How do linear probing and full-network fine-tuning compare in establishing empirical and certified ℓ_∞ robustness, with and without robust pre-training?

RQ2. Data Availability: How does the gap between robust fine-tuning of a standard backbone and a fully robust pipeline vary with downstream data availability at a fixed target domain?

RQ3. Internal Representations: Where in the network does robust fine-tuning take effect, and do pipelines reaching comparable robustness arrive at comparable internal representations?

To answer these questions, we treat the adaptation mechanism as an experimental variable rather than a fixed design choice: on image classification with convolutional backbones, we cross three pre-training regimes (none, standard, robust) with two fine-tuning objectives (standard, robust) and evaluate every configuration under both linear probing and full fine-tuning. We measure ℓ_∞ robustness both empirically, an upper bound, and through certification, a lower bound, so findings that hold under both do not depend on how robustness is measured. Synthetically reduced versions of a single dataset vary data availability at a fixed domain, and a layer-wise Centered Kernel Alignment (CKA) analysis of the same models locates where robustness arises. Table 1 summarises the resulting answers next to prior work. Our contributions are summarised as follows:

- We show that whether robust pre-training is necessary depends on the adaptation mechanism: with a frozen backbone, robustness can only be inherited from pre-training; with an updateable backbone, robust fine-tuning instils it, and standard fine-tuning largely erases it. This explains the opposing prior conclusions (RQ1).
- We extend the comparison to ℓ_∞ and to both empirical and certified robustness, and find that updating the backbone adds substantial empirical but little certified robustness (RQ1).
- Holding the target domain fixed, we show that the value of robust pre-training grows sharply as downstream data shrinks for empirical robustness, but barely changes under certification (RQ2).
- Using CKA, we find that robust fine-tuning acts mainly on the deeper layers, and that configurations with comparable robustness settle into distinct representations (RQ3).

Outline Section 2 provides the necessary background by formalising robust training, its empirical and certified evaluation, and representational similarity. Section 3 describes our experimental design

and its setup, leading to answer the research questions in Section 4. Section 5 discusses practical implications and limitations. A reproducibility package of the experiments is available online.[1]

2 BACKGROUND

2.1 ADVERSARIAL ROBUSTNESS IN TRANSFER LEARNING

Following Mądry et al. (2017), we formalise adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2015) through an (ℓ_p, ϵ)-adversary that may add any perturbation from the ϵ-ball $\Delta = \{\delta : \|\delta\|_p \leq \epsilon\}$ to an input $x \in \mathbb{R}^d$, so that the perturbed input lies in $x + \Delta$. For a classifier f_θ with loss $\mathcal{L}$ on data $(x, y) \sim \mathcal{D}$, robust training solves

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big[\max_{\delta\in\Delta} \mathcal{L}(f_\theta(x+\delta), y) \Big] \tag{1}$$

We call training on Equation 1 a *robust objective* and ordinary risk minimisation a *standard objective*. *Robust accuracy*, the fraction of test inputs classified correctly under every $\delta \in \Delta$, cannot be computed exactly for deep networks and is approximated either empirically or through certification.

Empirical robustness. Empirical methods approximate the inner maximisation with an attack, typically Projected Gradient Descent (PGD), which takes signed gradient-ascent steps and projects back onto Δ after each; adversarial training fits the model to the resulting examples (Mądry et al., 2017). A single attack can overestimate robustness, e.g., under poorly tuned step sizes or masked gradients (Athalye et al., 2018), which AutoAttack (Croce & Hein, 2020) mitigates with a parameter-free ensemble of gradient-based and gradient-free attacks. Since an attack can only show that a misclassifying perturbation exists, empirical robust accuracy is an upper bound.

Certified robustness. Certification instead proves that no δ in Δ changes the prediction, typically by bounding the network's outputs over the entire ϵ-ball $x + \Delta$ around the input. Exact verification is NP-complete for ReLU networks (Katz et al., 2017), so scalable verifiers are incomplete and certified robust accuracy is a lower bound. We use deterministic verifiers, whose guarantees hold for the model itself rather than, as in randomised smoothing (Cohen et al., 2019), for a noise-smoothed model with high probability. Interval Bound Propagation (IBP) (Gowal et al., 2018) passes element-wise bounds through the network at roughly the cost of two forward passes, but its bounds are loose; CROWN (Zhang et al., 2018) tightens them through linear relaxations of the activation functions, at considerably higher cost. Certified training minimises a loss on such bounds instead of the inner maximisation (Wong & Kolter, 2018; Gowal et al., 2018). CROWN-IBP (Zhang et al., 2019) weights the clean loss against this bound-based loss with a coefficient κ and mixes IBP and CROWN bounds with a coefficient β; as bounds on an untrained network are too loose to learn from, ϵ is ramped up from zero while β is annealed to zero, leaving the final phase trained on IBP bounds alone.

Robustness transfer. In transfer learning, a backbone is pre-trained on a source dataset and reused for a target task with a fresh head. For either stage, a standard or robust objective may be used. Adaptation mechanisms differ in the parameters fine-tuning may update: *linear probing* freezes the backbone and trains only the head, while *full fine-tuning* updates all parameters, with the pre-trained weights serving only as an initialisation. Under linear probing, downstream robustness is constrained by that of the frozen representation (Nern et al., 2023); under full fine-tuning, the backbone can acquire robustness during adaptation but can also lose the inherited robustness. The two mechanisms thus sit on opposite sides of the stability–plasticity dilemma (Abraham & Robins, 2005).

2.2 REPRESENTATIONAL SIMILARITY

Centered Kernel Alignment (CKA) (Kornblith et al., 2019a) compares two layers' representations of the same set of inputs by how each arranges the inputs relative to one another rather than by matching individual neurons, so it applies across layer widths and, being invariant to orthogonal transformations and isotropic scaling, across separately trained networks. Computed for every layer pair, it yields a similarity matrix, whose diagonal, between two networks of the same architecture,

[1] `anonymous.4open.science/r/robust-transfer-regimes-3C60`.

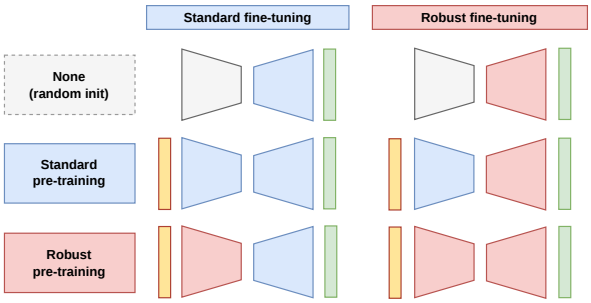


Figure 1: The six configurations: pre-training regime (rows) crossed with fine-tuning objective (columns). Each cell shows the backbone (left), coloured by its pre-training objective on source data, and the head (right), coloured by the fine-tuning objective on target data; blue denotes standard training, red robust training, and grey random initialisation.

Table 2: Configuration names. The two controls are evaluated under full fine-tuning only.

Configuration	Pre-train	Fine-tune
Standard Control	None	Standard
Standard Baseline	Standard	Standard
Direct Transfer	Robust	Standard
Robust Control	None	Robust
Robust Adaptation	Standard	Robust
Fully Robust	Robust	Robust

indicates how far matched-depth representations have drifted; within a network, the matrix reveals its internal organisation. Standard residual networks show a block structure of high similarity within stages, while adversarially trained networks show elevated similarity across distant layers, a reduction in layer specialisation that Cianfarani et al. (2022) identify as a signature of robust training.

3 Experimental Design

3.1 Study Design

Because certified training does not yet scale to ImageNet at full resolution, we study the two robustness types in separate tracks that differ in architecture, source data, and resolution (Section 5), and compare training recipes while holding all other factors constant within each track. Crossing three pre-training regimes with two fine-tuning objectives yields the six configurations in Table 2, illustrated in Figure 1. Standard Baseline, the conventional pipeline, captures any incidental robustness gain from pre-trained features. Direct Transfer shows whether robustness survives the transfer-learning pipeline without being optimised for, Robust Adaptation what a robust objective can instil in a standard backbone, and Fully Robust induces robustness at both stages. The two controls omit pre-training altogether and are only trained from scratch, setting the robustness achievable without it; linear probing them would merely fit a linear classifier to random features.

Evaluating both adaptation mechanisms addresses RQ1 and fills the ℓ_∞ rows of Table 1: Robust Adaptation versus Fully Robust shows whether robust pre-training adds anything once fine-tuning is robust, and Direct Transfer versus Fully Robust whether pre-trained robustness survives a standard objective. For RQ2, we track the former gap as a single target is reduced to 10% and 1% of its training set, at a fixed domain and test set. For RQ3, we compare each fine-tuned model with its backbone to locate where fine-tuning acts, and Robust Adaptation with Fully Robust to test whether equal robustness implies similar representations.

3.2 Threat model

We consider a test-time evasion adversary whose goal is to cause misclassification of individual inputs, with no target class specified (untargeted) (Biggio et al., 2013). The adversary is white-box: it has full access to the architecture, trained parameters, and gradients of the model, which is the worst case for the defender and the standard setting for both certified robustness and strong empirical attacks. Its capabilities are bounded by an (ℓ_p, ϵ)-adversary with $p = \infty$, at $\epsilon = 4/255$ on the empirical track and $\epsilon = 1/255$ on the certified track. The empirical budget matches the robust ImageNet weights of Salman et al. (2020) and the standard ℓ_∞ budget for ImageNet (Croce et al., 2020); the certified budget is the standard and currently highest level attainable at this scale (Xu et al., 2020; Shi et al., 2021). Explicitly out of scope are training-time attacks (poisoning, backdoors),

perturbations outside the ℓ_∞ ϵ-ball (common corruptions, spatial transforms, patch and unrestricted attacks), and variation in ϵ itself.

Both budgets differ from the ℓ_2 adversaries with $\epsilon \leq 0.5$ used in prior studies of robust transfer (Kopetzki et al., 2025; Vaishnavi et al., 2024). Since ℓ_2 and ℓ_∞ balls are not nested, the adversaries cannot be ranked: our empirical ℓ_∞ ϵ-ball admits perturbations spread over all pixels with an ℓ_2 norm of up to ≈ 6.09, at least an order of magnitude beyond these budgets, while their ℓ_2 ϵ-balls admit perturbations concentrated on single pixels far beyond our per-pixel limit (Appendix B). We therefore compare our findings with prior work at the level of conclusions rather than robust accuracy values (Section 4.1).

