A Guide to Robust Generalization: The Impact of Architecture, Pre-training, and Optimization Strategy

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep learning models are vulnerable to small input perturbations. For years, robustness to such perturbations was pursued by training models from scratch (i.e., with random initializations) using specialized loss objectives. Recently, robust fine-tuning has emerged as a more efficient alternative: instead of training from scratch, pretrained models are adapted to maximize predictive performance and robustness. To conduct robust fine-tuning, practitioners design an optimization strategy that includes the model update protocol (e.g., full or partial) and the specialized loss objective. Additional design choices include the architecture type and size, and the pretrained representation. These design choices affect robust generalization, which is the model's ability to maintain performance when exposed to new and unseen perturbations at test time. Understanding how these design choices influence generalization remains an open question with significant practical implications. In response, we present an empirical study spanning 6 datasets, 40 pretrained architectures, 2 specialized losses, and 3 adaptation protocols — yielding $1,440$ training configurations and $7,200$ robustness measurements across five perturbation types. To our knowledge, this is the most diverse and comprehensive benchmark of robust fine-tuning to date. While attention-based architectures and robust pretrained representations are increasingly popular, we find that convolutional neural networks pretrained in a supervised manner on large datasets often perform best. Our analysis both confirms and challenges prior design assumptions, highlighting promising research directions and offering practical guidance.

1 Introduction

Images processed by machine learning models can contain subtle perturbations that are invisible to the human eye. These perturbations may occur accidentally (e.g. sensor noise, blur, digital format conversions [Jung, 2018]) or intentionally (e.g., adversarial attacks [Szegedy et al., 2014]). Such perturbations can negatively affect the performance of machine learning systems, which is a serious obstacle to their adoption in the real world.

In practice, it is difficult to anticipate which type(s) of perturbation(s) a system may face [Sculley et al., 2015]. A key challenge is therefore to maximize robustness across diverse perturbation types. To achieve that, a typical approach is to assume a set of possible perturbations and induce robustness to this specific set during training [Croce and Hein, 2022, Tramèr and Boneh, 2019, Maini et al., 2020]. However, this strategy is inherently limited, as models may encounter unforeseen perturbations post-deployment [Bashivan et al., 2021, Ibrahim et al., 2022]. In this work, we focus on *robust generalization*: it refers to the ability of models trained for robustness on a specific perturbation type to remain robust to other, unseen, perturbations.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

We specifically focus on robust generalization in low data regimes. Robustness-critical applications often face data scarcity constraints, due to data collection costs [Rahimi et al., 2021]. In low data regimes, robust generalization can be induced by fine-tuning for robustness models pre-trained on large datasets [Hua et al., 2024, Xu et al., 2023, Hendrycks et al., 2019, Liu et al., 2023a].

Fine-tuning for robustness involves a wide range of design choices related to the pretrained backbone and the fine-tuning process. When selecting a pretrained backbone, one implicitly selects an architecture type (e.g., convolutional, attention-based, or hybrid), a model size, and a pretraining strategy (e.g., supervised vs. self-supervised, robust vs. non-robust). As for the robust fine-tuning process, one must select a fine-tuning protocol (e.g., partial vs full updates) and a loss objective. A standard loss objective is classic adversarial training (Classic AT) [Madry et al., 2018], which minimizes cross-entropy on adversarially perturbed observations. Another option is the so-called TRADES (TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization) [Zhang et al., 2019] loss, which optimizes both the cross-entropy and the Kullback-Leibler (KL) divergence between predictions on perturbed and unperturbed observations. Unfortunately, there is currently limited guidance available for practitioners to navigate these choices effectively.

Research question *What are the impacts of robust fine-tuning design choices on robust generalization?* Our main hypothesis is that the pretrained backbone interacts with the fine-tuning and optimization strategies to substantially influence robust generalization. Figure 1 motivates this hypothesis by showing important performance variability across design choices and complex interaction patterns among design components.