3.3 Setup

Models and data. The empirical track uses ResNet-50 at 224×224 with the torchvision `IMAGENET1K_V2` checkpoint as the standard backbone and `Salman2020Do_R50` from RobustBench, adversarially trained on ImageNet at $\epsilon = 4/255$ (Salman et al., 2020), as the robust one. The certified track uses a Wide ResNet (7.9M parameters) at 56×56. As no public certified-robust weights exist, we pre-train it ourselves on Tiny ImageNet (Russakovsky et al., 2015), with cross-entropy for the standard backbone and with CROWN-IBP via `auto_LiRPA` (Xu et al., 2020) for the certified one. On their source datasets, the backbones reach 80.86% clean accuracy (standard ResNet-50), 64.02% clean and 34.96% robust (robust ResNet-50), 55.10% clean (standard Wide ResNet), and 29.76% clean and 15.90% certified (certified Wide ResNet).

The five downstream tasks are CIFAR-10 (5,000 per class), Caltech-101 and Stanford Cars (tens to hundreds per class, with substantial class imbalance), and CIFAR-10 10% and 1% reductions (500 and 50 per class). Augmentation is limited to random crops and horizontal flips; dataset statistics and preprocessing are given in Appendices C.1 and C.2. Budgets ϵ are given in raw $[0, 1]$ pixel space on both tracks. On the empirical track, normalisation is the model's first layer, following the RobustBench convention, so the attack perturbs unnormalised inputs. On the certified track, bound propagation requires normalisation to remain in the pipeline, so we construct the bounds as $x \pm \epsilon/\sigma$ and clip them to the normalised image domain.

Training procedure. Standard and empirical robust training use SGD; certified training uses Adam (Zhang et al., 2019). Empirical robust fine-tuning uses PGD with 8 steps, a step size of $\epsilon/4$, and one restart, at a fixed $\epsilon = 4/255$ with no ϵ schedule. Certified fine-tuning uses CROWN-IBP with the ϵ-warmup expressed in fractions of the run length.

Hyperparameters are tuned separately for each combination of configuration, target task, and adaptation mechanism with Bayesian Optimisation, using 20 trials for standard and 32 for robust fine-tuning (Appendix C.4). Both the search and early stopping maximise the validation objective $\mathrm{Acc}_{\mathrm{clean}} + 1.8 \cdot \mathrm{Acc}_{\mathrm{rob}}$. We tune once under a single seed, then retrain the selected configuration under three seeds, varying head initialisation, data ordering, augmentation sampling, and attack randomness, with the pre-trained backbones fixed. All reported figures are means over the three runs with the sample standard deviation, reflecting training stochasticity rather than tuning variance.

Robustness evaluation. Empirical robust accuracy is measured with AutoAttack (Croce & Hein, 2020) at $\epsilon = 4/255$ over the full test set of each task. Certified robust accuracy is obtained with IBP through `auto_LiRPA` (Xu et al., 2020) at $\epsilon = 1/255$, also over the full test set. Since CROWN-IBP training ends on pure IBP bounds (Section 2), IBP certifies the models with the bounds they were trained on, at substantially lower cost than CROWN.

Representational analysis. Layer-wise similarity is computed with linear CKA (Subramanian, 2021). Following Cianfarani et al. (2022), activations are extracted at all layers, including pooling, activation, and normalisation layers, from benign inputs of the CIFAR-10 10% test split, matching the fine-tuning target. Comparisons between two models are read on the matched-depth diagonal of the similarity matrix; comparisons of a model with itself use the full matrix. As CKA values have no absolute scale (Kornblith et al., 2019a), and a low diagonal value can reflect a benign shift rather than overwritten features (Davari et al., 2022), we calibrate against seed variation and inspect full matrices. As a noise floor, we compute the mean diagonal CKA for each configuration over all

Table 3: Robust accuracy (%) of every configuration, grouped by adaptation mechanism. Empirical: AutoAttack, ℓ_∞, $\epsilon = 4/255$, ResNet-50. Certified: IBP, ℓ_∞, $\epsilon = 1/255$, Wide ResNet. Best per column within each mechanism in **bold**. Clean accuracies and standard deviations in Appendix D.

	Empirical					Certified				
	CIFAR-10					CIFAR-10				
Configuration	100%	10%	1%	Caltech	Cars	100%	10%	1%	Caltech	Cars
Linear probing										
Standard Baseline	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
Direct Transfer	36.05	35.92	32.86	82.11	11.41	26.76	27.44	25.12	43.05	2.86
Robust Adaptation	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fully Robust	**46.03**	**42.30**	**35.55**	**88.38**	**14.14**	**34.90**	**33.40**	**25.90**	**45.92**	**3.10**
Full fine-tuning										
Standard Control	0.00	0.00	3.64	5.84	0.00	0.00	0.00	0.00	0.00	0.00
Standard Baseline	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Robust Control	63.10	42.91	18.83	81.40	14.35	**46.65**	36.70	24.99	54.10	**4.98**
Direct Transfer	0.00	0.02	0.39	64.05	0.01	0.00	0.00	0.00	0.00	0.00
Robust Adaptation	57.53	35.34	0.02	91.72	1.04	43.04	36.68	18.10	**55.18**	0.00
Fully Robust	**68.19**	**57.30**	**41.46**	**95.52**	**64.56**	43.42	**36.72**	**26.51**	51.39	4.06

three pairs of its seeded runs, so similarities materially below that value are read as representational differences rather than seed variation.

4 RESULTS

Table 3 reports robust accuracy for every configuration, adaptation mechanism, and downstream task on both tracks; clean accuracies and standard deviations are in Appendix D.

4.1 RQ1: ADAPTATION MECHANISMS

A frozen backbone passes on robustness but cannot gain it. Under linear probing, Robust Adaptation (standard backbone, robust objective) reaches at most 0.01% robust accuracy on any target in either track. A robust objective applied to the head alone cannot instil robustness, consistent with the bound of Nern et al. (2023), under which a linear head is no more robust than the features beneath it. Direct Transfer (robust backbone, standard objective) retains 11.41–82.11% empirical and 2.86–43.05% certified robust accuracy, within 10 points of Fully Robust on every target, so a standard trained head does not remove pre-trained robustness.

An updateable backbone gains robustness but also forgets it. Under full fine-tuning, both results reverse. With a standard objective, Direct Transfer's robust accuracy falls to $\leq 0.39\%$ on four of five empirical targets and to 0.00% on all five certified targets, although its clean accuracy rises above the linear-probing level on nine of the ten target-track pairs (Appendix D). This holds even though model selection rewards robust accuracy, and mirrors Yamada & Otani (2022), who find that corruption robustness acquired on ImageNet is partially preserved downstream under a frozen backbone but not retained after full fine-tuning on CIFAR-10. Only empirical Caltech-101, plausibly the target closest to the ImageNet source, retains robustness (64.05%). With a robust objective, Robust Adaptation instils substantial robustness in a standard backbone on three of five empirical (35.34–91.72%) and four of five certified targets (18.10–55.18%). It fails on empirical CIFAR-10 1% (Section 4.2), and on Stanford Cars, the hardest target, where no certified configuration exceeds 5% and the empirical run does not converge (1.26% clean accuracy). We attribute the non-convergence to robust fine-tuning at a fixed ϵ from a non-robust initialisation, the suboptimal transfer that Ngnawé et al. (2026) mitigate with ϵ-scheduling; we kept one recipe for all configurations instead.

The value of updating the backbone depends on the robustness type. For Fully Robust, full fine-tuning adds 5.91–50.42 points (mean 20.13) of empirical robust accuracy over linear probing, but only 0.61–8.52 (mean 3.78) of certified robust accuracy. The empirical gain is largest on Stan-

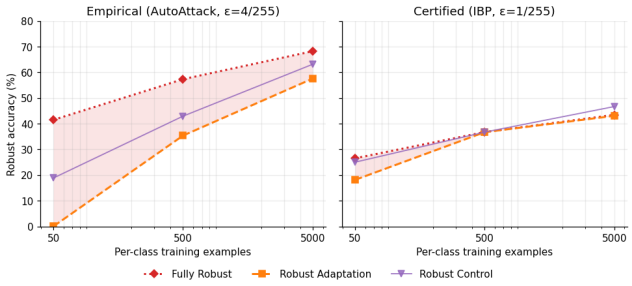


Figure 2: Robust accuracy (%) under full fine-tuning on the CIFAR-10 reductions (5,000, 500, and 50 training examples per class; fixed target domain and test set). The shaded band marks the gap between Robust Adaptation (standard pre-training) and Fully Robust (robust pre-training).

ford Cars (from 14.14 to 64.56%), the fine-grained target furthest from the source domain, and smallest on CIFAR-10 1%, which has the least data to adapt on. On the certified track, a frozen robust backbone thus already recovers most of the robustness full fine-tuning reaches, while on the empirical track, adapting the feature extractor to the target pays off.

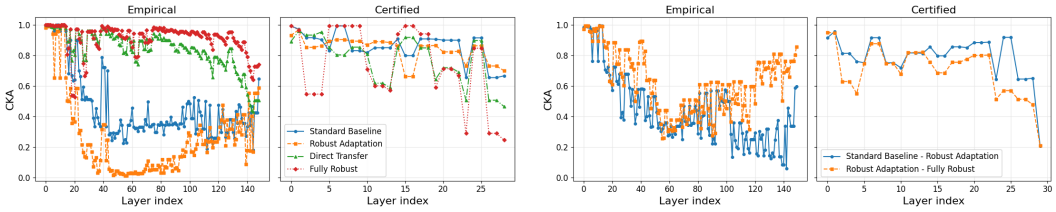
The adaptation mechanism selects between stability and plasticity. A frozen backbone keeps its robustness but cannot gain more, whereas an updateable one can gain robustness but may also lose it, so the mechanism selects which horn of the stability–plasticity dilemma applies. This offers an explanation for the opposite conclusions of prior work, reached under ℓ_2 adversaries, since each study fixes a different adaptation mechanism. Retraining only on top of frozen features, Kopetzki et al. (2025) conclude that the source regime dominates downstream robustness, in line with our finding that robust pre-training is necessary. Since we reach this under a different, non-nested adversary (Section 3.2), the agreement extends their conclusion to a threat model closer to the canonical ℓ_∞ setting. Updating the whole network, Vaishnavi et al. (2024) conclude that robust fine-tuning of a standard backbone comes close to a fully robust pipeline. We recover their conclusion only in part, and not their explanation: on full CIFAR-10, Robust Adaptation matches Fully Robust on the certified track but trails it by 10.66 points on the empirical track, and on the CIFAR-10 reductions it gains nothing over training from scratch (Section 4.2). As both studies update the whole network, the adaptation mechanism does not account for this difference, which may partly reflect the distinct failure modes that ℓ_2 and ℓ_∞ adversaries cover (Tramer & Boneh, 2019).

Answer to RQ1: With a frozen backbone, downstream robustness can only be inherited from pre-training, and a standard head does not remove it. With an updateable backbone, robust fine-tuning instils robustness in a standard backbone on most targets, whereas standard fine-tuning erases it. Updating the backbone adds substantial empirical gains but little to certified robustness.

4.2 RQ2: Data Availability

Robust pre-training gains value as data shrinks on the empirical track. In Figure 2, the empirical gap between Robust Adaptation and Fully Robust widens monotonically as data shrinks, from 10.66 points at 5,000 examples per class to 21.96 at 500 and 41.44 at 50. At the smallest scale, Robust Adaptation still trains (49.37% clean accuracy) but acquires no robustness (0.02%), while Fully Robust holds 41.46%. Relative to the Standard Baseline, Robust Adaptation also loses 10.59–35.81 points of clean accuracy, more as data shrinks, whereas Fully Robust loses only 4.67–6.17. This matches the suboptimal transfer that Ngnawé et al. (2026) report for fixed-ϵ robust fine-tuning of standard backbones; here it grows with data scarcity, and a robust backbone largely avoids it. On the certified track, the gap is smaller than the standard deviation at 5,000 and 500 examples per class (0.38 and 0.04 points) and widens only at the smallest scale (8.41 points), reaching a fifth of the empirical gap at the same scale.

Robust pre-training beats training from scratch only on the empirical track. Against Robust Control, Fully Robust gains 5.09, 14.39, and 22.63 points of empirical robust accuracy as data shrinks, and 14.12 and 50.21 points on Caltech-101 and Stanford Cars; it is the best configuration in



(a) Each fine-tuned configuration against its own pre-trained backbone.

(b) Fine-tuned configurations against each other.

Figure 3: Layer-wise diagonal CKA on CIFAR-10 10% for the empirical (left) and certified (right) tracks, under full fine-tuning. Seed floors are given in Table 10 (Appendix D).

all empirical rows of Table 3, in line with the gains Hendrycks et al. (2019) report for adversarial pre-training over adversarial training from scratch. For certified robustness, Fully Robust stays within -3.23 to $+1.52$ points of Robust Control on every target, and Robust Control is best on CIFAR-10 and Stanford Cars.

Standard pre-training alone adds no robustness on the CIFAR-10 reductions. Because Robust Adaptation also starts from a pre-trained backbone, Figure 2 directly tests the attribution by Vaishnavi et al. (2024) of robust-backbone gains to the general utility of pre-training, here under ℓ_∞ and with one recipe for all configurations. On every CIFAR-10 reduction and on both tracks, Robust Adaptation stays at or below Robust Control, so standard pre-training adds no robustness over training from scratch, while robust pre-training adds up to 22.63 points on the empirical track.

Answer to RQ2: On the empirical track, the advantage of robust over standard pre-training grows as downstream data shrinks, until robust fine-tuning of a standard backbone yields no robustness and robust pre-training becomes necessary. On the certified track, the advantage stays small at every scale, and robust training from scratch performs on par with the fully robust pipeline, with neither consistently ahead.

4.3 RQ3: Internal Representations

We analyse the four configurations fully fine-tuned on CIFAR-10 10%, a reduced data scale at which Robust Adaptation and Fully Robust match on certified robust accuracy to within 0.04 points. As a noise floor, the mean diagonal CKA between seeds (Table 10 in Appendix D.2) is near one only for empirical Direct Transfer (0.972) and Fully Robust (0.988) and lies between 0.768 and 0.891 for the other six configurations. We read a matched-layer similarity as a representational difference only when it falls clearly below the applicable floor, taking the lower of the two floors when comparing two fine-tuned models.

Robust fine-tuning leaves a robust backbone largely intact but reshapes a standard one. Against its own backbone (Figure 3a), empirical Fully Robust stays near 0.90 over most of the depth and falls to roughly 0.75 over the final ~ 20 layers. The standard backbones instead change from early on: Standard Baseline and Robust Adaptation drop within the first ~ 30 layers to roughly 0.25–0.5 and below 0.25, respectively, far below their floors, and the full matrices show no off-diagonal band that would indicate a benign depth shift (Davari et al., 2022) (Appendix D.2). Direct Transfer stays close to its backbone through most of the network on both tracks and diverges mainly in the deeper layers, while retaining at most 0.02% robust accuracy (Table 3); the erasure under a standard objective thus sits mainly in the deeper layers. On the certified track, Standard Baseline and Robust Adaptation do not fall far below their floors, unlike on the empirical track, so we cannot separate their changes from seed variation. Fully Robust drops furthest, to 0.25–0.30 over the final layers, with smaller dips earlier in the network. So even an already robust representation still adapts to the target.

Matching certified robustness arises from distinct deep representations. Comparing the fine-tuned models directly (Figure 3b), empirical Standard Baseline and Robust Adaptation fall below

0.4 in the deeper layers, far below the pair's 0.774 floor. Robust Adaptation and Fully Robust drop to 0.30–0.60 through the mid-network, but reconverge to 0.70–0.85 over the final layers, within seed variation, consistent with robustness acting there, although their robust accuracy differs by 21.96 points. The certified track, where the two match to within 0.04 points, provides the direct test, and the ordering reverses: Robust Adaptation and Fully Robust fall to about 0.50–0.57 across the deepest quarter of the network, below the pair's floor of 0.768, while Standard Baseline and Robust Adaptation stay at or above it over most of the depth. The robustly fine-tuned standard backbone thus stays closer to its starting point than to the fully robust model it matches on accuracy.

Within-model CKA matrices (Appendix D.2) show on both tracks the reduced layer specialisation that Cianfarani et al. (2022) identify as a signature of robust training: the block structure of the Standard Baseline is largely lost in Robust Adaptation and Fully Robust.

Answer to RQ3: On both tracks, robust fine-tuning changes a robust backbone most strongly in its final layers. A standard backbone is reshaped from early layers on under empirical robust fine-tuning, while under certified robust fine-tuning its change cannot be separated from seed variation. Matching certified robustness can arise from measurably distinct deep representations.

5 Conclusion

In this paper, we addressed the question whether robust pre-training is necessary for downstream robustness, treating the adaptation mechanism as an experimental variable and evaluating both empirical and certified robustness against an ℓ_∞ adversary. We found that the answer depends on what the adaptation mechanism may update. With a frozen backbone, robustness can only be inherited from pre-training. With an updateable backbone, a robust objective instils it, and a standard objective erases it. This dependence also explains why prior studies, each fixing a single mechanism, reached opposite conclusions. What robust pre-training adds beyond that depends on the robustness type and on the downstream data: for empirical robustness its value grows as downstream data becomes scarce, until it is once more necessary, while for certified robustness, it stays small at every scale we test. Comparable robustness, finally, need not mean comparable representations, as two pipelines that match on certified robust accuracy differ measurably in their deepest layers.

In practice, the cost of robust pre-training is best justified in two cases: when the backbone will be frozen, and when empirical robustness is needed with scarce downstream data. Otherwise, robust fine-tuning of a freely available standard backbone is a viable alternative, provided the recipe suits a standard initialisation. On the certified track, at the scale certified training currently reaches, robust pre-training buys little over robust training from scratch.

Limitations and future work. We provided the first comparative study of empirical and certified robustness varying parameters and tracks. Therefore, more extensive studies can naturally emerge, exceeding the limitations of our setup. Our comparisons were controlled within each track; across tracks, we compared the direction and ordering of effects rather than their magnitudes, since certified training does not scale to the empirical setting and the two tracks thus differ in architecture, source data, resolution, attack budget, and attainable clean accuracy (Section 3.3). Certified methods that reach the scale of the empirical track would also allow other magnitudes of comparisons. Our backbones are all convolutional and all targets are image classification; each track uses a single supervised checkpoint, and the certified backbones are pre-trained on Tiny ImageNet, a much smaller source than is common in practice. We also omitted self-supervised pre-training, under which robust pre-training and robust fine-tuning each contribute to downstream robustness (Chen et al., 2020; Jiang et al., 2020). Vision Transformers may behave differently, as they need standard pre-training to be adversarially trainable (Mo et al., 2022), leaving our randomly initialised controls without a counterpart. Finally, we treated adaptation as a binary choice between a frozen and a fully updated backbone; staged partial unfreezing and parameter-efficient methods such as LoRA (Hu et al., 2021) lie in between and may resolve the stability–plasticity trade-off differently. Extending our studies to these mechanisms and to transformer backbones is the natural next step.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on our research methodology and experimental design, and to assist in implementing methods. We have not used generative AI tools to propose or refine hypotheses, develop formal models or conceptual frameworks, generate synthetic datasets, clean or reformat datasets, or interpret results. Formulating mathematical claims, providing ingredients for or writing proofs, translation, and qualitative or thematic data analysis do not apply to this work. We also note that generative AI is not part of our proposed method or experimental pipeline; we used it only to assist the authors during the research and writing process. Additionally, we used generative AI tools for brainstorming, identifying research topics and gaps, sourcing information and relevant literature, summarising existing literature, proposing experimental parameters, generating software code, and editing the paper to improve readability.

We have reviewed all AI-assisted work. Throughout the project, we used generative AI as a tool for discussion and iteration, while the authors made all research directions, design decisions, interpretations and conclusions. We treated feedback on our methodology and experimental design as suggestions, evaluated them against our research goals, and adopted them only when the authors judged them sound. One author reviewed the AI-assisted code, including method implementations, and validated it through a combination of unit tests and sanity checks against known results. We treated suggested experimental parameters as starting points and tuned them using the methods described. We verified literature identified or summarised with AI assistance against the sources, and checked every citation to confirm it exists and supports the claim attributed to it. We checked proposed ideas against a manual literature survey to confirm their novelty and avoid unintentional plagiarism. We reviewed readability edits to ensure they did not alter the paper's technical content or claims. We take responsibility for the final content of this work, including text, claims, or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All code needed to reproduce our results is available in an anonymised repository at `anonymous.4open.science/r/robust-transfer-regimes-3C60`. Section 3.3 specifies the models, datasets, training procedure, and evaluation protocol. The empirical track builds on publicly released checkpoints (torchvision `IMAGENET1K_V2` and RobustBench `Salman2020Do_R50`); the certified-track backbones, which we pre-train on Tiny ImageNet, follow the schedule in Appendix C.3 and will be released upon acceptance. All datasets are public. Appendix C.1 describes the class-stratified construction of the CIFAR-10 reductions and validation splits, and Appendix C.2 describes the preprocessing of every target. Appendix C.3 gives the full training schedules, including the ϵ-warmup of certified training, and Appendix C.4 gives the hyperparameter search, whose search spaces are defined in the code. Tables 6 and 7 (Appendix C.5) list the selected hyperparameters for every configuration, target, and adaptation mechanism, so our results can be reproduced without repeating the search. All reported figures are means over three seeds, with standard deviations in Tables 8 and 9 (Appendix D) and CKA seed floors in Table 10 (Appendix D.2), against which reruns can be compared.