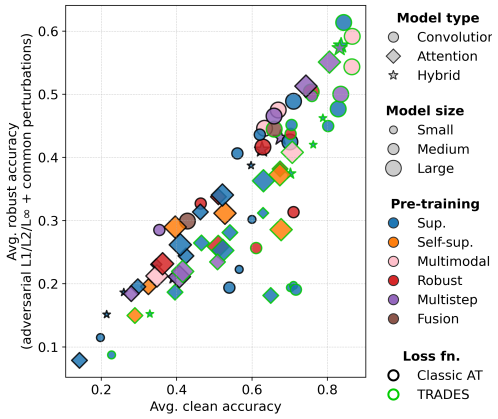


Figure 1: Performance variation across fine-tuning design choices (full fine tuning with 50 epochs). Accuracy averaged over 6 datasets.

Key findings We conduct a study on 6 datasets and a total of 240 design choices combinations (40 pretrained backbones $\times$ 2 robust losses $\times$ 3 fine-tuning protocols). We obtain $7,200$ measurements of robustness on 5 perturbation types. We uncover actionable lessons for practitioners and for future research on robust fine-tuning:

① TRADES loss performs better than Classic AT overall and significantly better in large models. ② Despite growing interest in attention architectures, convolutional architectures show better robust generalization in the considered setups. ③ Hybrid architectures are a promising avenue in robust fine-tuning. ④ With enough compute, supervised pre-training yields best robust generalization, but multi-modal pre-training is also promising. ⑤ Robust pre-training is the clear winner in resource constrained fine-tuning settings. ⑥ When fine-tuning robust backbones with enough compute, using a loss different from the one used for pre-training can boost performance. ⑦ Robust pre-training yields limited returns when scaled to larger architectures. ⑧ Full finetuning is the best overall, and there exist a cost-effective proxy to guide practitioners in finding successful design choices faster.

Related benchmarks Tang et al. [2021], Liu et al. [2023b], and Shao et al. [2021] benchmark the performance of different architectures and training strategies on robustness. A main difference is that they all consider "training from scratch" (i.e., training from random initializations). In contrast, our study focuses on fine-tuning from pretrained backbones. Training dynamics observed in one setting do not necessarily transfer to the other [Kornblith et al., 2019]. Another key difference is that the current benchmark analyzes configurations with optimized hyper-parameters (see details in Appendix B), while prior works consider fixed hyper-parameters. This study is therefore better geared towards practitioners. More broadly, this work is inspired by design choices studies in non-robust computer vision [Goldblum et al., 2024] and in robust vision-language [Bhagwatkar et al., 2024].

2 Design choices

We study 80 combinations (40 pre-trained backbones $\times$ 2 objective losses) using 3 fine-tuning protocols over 6 classification tasks with C classes. Each observation-label pair (x, y) is drawn i.i.d.

from a stationary distribution (X, Y). Each configuration results in a classifier model $f_\theta : X \to \Delta_C$, where θ are the model parameters and Δ_C denotes the $(C-1)$-dimensional probability simplex.

2.1 Pre-trained backbones

Tremendous progress has been made in the development of pre-trained backbones, and each technique is usually followed by multiple variations. The options available in the open-source community are endless [Wightman, 2019], which motivates an extensive benchmarking of pre-trained backbones.

Size	Param. Range	Type	Architectures
Small	5–10M	Conv.	Regnetx004, Efficientnet-b0, Edgenext (small)
		Attn.	DeiT (tiny)
		Hybrid	Coat (tiny), MobileViT (small)
Medium	25–30M	Conv.	Convnext (tiny), Resnet50
		Attn.	DeiT (small), ViT (small), Eva02 (tiny), Swin (tiny)
		Hybrid	Coatnet-0
Large	80–90M	Conv.	Convnext (base)
		Attn.	ViT (base), Eva02 (base), Swin (base)
		Hybrid	Coatnet-2

Table 1: Overview of the 19 considered architectures.