REFERENCES

Wickliffe C Abraham and Anthony Robins. Memory retention–the synaptic stability versus plasticity dilemma. *Trends in neurosciences*, 28(2):73–78, 2005.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

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.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL http://arxiv.org/abs/1910.06403.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 696–705. IEEE, 2020.

Christian Cianfarani, Arjun Nitin Bhagoji, Vikash Sehwag, Ben Zhao, Heather Zheng, and Prateek Mittal. Understanding robust learning through the lens of representation similarities. *Advances in Neural Information Processing Systems*, 35:34912–34925, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

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

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of cka as a similarity measure in deep learning. *arXiv preprint arXiv:2210.16156*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, 2009.

Zhun Deng, Linjun Zhang, Kailas Vodrahalli, Kenji Kawaguchi, and James Zou. Adversarial training helps transfer learning via better representations. *Advances in Neural Information Processing Systems*, 34:25179–25191, 2021.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Mądry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4918–4927, 2019.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Andong Hua, Jindong Gu, Zhiyu Xue, Nicholas Carlini, Eric Wong, and Yao Qin. Initialization matters for adversarial transfer learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24831–24840. IEEE, 2024.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great in high dimensions. *arXiv preprint arXiv:2402.02229*, 2024.

Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in neural information processing systems*, 33:16199–16210, 2020.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pp. 97–117. Springer, 2017.

Anna-Kathrin Kopetzki, Aleksandar Bojchevski, and Stephan Guennemann. Evaluating the transferability of adversarial robustness to target domains. *Knowledge and Information Systems*, 67 (5):4139–4206, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019a.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2661–2671, 2019b.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Linyi Li, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. In *2023 IEEE symposium on security and privacy (SP)*, pp. 1289–1310. IEEE, 2023.

Ziquan Liu, Yi Xu, Xiangyang Ji, and Antoni B Chan. Twins: A fine-tuning framework for improved transferability of adversarial robustness and generalization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16436–16446. IEEE, 2023.

Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. *Advances in Neural Information Processing Systems*, 35:18599–18611, 2022.

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

Laura F Nern, Harsh Raj, Maurice André Georgi, and Yash Sharma. On transfer of adversarial robustness from pretraining to downstream tasks. *Advances in neural information processing systems*, 36:59206–59226, 2023.

Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7345–7354, 2020.

Jonas Ngnawé, Maxime Heuillet, Sabyasachi Sahoo, Yann Pequignot, Ola Ahmad, Audrey Durand, Frédéric Precioso, and Christian Gagné. Robust fine-tuning from non-robust pretrained models: Mitigating suboptimal transfer with epsilon-scheduling. In *International Conference on Learning Representations*, volume 2026, pp. 47457–47477, 2026.

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. *arXiv preprint arXiv:2010.15327*, 2020.

Miles Olson, Elizabeth Santorella, Louis C. Tiao, Sait Cakmak, Mia Garrard, Samuel Daulton, Zhiyuan Jerry Lin, Sebastian Ament, Bernard Beckerman, Eric Onofrey, Paschal Igusti, Cristian Lara, Benjamin Letham, Cesar Cardoso, Shiyun Sunny Shen, Andy Chenyuan Lin, Matthew Grange, Elena Kashtelyan, David Eriksson, Maximilian Balandat, and Eytan. Bakshy. Ax: A Platform for Adaptive Experimentation. In *AutoML 2025 ABCD Track*, 2025.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Mądry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Fast certified robust training with short warmup. *Advances in Neural Information Processing Systems*, 34:18335–18349, 2021.

Anand Subramanian. torch_cka. `https://github.com/AntixK/PyTorch-Model-Compare`, 2021.

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

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32, 2019.

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

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better: Illustration on image classification. *arXiv preprint arXiutrera2021adversariallyv:2007.05869*, 2020.

Pratik Vaishnavi, Kevin Eykholt, and Amir Rahmati. A study of the effects of transfer learning on adversarial robustness. *Transactions on Machine Learning Research*, 2024.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. *arXiv preprint arXiv:2102.07188*, 2021.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295. PMLR, 2018.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

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

Yutaro Yamada and Mayu Otani. Does robustness on imagenet transfer to downstream tasks? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9215–9224, 2022.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems*, 31, 2018.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Extended related work

Transfer of adversarial robustness Hendrycks et al. (2019) were among the first to show that pre-training improves downstream adversarial robustness: adversarially pre-training on a downsampled ImageNet and then adversarially fine-tuning the entire network surpassed the prior state of the art on CIFAR-10 and CIFAR-100 against an ℓ_∞ adversary at $\epsilon = 8/255$. Shafahi et al. (2019) showed that robust networks contain robust feature extractors, so that a classifier trained on natural examples on top of a frozen robust backbone inherits much of its robustness without any adversarial training, although not always without a loss of downstream standard accuracy. Salman et al. (2020), whose robust weights we use, found that robust ImageNet backbones also improve downstream standard accuracy under both fixed-feature and full-network transfer, even though adversarial training lowers accuracy on the source task itself (Tsipras et al., 2018), and relate this to the more perceptually aligned representations that adversarial training induces (Engstrom et al., 2019); further studies confirm that adversarially trained networks transfer better (Utrera et al., 2020; Deng et al., 2021). Under self-supervised pre-training, robust pre-training both raises final robustness and reduces the cost of subsequent adversarial fine-tuning (Chen et al., 2020), and adversarial contrastive learning improves on it further (Jiang et al., 2020). The transfer has boundary conditions. Adversarial fine-tuning of a robust backbone outperforms standard fine-tuning in most low-data regimes, but not in the smallest (Shafahi et al., 2019). For corruption robustness, CNNs robustified on ImageNet do not retain their robustness when fully fine-tuned on CIFAR-10, while for detection and segmentation a standard Swin Transformer transfers more robustness than such CNNs, making architecture a strong source of transferred robustness in its own right (Yamada & Otani, 2022).

Is robust pre-training necessary? Whether robust pre-training still matters once fine-tuning is itself robust has been answered in both directions. Hendrycks et al. (2019) already report that pre-training without an adversary does not provide substantial robustness in their setting. Vaishnavi et al. (2024) reach the opposite conclusion under an ℓ_2 adversary at $\epsilon = 0.5$. They pre-train a ResNet-50 on ImageNet robustly or with SimCLR, robustly fine-tune the full network on twelve downstream tasks, and measure both empirical robustness and certified robustness via randomised smoothing (Cohen et al., 2019) trained with consistency regularisation (Jeong & Shin, 2020). Both forms of pre-training improve robustness over training from scratch on most targets, and a direct comparison on CIFAR-10 and CIFAR-100, which adds cross-entropy pre-training, places the non-robust backbones close behind the robust one. They attribute these gains to pre-training's effect on standard accuracy rather than to transferring robust features: relative gains in standard and robust accuracy are strongly linearly correlated across targets, and where pre-training lowers standard accuracy, as on Birdsnap and Stanford Cars, robust accuracy also falls below training from scratch. They further find that standard fine-tuning of the full robust network yields neither empirical nor certified robustness, and that standard retraining of the final layer alone, which preserves empirical robustness (Shafahi et al., 2019), does not preserve certified robustness. Kopetzki et al. (2025) instead cross eleven source-domain training procedures, spanning standard, adversarial, contrastive, and Lipschitz-constrained training, with three retraining schemes on the target domain, each operating on top of the frozen source representation, and attack both representations and predictions at ℓ_2 budgets of 0.05 and 0.1 on 32×32 inputs. They find that the source regime dominates downstream robustness while target retraining has only a minor effect, that the most robust source model is rarely the most robust target model, and that methods enforcing robustness independently of the training data, such as Lipschitz constraints, transfer most reliably. Nern et al. (2023) account for the frozen-feature case theoretically, proving that the robustness of a linear predictor on any downstream task is bounded by the robustness of the representation beneath it, whatever the pre-training protocol. Section 1 sets out how these designs differ from ours: each fixes a single adaptation mechanism, both use ℓ_2 adversaries that cannot be ranked against ours (Appendix B), and Vaishnavi et al. (2024) vary data availability only across targets, so that it changes together with the domain.

Adaptation beyond the binary choice and promoting robustness Between linear probing and full-network fine-tuning lie staged and parameter-efficient methods. For standard transfer, Kumar et al. (2022) show that fine-tuning under a randomly initialised head distorts pre-trained features, and propose LP-FT, which trains the head first to limit this distortion. Most robust fine-tuning methods start from a robust backbone and aim to retain its robustness while adapting. Shafahi et al. (2019) regularise end-to-end fine-tuning with a lifelong-learning objective that keeps features close

to those of the source network, preserving robustness without adversarial training. TWINS (Liu et al., 2023) keeps the batch-normalisation statistics of robust pre-training alongside adaptive ones, which also raises the effective learning rate and speeds the escape from the pre-trained initialisation. AutoLoRa (Xu et al., 2024), having found that natural and adversarial objectives produce divergent gradients when both act on the feature extractor, routes the natural objective through an auxiliary low-rank branch. Robust linear initialisation (Hua et al., 2024) initialises the head by adversarial linear probing before adversarial fine-tuning; the same study finds, for vision transformers, that parameter-efficient methods applied to a standard backbone fail to become robust, or remain far less robust, even under adversarial fine-tuning, and that with a robust backbone linear probing can outperform full fine-tuning under a randomly initialised head on some targets. Ngnawé et al. (2026) instead fine-tune standard backbones robustly, and find that a robust objective at a fixed ϵ impedes task adaptation early in training, degrading performance and, on difficult tasks or at large ϵ, causing transfer to fail altogether; scheduling ϵ over training mitigates this suboptimal transfer, which we also observe (Sections 4.1 and 4.2). We treat adaptation as a binary choice and leave these intermediate mechanisms to future work (Section 5).

Certified robustness in transfer Deterministic certified training has been developed and benchmarked almost exclusively by training from scratch on a single dataset. IBP (Gowal et al., 2018) and CROWN-IBP (Zhang et al., 2019) made it tractable for convolutional networks, loss fusion in auto_LiRPA (Xu et al., 2020) extended it to Tiny ImageNet and downscaled ImageNet, and Shi et al. (2021) shortened the required ϵ-warmup through improved initialisation and regularisation, but none of these studies transfers a certified backbone to a new task. To our knowledge, the only transfer study that addresses certification (Vaishnavi et al., 2024) relies on randomised smoothing, which certifies an ℓ_2 ϵ-ball for a noise-smoothed classifier with high probability and requires the base classifier to be trained under the smoothing noise; Vaishnavi et al. (2024) identify this coupling as the reason that a final layer retrained without noise loses certified robustness. Deterministic verifiers such as IBP instead bound the network itself, and the bounds on a frozen backbone's features do not depend on how the head is trained. Our certified track therefore also tests whether conclusions drawn under randomised smoothing carry over to deterministic certification.

Representations of robust models Robust training also leaves a signature in internal representations. Using CKA (Kornblith et al., 2019a), Cianfarani et al. (2022) compare robust and standard networks layer by layer, for Wide ResNets on CIFAR-10 and ResNet-50 on the ImageNette and ImageWoof subsets of ImageNet under ℓ_∞ adversaries, and identify three properties. First, robust networks show reduced layer specialisation: the block structure of large standard residual networks (Nguyen et al., 2020) fades as ϵ grows and gives way to similarity between distant layers, and, unlike in standard networks, increasing width does not restore it, which the authors read as the harder objective consuming capacity that would otherwise support layer differentiation. Second, adversarial perturbations alter representations mainly in the final layers, and robust training suppresses this divergence almost entirely. Third, robust overfitting is predominantly a deeper-layer phenomenon: early layers stabilise within a few epochs, whereas later layers keep drifting after the adversarial validation loss has begun to rise. Their analysis involves no transfer, so how these properties evolve under fine-tuning, and whether pipelines of comparable robustness arrive at similar representations, remains open. For standard transfer, earlier layers are known to be more general and deeper layers more task-specific (Yosinski et al., 2014), and fine-tuning can distort pre-trained features (Kumar et al., 2022). Because CKA values have no absolute scale and a low value need not indicate functionally different features (Davari et al., 2022), we calibrate our comparisons against seed variation (Section 3.3).

B THREAT MODEL

To compare our ℓ_∞ adversary with the ℓ_2 adversaries of prior work (Section 3.2), we measure each perturbation set along two axes: the largest ℓ_2 norm it admits and the largest change it permits per input dimension (one colour channel of one pixel, hereafter a pixel). For $\delta \in \mathbb{R}^d$ with $\|\delta\|_\infty \leq \epsilon$,

$$\|\delta\|_2 \;\leq\; \epsilon\sqrt{d}, \tag{2}$$

with equality when every component takes the value $\pm\epsilon$. This bound concerns perturbation magnitude rather than attack strength, and it grows with resolution through d. The per-pixel measure

Table 4: Attack budgets across the studies considered. "Max. ℓ_2" is the largest ℓ_2 norm admitted by the constraint. The per-pixel columns give the change in $[0, 255]$ colour steps under (i) uniform distribution of the budget over all input dimensions and (ii) full concentration on a single pixel; these coincide for ℓ_∞ and differ by $\sqrt{d}$ for ℓ_2.

Study	**Norm**	ϵ	**Res.**	**Max. ℓ_2**	**Px. unif.**	**Px. max**
Ours (empirical)	ℓ_∞	$4/255$	$224^2 \times 3$	≈ 6.09	4.00	4.00
Ours (certified)	ℓ_∞	$1/255$	$56^2 \times 3$	≈ 0.38	1.00	1.00
Vaishnavi et al. (2024)	ℓ_2	0.5	$224^2 \times 3$	0.50	≈ 0.33	127.5
Kopetzki et al. (2025)	ℓ_2	0.05	$32^2 \times 3$	0.05	≈ 0.23	12.75
Kopetzki et al. (2025)	ℓ_2	0.1	$32^2 \times 3$	0.10	≈ 0.46	25.50

Table 5: Source and target datasets. N is the number of training instances per class.

Dataset	**Classes**	**Train**	**Test**	N
ImageNet	1,000	1,281,167	50,000	$732 \leq N \leq 1{,}300$
Tiny ImageNet	200	100,000	10,000	500
CIFAR-10	10	50,000	10,000	5,000
Caltech-101	101	6,942	1,735	$23 \leq N \leq 639$
Stanford Cars	196	8,141	8,041	$24 \leq N \leq 68$
CIFAR-10 10%	10	5,000	10,000	500
CIFAR-10 1%	10	500	10,000	50

behaves differently across norms. Under ℓ_∞, every pixel may move independently by up to ϵ, so the uniform and maximum per-pixel changes coincide at 255ϵ colour steps in $[0, 255]$. An ℓ_2 budget may instead be spread across all d dimensions ($\epsilon/\sqrt{d}$ per pixel) or concentrated on a single one (up to $\min(\epsilon, 1)$). Table 4 resolves each budget at the resolution at which it is applied: $224 \times 224 \times 3$ for our empirical track, $56 \times 56 \times 3$ for our certified track, and the input resolutions of the prior studies. CIFAR-10 targets are upsampled before entering the model and perturbed after upsampling, so their budgets are also resolved at the track resolution.

The ϵ-balls are not nested. Where the input spaces coincide, for our empirical track and Vaishnavi et al. (2024) at $224^2 \times 3$, neither contains the other; elsewhere the resolutions differ, and the ϵ-balls can only be compared along the two axes. Our empirical ℓ_∞ ϵ-ball admits uniform perturbations whose ℓ_2 norm (≈ 6.09) exceeds the entire ℓ_2 budget of either prior study, and even our certified ℓ_∞ ϵ-ball (≈ 0.38) exceeds both budgets of Kopetzki et al. (2025). Conversely, their ℓ_2 ϵ-balls admit perturbations concentrated on a single pixel of 12.75 to 127.5 colour steps, far beyond the 1 to 4 steps our budgets permit. An absolute ranking of the adversaries is therefore not well defined: the two norms admit different perturbations rather than more or fewer of the same ones, which is how Section 4.1 interprets agreement and disagreement with prior work.

C Experimental Setup Details

This appendix collects the details omitted from Section 3.3, including dataset statistics, preprocessing, training schedules, and the hyperparameter search.

C.1 Datasets

Table 5 lists the source and target datasets. Pre-training uses ImageNet (Russakovsky et al., 2015; Deng et al., 2009) on the empirical track and Tiny ImageNet (Russakovsky et al., 2015) on the certified track. We draw the CIFAR-10 10% and 1% variants by class-stratified subsampling of the CIFAR-10 training set, holding the label set and the full test set fixed so only the number of training examples per class varies. When a dataset provides no canonical validation split, we draw one from the training set by stratified sampling.

Prior work evaluates a broader suite of downstream tasks (Salman et al., 2020; Vaishnavi et al., 2024); however, we omit those additional datasets owing to compute constraints and use the synthetic CIFAR-10 reductions instead to separate data availability from domain shift.

C.2 PREPROCESSING

Random cropping and horizontal flipping (with probability 0.5) are the only augmentations used for every target; no colour jitter, rotation, mixup, cutmix, or random erasing is applied, and test-time transforms are deterministic. Padded crops use a padding of four pixels, and final inputs are at the resolution of the track in question, 224×224 for the empirical track and 56×56 for the certified track.

CIFAR-10 and its reductions are cropped at their native 32×32, flipped, and then upsampled to the track resolution; test images are upsampled only. Caltech-101, which contains greyscale images, is converted to RGB, resized, padded-cropped, and flipped; test images are resized and centre-cropped. Stanford Cars is the one target treated differently across tracks: at 224×224 it uses a scale-jittered `RandomResizedCrop` with scale $(0.08, 1.0)$ and a flip, and is resized to 256 then centre-cropped at test time, whereas at 56×56 it falls back to the resize and padded crop used for the other targets. On the source side, Tiny ImageNet is cropped to 56×56, with a centre crop at test time.

Each pipeline ends with conversion to a tensor. Normalisation with per-dataset channel statistics follows as the model's first layer on the empirical track and as the last pipeline step on the certified track (Section 3.3).

C.3 TRAINING SCHEDULES

Pre-training The empirical-track backbones are used as released (Section 3.3). On the certified track, both backbones are trained from scratch on Tiny ImageNet with Adam as the optimiser for certified training, which is more stable for bound-propagation objectives (Zhang et al., 2019), and a stepwise learning-rate schedule, following prior work (Xu et al., 2020; Zhang et al., 2019). We train the standard backbone with cross-entropy for 300 epochs, and the certified backbone with CROWN-IBP for 800 epochs, the longer run accommodating the ϵ-warmup. Training directly at the full budget is unstable, as bound relaxation on an untrained network yields loose intermediate bounds and ill-conditioned gradients, so ϵ starts at zero and is raised by a smooth scheduler: 100 epochs of standard training, a 320-epoch ramp to a training budget of $1.1\times$ the target, and the remaining 380 epochs at that budget.

Fine-tuning. Standard and empirical robust fine-tuning use a linear learning-rate warmup from a factor of 0.01 to the tuned value, spanning 10 epochs for runs longer than 50 epochs and 5 epochs otherwise, followed by cosine annealing. The PGD configuration of Section 3.3 (8 steps of size $\epsilon/4$) lets the attack traverse 2ϵ per input dimension, the full width of the ℓ_∞ ϵ-ball. Certified fine-tuning follows the ϵ-warmup structure of pre-training, with the phase boundaries expressed as fractions of the run length, since the number of epochs is tuned: standard training over the first 10%, a smooth ramp to the full budget over the next 40%, and the final 50% at full budget. Certified fine-tuning uses no learning-rate warmup or early stopping; its learning rate is decayed tenfold at 75% and 87.5% of the run. All other runs stop early if the validation objective of Appendix C.4 does not improve for 15 consecutive epochs.

C.4 HYPERPARAMETER OPTIMISATION

Robust training is costly and sensitive enough that grid search is impractical, so we use sequential model-based optimisation (SMBO) (Shahriari et al., 2015) for its sample efficiency, implemented with Ax (Olson et al., 2025) and BoTorch (Balandat et al., 2020). Each search runs 20 trials under standard fine-tuning and 32 under robust fine-tuning, in batches of at most four. We consider the unequal budget fair given the complexity of the robust objectives: their loss landscapes are harder to optimise, and certified fine-tuning adds two hyperparameters to the search space, so an equal budget would systematically under-tune the robust configurations. The first five trials are drawn quasi-randomly from a Sobol sequence to cover the space; subsequent trials use a SingleTaskGP surrogate with the dimension-scaled priors of Hvarfner et al. (2024) with the qLogNEI acquisition function (Ament et al., 2023), which suits a noisy single-objective measurement and avoids the

vanishing-gradient pathology of standard expected improvement, and a mixed optimiser (Wan et al., 2021) for the combined continuous and discrete space. Under a fixed seed, the Sobol initialisation and the acquisition-driven trials are deterministic, so a search is reproducible up to the nondeterminism of GPU training.

We tune the number of epochs and batch size as discrete parameters, and the learning rate and weight decay as continuous parameters on a log scale. Certified fine-tuning additionally tunes the final κ value on a linear scale and the maximum gradient norm for clipping on a log scale. We define separate search spaces for linear probing and full fine-tuning: full fine-tuning uses narrower learning-rate and weight-decay ranges to limit drift from the pre-trained representation, and a longer epoch budget.

The search maximises the scalarised validation objective

$$J = 1 \cdot \mathrm{Acc}_{\mathrm{clean}} + 1.8 \cdot \mathrm{Acc}_{\mathrm{rob}}, \tag{3}$$

where $\mathrm{Acc}_{\mathrm{rob}}$ is the PGD-8 robust accuracy on the empirical track and the IBP-certified accuracy on the certified track, both on the validation split. This also serves as the early-stopping criterion. We optimise accuracy rather than loss because robust objectives can produce unstable or exploding losses, whereas accuracy is bounded, and we include both terms because they are in tension in configurations such as Direct Transfer, where preserving robustness calls for a smaller learning rate than clean accuracy alone would select.

The selected configuration is retrained under three seeds without further search (Section 3.3).

C.5 Selected hyperparameters

Tables 6 and 7 list the hyperparameters selected by the search in Appendix C.4 for every configuration, downstream task, and adaptation mechanism, on the empirical (ResNet-50) and certified (Wide ResNet) tracks, respectively. All three seeded runs behind the reported results use these values; the search ranges are defined in our code (`anonymous.4open.science/r/robust-transfer-regimes-3C60`). The number of epochs is the tuned budget, and runs with early stopping may end before it.

Table 6: Final hyperparameters for the empirical track (ResNet-50), selected via SMBO. Configurations are named by (source pre-training, target fine-tuning). LP = linear probing, FT = full fine-tuning. Control configurations are evaluated under FT only.

Configuration	Mode	Learn. rate	Weight decay	Batch	Epochs
CIFAR-10					
Standard Control (none, std)	FT	0.091421	0.001958	256	100
Standard Baseline (std, std)	LP	0.080252	0.000193	512	50
Standard Baseline (std, std)	FT	0.020410	0.001	256	30
Robust Control (none, rob)	FT	0.076620	0.000955	128	100
Direct Transfer (rob, std)	LP	0.372760	0.001	512	30
Direct Transfer (rob, std)	FT	0.016484	0.001	512	50
Robust Adaptation (std, rob)	LP	0.001	0.00001	128	50
Robust Adaptation (std, rob)	FT	0.020	0.001	128	50
Fully Robust (rob, rob)	LP	0.078543	0.00001	256	50
Fully Robust (rob, rob)	FT	0.006262	0.001	128	100
Caltech-101					
Standard Control (none, std)	FT	0.5	0.000355	128	100
Standard Baseline (std, std)	LP	0.5	0.000111	128	50
Standard Baseline (std, std)	FT	0.05	0.00005	128	20
Robust Control (none, rob)	FT	0.372213	0.000506	256	200
Direct Transfer (rob, std)	LP	0.090051	0.001	256	30
Direct Transfer (rob, std)	FT	0.001663	0.000227	128	40
Robust Adaptation (std, rob)	LP	0.5	0.000021	128	30
Robust Adaptation (std, rob)	FT	0.0479	0.001	128	40
Fully Robust (rob, rob)	LP	0.042493	0.000035	256	100

continued on next page

Table 6 – continued from previous page

Configuration	Mode	Learn. rate	Weight decay	Batch	Epochs
Fully Robust (rob, rob)	FT	0.031272	0.000403	256	50
Stanford Cars					
Standard Control (none, std)	FT	0.086563	0.001008	128	100
Standard Baseline (std, std)	LP	0.5	0.001	128	30
Standard Baseline (std, std)	FT	0.05	0.0001	128	50
Robust Control (none, rob)	FT	0.076485	0.001187	128	100
Direct Transfer (rob, std)	LP	0.5	0.001	128	30
Direct Transfer (rob, std)	FT	0.05	0.000583	128	50
Robust Adaptation (std, rob)	LP	0.018143	0.001327	128	30
Robust Adaptation (std, rob)	FT	0.05	0.005	256	100
Fully Robust (rob, rob)	LP	0.126249	0.001	128	50
Fully Robust (rob, rob)	FT	0.05	0.00056	128	100
CIFAR-10 10%					
Standard Control (none, std)	FT	0.121019	0.002609	256	100
Standard Baseline (std, std)	LP	0.043412	0.001006	256	30
Standard Baseline (std, std)	FT	0.016286	0.000722	128	30
Robust Control (none, rob)	FT	0.041349	0.005	128	100
Direct Transfer (rob, std)	LP	0.022064	0.01	128	30
Direct Transfer (rob, std)	FT	0.00921	0.005	128	50
Robust Adaptation (std, rob)	LP	0.5	0.01	128	30
Robust Adaptation (std, rob)	FT	0.017914	0.005	128	100
Fully Robust (rob, rob)	LP	0.007883	0.000258	256	100
Fully Robust (rob, rob)	FT	0.002315	0.000186	256	100
CIFAR-10 1%					
Standard Control (none, std)	FT	0.064003	0.003413	32	200
Standard Baseline (std, std)	LP	0.01032	0.1	128	30
Standard Baseline (std, std)	FT	0.006475	0.0005	64	80
Robust Control (none, rob)	FT	0.067292	0.001	32	200
Direct Transfer (rob, std)	LP	0.013512	0.089367	64	30
Direct Transfer (rob, std)	FT	0.01	0.0005	32	30
Robust Adaptation (std, rob)	LP	0.019734	0.001	128	20
Robust Adaptation (std, rob)	FT	0.002834	0.000963	32	100
Fully Robust (rob, rob)	LP	0.014651	0.01644	64	30
Fully Robust (rob, rob)	FT	0.008094	0.0005	64	50

Table 7: Final hyperparameters for the certified track (Wide ResNet), selected via SMBO. Configurations are named by (source pre-training, target fine-tuning). LP = linear probing, FT = full fine-tuning. Control configurations are evaluated under FT only. Standard fine-tuning uses SGD and certified fine-tuning uses Adam, so learning rates are not comparable between the two. Final κ and the gradient-clip norm apply only to certified fine-tuning.

Configuration	Mode	Learn. rate	Weight decay	Batch	Epochs	Final κ	Grad. clip
CIFAR-10							
Standard Control (none, std)	FT	0.025611	0.000050	64	200		
Standard Baseline (std, std)	LP	0.045577	0.000022	64	50		
Standard Baseline (std, std)	FT	0.035453	0.001000	128	30		
Robust Control (none, rob)	FT	0.000365	0.000050	96	200	0.500000	1.000000
Direct Transfer (rob, std)	LP	0.038179	0.000010	128	30		
Direct Transfer (rob, std)	FT	0.050000	0.000097	64	50		
Robust Adaptation (std, rob)	LP	0.001000	0.000039	96	80	0.064683	1.000000
Robust Adaptation (std, rob)	FT	0.000314	0.000120	64	80	0.500000	4.745325
Fully Robust (rob, rob)	LP	0.001694	0.000203	96	80	0.314852	2.873345
Fully Robust (rob, rob)	FT	0.000526	0.000154	96	80	0.452643	2.851134
Caltech-101							
Standard Control (none, std)	FT	0.199227	0.000482	64	100		
Standard Baseline (std, std)	LP	0.121504	0.000010	64	30		

continued on next page

Table 7 – continued from previous page

Configuration	Mode	Learn. rate	Weight decay	Batch	Epochs	Final κ	Grad. clip
Standard Baseline (std, std)	FT	0.050000	0.000272	64	50		
Robust Control (none, rob)	FT	0.001504	0.000168	96	200	0.500000	1.412990
Direct Transfer (rob, std)	LP	0.178823	0.000010	128	50		
Direct Transfer (rob, std)	FT	0.041213	0.000050	64	50		
Robust Adaptation (std, rob)	LP	0.100000	0.000013	96	80	0.000000	1.000000
Robust Adaptation (std, rob)	FT	0.000674	0.000050	64	80	0.449215	2.517920
Fully Robust (rob, rob)	LP	0.002158	0.000044	64	80	0.188971	8.514866
Fully Robust (rob, rob)	FT	0.001076	0.000246	96	80	0.346158	5.915038
Stanford Cars							
Standard Control (none, std)	FT	0.100000	0.001873	128	100		
Standard Baseline (std, std)	LP	0.148467	0.001000	64	30		
Standard Baseline (std, std)	FT	0.050000	0.000210	64	50		
Robust Control (none, rob)	FT	0.000910	0.000190	96	200	0.000000	1.000000
Direct Transfer (rob, std)	LP	0.500000	0.001000	128	30		
Direct Transfer (rob, std)	FT	0.050000	0.000100	64	50		
Robust Adaptation (std, rob)	LP	0.100000	0.000010	64	80	0.000000	4.069106
Robust Adaptation (std, rob)	FT	0.000345	0.000499	96	80	0.500000	6.673864
Fully Robust (rob, rob)	LP	0.004936	0.000575	64	50	0.375011	2.675668
Fully Robust (rob, rob)	FT	0.001260	0.000431	96	80	0.091215	4.557789
CIFAR-10 10%							
Standard Control (none, std)	FT	0.108724	0.000050	128	200		
Standard Baseline (std, std)	LP	0.386892	0.000102	64	30		
Standard Baseline (std, std)	FT	0.027810	0.000050	64	50		
Robust Control (none, rob)	FT	0.001242	0.000156	96	200	0.336834	1.111131
Direct Transfer (rob, std)	LP	0.133065	0.000558	128	30		
Direct Transfer (rob, std)	FT	0.050000	0.000145	64	50		
Robust Adaptation (std, rob)	LP	0.001694	0.000203	96	80	0.314852	2.873345
Robust Adaptation (std, rob)	FT	0.001219	0.000116	96	80	0.314116	2.656192
Fully Robust (rob, rob)	LP	0.001694	0.000203	96	80	0.314852	2.873345
Fully Robust (rob, rob)	FT	0.003544	0.000507	96	80	0.419741	1.000000
CIFAR-10 1%							
Standard Control (none, std)	FT	0.023988	0.001249	128	200		
Standard Baseline (std, std)	LP	0.048745	0.004243	64	30		
Standard Baseline (std, std)	FT	0.003557	0.000218	64	50		
Robust Control (none, rob)	FT	0.003141	0.001000	96	200	0.000000	1.991456
Direct Transfer (rob, std)	LP	0.133065	0.055829	128	20		
Direct Transfer (rob, std)	FT	0.038689	0.000227	64	30		
Robust Adaptation (std, rob)	LP	0.011874	0.000049	96	80	0.028903	1.032410
Robust Adaptation (std, rob)	FT	0.000596	0.001000	96	50	0.000000	1.443244
Fully Robust (rob, rob)	LP	0.001037	0.000247	96	80	0.340903	2.912425
Fully Robust (rob, rob)	FT	0.004214	0.000139	96	80	0.000000	1.000000

D Additional results

This appendix provides additional results for completeness and reference that support the results and findings presented in Section 4.

D.1 Full results tables

Tables 8 and 9 contain the full results, including standard accuracies and measured standard deviations over the re-seeded runs, as a complement to Table 3.

The clean cost of robustness differs sharply between tracks Under full fine-tuning, Fully Robust gives up 0.54–6.17 points of clean accuracy relative to the Standard Baseline on the empirical track, but 26.44–51.02 points on the certified track, where even the Standard Control trained from scratch is more accurate than Fully Robust on four of five targets. Robust Control, which never uses the certified backbone, is also 21.18–25.38 points less accurate than the Standard Control on four of

five targets, so a large part of this cost comes with the certified objective itself rather than with the backbone.

Robust pre-training buys clean accuracy only on the empirical track Relative to Robust Control, Fully Robust gains 3.40–52.15 points of clean accuracy on the empirical track, most on Stanford Cars and CIFAR-10 1%, where adversarial training of a ResNet-50 from scratch reaches only 32.33% and 34.24%. On the certified track, it is less accurate than Robust Control on four of five targets, by 1.76–7.55 points, and more accurate only on CIFAR-10 1%, by 4.37 points, mirroring the robust accuracies of Section 4.2. Under linear probing, robust features yield lower clean accuracy than standard features on every target (Direct Transfer versus Standard Baseline), by 0.44–13.93 points on the empirical track and 8.06–30.49 points on the certified track. Salman et al. (2020) find that robust backbones can improve downstream clean accuracy (Appendix A), but that the gain depends on the robustness level and the target; at the single budget we evaluate, we do not observe it.

A robust head on a standard backbone changes, but gains no robustness Under linear probing, Robust Adaptation reaches at most 0.01% robust accuracy on either track (Table 3), yet on the empirical track, its clean accuracy falls 14.65–41.01 points below that of the Standard Baseline. The robust objective thus reshapes the head substantially, but cannot make the frozen standard features robust. On the certified track, clean accuracy falls by at most 4.76 points.

Variation across seeds Most standard deviations are below one point, and the near-ties that Section 4 discusses, such as Robust Adaptation and Fully Robust on certified CIFAR-10 10%, are read there as such. Larger deviations concentrate in two settings: training the ResNet-50 from scratch, as in Standard Control on CIFAR-10 and CIFAR-10 1% (11.12 and 12.86 points of clean accuracy) and Robust Control on Caltech-101 (4.81 clean, 9.46 robust); and robust linear heads on standard backbones, as in Robust Adaptation on CIFAR-10 10% and 1% (6.31 and 4.04 clean). As hyperparameters are tuned under a single seed, these deviations reflect training stochasticity at the selected configuration, not tuning variance (Section 3.3).

D.2 CKA RESULTS

This subsection supports the representational analysis of Section 4.3 with three sets of results, all computed with linear CKA on benign inputs from the CIFAR-10 10% test split (Section 3.3): the seed floors against which similarities are read (Table 10), the full cross-model matrices whose diagonals Figure 3 plots (Figures 4 and 5), and the within-model matrices of the fine-tuned configurations (Figure 6). As activations are extracted at every layer, including pooling, activation, and normalisation layers, ResNet-50 yields about 150 layers and the Wide ResNet about 30. This difference in sampling also explains why the empirical profiles in Figure 3 are more jagged. Residual connections produce a grid pattern in which post-residual activations are more similar to one another than activations within a block (Kornblith et al., 2019a), and the alternation of convolution, normalisation, and activation layers resolves this pattern finely on ResNet-50 but only coarsely on the Wide ResNet. The jaggedness thus reflects architecture and sampling granularity rather than measurement noise.

Seed floors On the empirical track, reproducibility follows the backbone. Both configurations initialised from the robust backbone reproduce almost exactly (mean ≥ 0.972), whereas both initialised from the standard backbone do not (0.774–0.796), regardless of the fine-tuning objective. This matches Figure 3a, where fine-tuning moves the robust backbone little: a model that stays close to its initialisation leaves little room for seeds to diverge. On the certified track, reproducibility follows the fine-tuning objective instead. The standard objective reproduces at 0.884–0.891 from either backbone and the certified objective at 0.768–0.828, least for Robust Adaptation, whose least similar seed pair reaches only 0.671. This plausibly reflects the less stable optimisation of certified training, with its ϵ-warmup and loose bounds early in training. We use each configuration's mean as its noise floor, and the lower of the two floors when comparing two configurations.

Full cross-model matrices Figure 3 plots only the diagonals of the cross-model similarity matrices, and a low diagonal value does not by itself show that two representations have diverged: Davari et al. (2022) show that CKA can be driven towards zero by transformations that leave a network's

Table 8: Standard and empirical robust accuracy of ResNet-50 across downstream datasets and adaptation modes (linear probing, LP; full fine-tuning, FT). Robust accuracy is reported under AutoAttack with an ℓ_∞ adversary at $\epsilon = 4/255$. Each cell is the mean across seeds with the sample standard deviation subscripted. The best robust accuracy in each row is shown in **bold**.

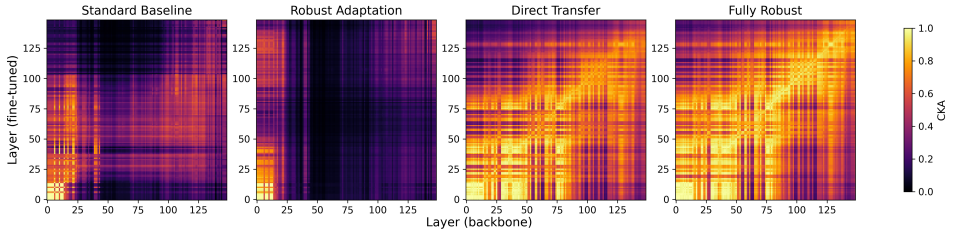
		Standard Control (none, std)		Standard Baseline (std, std)		Robust Control (none, rob)		Direct Transfer (rob, std)		Robust Adaptation (std, rob)		Fully Robust (rob, rob)	
Downstream task	Mode	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob
CIFAR-10	LP	—	—	91.73 $\pm$0.02	0.00 $\pm$0.00	—	—	86.47 $\pm$0.17	36.05 $\pm$0.06	67.89 $\pm$0.57	0.00 $\pm$0.00	83.11 $\pm$1.48	**46.03** $\pm$0.27
	FT	88.42 $\pm$11.12	0.00 $\pm$0.00	97.61 $\pm$0.05	0.00 $\pm$0.00	89.54 $\pm$0.24	63.10 $\pm$0.11	97.99 $\pm$0.16	0.00 $\pm$0.00	87.02 $\pm$0.73	57.53 $\pm$0.02	92.94 $\pm$0.02	**68.19** $\pm$0.27
Caltech-101	LP	—	—	98.48 $\pm$0.24	0.00 $\pm$0.00	—	—	98.04 $\pm$0.21	82.11 $\pm$0.63	79.15 $\pm$2.30	0.00 $\pm$0.00	97.16 $\pm$0.73	**88.38** $\pm$2.84
	FT	95.33 $\pm$0.32	5.84 $\pm$0.64	99.14 $\pm$0.26	0.00 $\pm$0.00	91.59 $\pm$4.81	81.40 $\pm$9.46	95.50 $\pm$2.07	64.05 $\pm$2.26	97.75 $\pm$0.12	91.72 $\pm$0.23	98.60 $\pm$0.19	**95.52** $\pm$0.20
Stanford Cars	LP	—	—	46.93 $\pm$0.47	0.02 $\pm$0.02	—	—	42.12 $\pm$0.23	11.41 $\pm$0.30	5.92 $\pm$0.62	0.00 $\pm$0.00	39.52 $\pm$0.27	**14.14** $\pm$0.26
	FT	67.41 $\pm$1.18	0.00 $\pm$0.01	86.97 $\pm$0.22	0.00 $\pm$0.00	32.33 $\pm$1.76	14.35 $\pm$1.06	87.27 $\pm$0.12	0.01 $\pm$0.01	1.26 $\pm$0.54	1.04 $\pm$0.51	84.48 $\pm$0.38	**64.56** $\pm$0.36
CIFAR-10 10%	LP	—	—	88.93 $\pm$0.21	0.00 $\pm$0.00	—	—	82.24 $\pm$0.05	35.92 $\pm$0.22	62.01 $\pm$6.31	0.00 $\pm$0.00	80.08 $\pm$0.66	**42.30** $\pm$0.45
	FT	82.01 $\pm$0.45	0.00 $\pm$0.00	94.34 $\pm$0.24	0.00 $\pm$0.00	71.27 $\pm$0.78	42.91 $\pm$0.14	95.37 $\pm$0.23	0.02 $\pm$0.02	71.53 $\pm$1.67	35.34 $\pm$0.79	88.64 $\pm$0.26	**57.30** $\pm$0.29
CIFAR-10 1%	LP	—	—	82.78 $\pm$0.27	0.00 $\pm$0.00	—	—	68.85 $\pm$2.24	32.86 $\pm$0.32	68.13 $\pm$4.04	0.01 $\pm$0.01	71.04 $\pm$0.89	**35.55** $\pm$0.40
	FT	28.74 $\pm$12.86	3.64 $\pm$3.56	85.18 $\pm$1.07	0.00 $\pm$0.01	34.24 $\pm$3.22	18.83 $\pm$1.62	88.10 $\pm$0.03	0.39 $\pm$0.14	49.37 $\pm$1.55	0.02 $\pm$0.01	79.01 $\pm$0.81	**41.46** $\pm$0.68

Table 9: Standard and certified robust accuracy of Wide ResNet across downstream datasets and adaptation modes (linear probing, LP; full fine-tuning, FT). Certified robust accuracy is obtained via IBP under an ℓ_∞ adversary at $\epsilon = 1/255$. Each cell is the mean across seeds with the sample standard deviation subscripted. The best robust accuracy in each row is shown in **bold**.

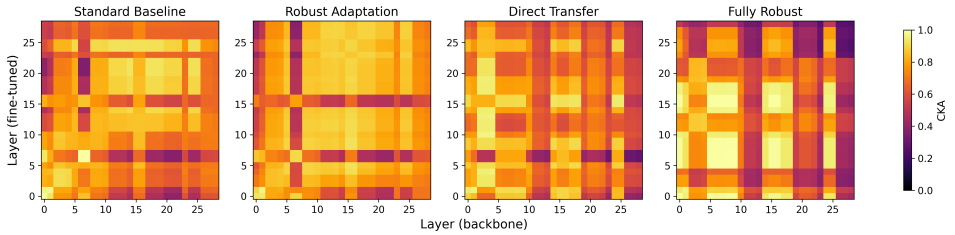
		Standard Control (none, std)		Standard Baseline (std, std)		Robust Control (none, rob)		Direct Transfer (rob, std)		Robust Adaptation (std, rob)		Fully Robust (rob, rob)	
Downstream task	Mode	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob
CIFAR-10	LP	—	—	72.17 $\pm$0.08	0.00 $\pm$0.00	—	—	47.33 $\pm$0.51	26.76 $\pm$1.63	68.43 $\pm$0.69	0.00 $\pm$0.00	44.41 $\pm$0.13	**34.90** $\pm$0.19
	FT	90.79 $\pm$0.60	0.00 $\pm$0.00	92.24 $\pm$0.11	0.00 $\pm$0.00	66.19 $\pm$0.11	**46.65** $\pm$0.62	82.53 $\pm$0.28	0.00 $\pm$0.00	61.80 $\pm$0.24	43.04 $\pm$0.42	61.15 $\pm$0.75	43.42 $\pm$0.83
Caltech-101	LP	—	—	88.36 $\pm$0.42	0.00 $\pm$0.00	—	—	57.87 $\pm$0.72	43.05 $\pm$0.93	85.38 $\pm$0.77	0.00 $\pm$0.00	55.77 $\pm$0.93	**45.92** $\pm$0.95
	FT	94.72 $\pm$0.50	0.00 $\pm$0.00	95.75 $\pm$0.64	0.00 $\pm$0.00	73.54 $\pm$0.25	54.10 $\pm$0.47	75.52 $\pm$0.15	0.00 $\pm$0.00	73.74 $\pm$0.67	**55.18** $\pm$0.99	65.99 $\pm$0.26	51.39 $\pm$0.13
Stanford Cars	LP	—	—	14.08 $\pm$0.17	0.00 $\pm$0.00	—	—	6.02 $\pm$0.33	2.86 $\pm$0.09	14.42 $\pm$0.36	0.00 $\pm$0.00	5.88 $\pm$0.50	**3.10** $\pm$0.18
	FT	35.92 $\pm$0.90	0.00 $\pm$0.00	58.58 $\pm$0.32	0.00 $\pm$0.00	10.54 $\pm$0.28	**4.98** $\pm$0.43	11.91 $\pm$0.62	0.00 $\pm$0.00	24.35 $\pm$0.86	0.00 $\pm$0.00	7.56 $\pm$0.53	4.06 $\pm$0.27
CIFAR-10 10%	LP	—	—	66.99 $\pm$0.11	0.00 $\pm$0.00	—	—	45.23 $\pm$0.94	27.44 $\pm$1.05	63.40 $\pm$1.38	0.00 $\pm$0.00	44.41 $\pm$0.42	**33.40** $\pm$0.07
	FT	76.17 $\pm$2.91	0.00 $\pm$0.00	84.03 $\pm$0.16	0.00 $\pm$0.00	50.90 $\pm$1.04	36.70 $\pm$0.37	69.45 $\pm$0.42	0.00 $\pm$0.00	50.90 $\pm$0.56	36.68 $\pm$0.38	49.14 $\pm$0.35	**36.72** $\pm$0.28
CIFAR-10 1%	LP	—	—	54.80 $\pm$0.95	0.00 $\pm$0.00	—	—	34.64 $\pm$1.57	25.12 $\pm$0.92	50.04 $\pm$2.42	0.00 $\pm$0.00	34.22 $\pm$2.64	**25.90** $\pm$1.57
	FT	35.56 $\pm$0.99	0.00 $\pm$0.00	62.89 $\pm$1.55	0.00 $\pm$0.00	32.08 $\pm$1.16	24.99 $\pm$1.99	41.90 $\pm$1.05	0.00 $\pm$0.00	28.55 $\pm$1.10	18.10 $\pm$1.23	36.45 $\pm$1.02	**26.51** $\pm$0.73

Table 10: Diagonal CKA between runs differing only in random seed, over all three pairs from three seeds per configuration on CIFAR-10 10% (empirical: ResNet-50; certified: Wide ResNet). Values near one indicate that matched-layer representations are reproducible.

	Standard Baseline		Direct Transfer		Robust Adaptation		Fully Robust	
Track	Mean	Min	Mean	Min	Mean	Min	Mean	Min
Empirical	0.796	0.747	0.972	0.970	0.774	0.706	0.988	0.987
Certified	0.891	0.870	0.884	0.840	0.768	0.671	0.828	0.789



(a) Empirical (ResNet-50).

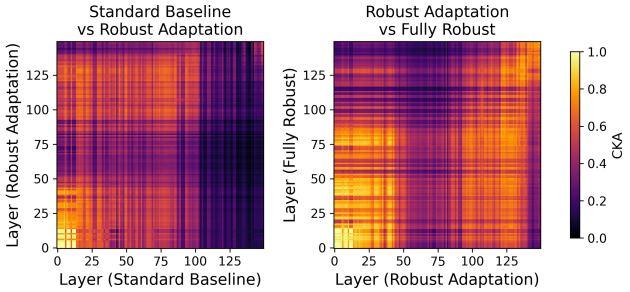


(b) Certified (Wide ResNet).

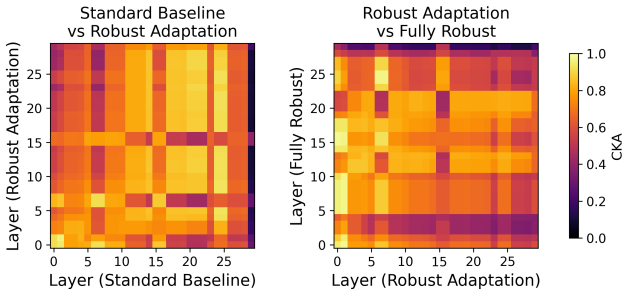
Figure 4: Full cross-model CKA matrices between each configuration fully fine-tuned on CIFAR-10 10% and its own pre-trained backbone. Diagonals are plotted in Figure 3a.

function intact. Were such a shift, or a depth offset between the two models, responsible for a low diagonal, the displaced similarity would reappear off the diagonal as a bright band linking layer i of one model to some layer $j \neq i$ of the other. We therefore read a low diagonal as divergence only when the surrounding off-diagonal entries are also low. Figures 4 and 5 report the full matrices for this check. Against their backbones (Figure 4), the empirical Standard Baseline and Robust Adaptation matrices darken from the early layers onwards, and no bright band appears away from the diagonal, so their low diagonals reflect genuine divergence rather than a shift in depth. Direct Transfer and Fully Robust keep a bright diagonal through most of the network. The certified matrices of Standard Baseline and Robust Adaptation stay close to their floors and do not appear to separate the configurations. Between configurations (Figure 5), the Robust Adaptation–Fully Robust matrix darkens through the mid-network on the empirical track and across the deepest quarter on the certified track, in both cases without an off-diagonal band, so the distinct representations reported in Section 4.3 are not an artefact of reading only the diagonal.

Within-model matrices Figure 6 shows the similarity between every pair of layers within each fine-tuned model. On the empirical track, the Standard Baseline shows the block structure of standard residual networks, with a bright final block separated from the earlier layers by low similarity. Both robust configurations mostly lose this structure, the reduced layer specialisation that Cianfarani et al. (2022) identify as a signature of robust training: Fully Robust shows uniformly elevated similarity with no block structure left, and Robust Adaptation raises cross-layer similarity throughout, leaving only faint traces of the blocks. Both thus reorganise in the same direction, although Figure 3b shows that they do not converge at matched layers. On the certified track, the patterns are coarser, as the Wide ResNet is sampled at far fewer layers, but the picture is the same: Robust Adaptation is almost uniformly similar across layers, and Fully Robust loses most of the Standard Baseline's

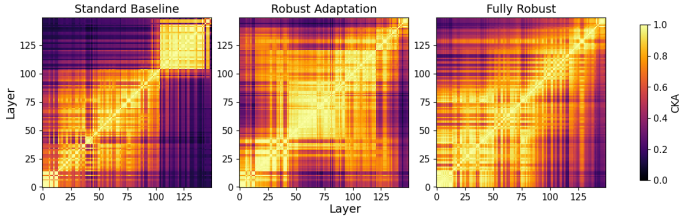


(a) Empirical (ResNet-50).

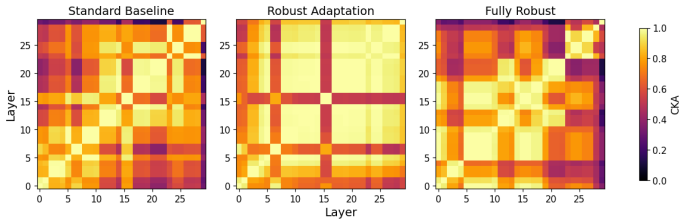


(b) Certified (Wide ResNet).

Figure 5: Full cross-model CKA matrices between fine-tuned configurations on CIFAR-10 10%: Standard Baseline versus Robust Adaptation (left) and Robust Adaptation versus Fully Robust (right). Diagonals are plotted in Figure 3b.



(a) Empirical (ResNet-50).



(b) Certified (Wide ResNet).

Figure 6: Within-model linear CKA for the fine-tuned configurations on CIFAR-10 10%. Bright off-diagonal regions indicate reduced layer specialisation. The Standard Baseline retains block structure on both tracks; in Robust Adaptation and Fully Robust it is mostly lost, with traces remaining only in empirical Robust Adaptation and in the final block of certified Fully Robust.

block structure, keeping a distinct final block. As the certified floors are lower (Table 10), we do not read finer differences into these matrices.