Architectures We consider a total of 19 architectures, spanning into three size categories: large (80–90 million parameters), medium (25–30 million), and small (5–10 million) – see summary in Table 1. Each architecture is further categorized between one of three structural types: convolutional, attention-based, and hybrid (i.e., mixture of convolution and attention layers). Small architectures are relevant for deployment in low resource environments (e.g., Jetson Nano, Orion) or with low latency requirements. To our knowledge, this is the first study in robust fine-tuning that considers small size architectures (5-10M) [Hua et al., 2024, Xu et al., 2023, Hendrycks et al., 2019, Liu et al., 2023a]. The largest architectures considered are aligned with existing works [Goldblum et al., 2024, Hua et al., 2024]. For larger architectures, we refer to works on scaling robustness [Wang et al., 2024].

Category	Total	Technical Details
Supervised	20	ImageNet-1k/22k; variants with and without data-aug. & regularization
Multistep Supervised	6	Imagenet-22k then 1k, Imagenet-12k then 1k, variants with and without data-aug. & regularization
Robust Supervised	5	4× APGD-K, 1× PGD-K adversarial pre-training; all based on Classic AT on In1k
Unimodal Self-Sup.	4	MAE, DINO, MIM
Multimodal Self-Sup.	3	CLIP on LAION-2B / LAION-Aesthetics
Fusion	2	CLIP (LAION-2B) followed by fine-pass on Imagenet-1k, and Imagenet-12k/1k

Table 2: Overview of the 40 considered backbones.

Pre-training protocol Prior works have studied the influence of supervised pre-training [Hendrycks et al., 2019, Mo et al., 2022], robust pre-training [Hua et al., 2024, Xu et al., 2023, Liu et al., 2023a], and multimodal self-supervised (Multi-SS) pre-training [Hua et al., 2024] in robust fine-tuning. However, these studies are confined to single architecture types and sizes, which restricts the scope of conclusions. In Section 4, we will see that some conclusions do no not hold uniformly across all architecture sizes and types. Furthermore, this study is the first to compare the performance of pre-training protocols such as supervised (multistep), unimodal self-supervised (Uni-SS), and fusion (i.e., mixture of supervised and self-supervised pre-training) in robust fine-tuning. Understanding how such state-of-the-art pre-training protocols contribute to robust generalization remains a knowledge gap for practitioners.

Summary Based on the considered architectures and pre-training protocols, a set of 40 backbones are selected – see summary in Table 2. A global summary of the considered backbones, including exhaustive references and Hugging Face identifiers is available in Appendix A.

2.2 Fine-tuning protocols

Consider a pre-trained backbone $g_{\theta_1} : X \to L$, where L denotes an arbitrary latent space. Further consider a classifier $h_{\theta_2} : L \to \Delta_C$ consisting of a linear layer followed by a softmax. The goal of fine-tuning is to combine the pre-trained backbone and the classifier together to obtain a final model $f_\theta : X \to \Delta_C$, with $\theta = \{\theta_1, \theta_2\}$. An observation x is associated to a probability prediction $f_\theta(x) = h_{\theta_2}(g_{\theta_1}(x))$. The fine-tuning process consists of E epochs over the training dataset.

Full fine-tuning (FFT) All parameters $\theta = \{\theta_1, \theta_2\}$ are updated for the downstream task. The proposed FFT setup differs from prior works [Hua et al., 2024, Jeddi et al., 2020], who employ a single learning rate across the entire model f_θ. In contrast, our setup allows for distinct learning rates, η_1 and η_2, for g_{θ_1} and h_{θ_2}, respectively, as well as separate weight decay parameters, γ_1 and γ_2.

Linear probing (LP) Only the classifier layer h_{θ_2} is updated, while the parameters of the feature extractor are frozen. The learning rate is η_2 and the weight decay γ_2.

We consider three fine-tuning protocols: FFT with $E = 50$ epochs (denoted FFT-50), FFT with $E = 5$ epochs (denoted FFT-5), and LP with $E = 50$ epochs (denoted LP-50). These protocols represent different trade-offs between compute and parameter efficiency. Specifically, LP-50 is parameter-efficient (few trainable weights), while FFT-5 is compute-efficient (short training duration). We do not include LP with 5 epochs as it would combine both constraints and would be very restrictive. Although FFT and LP have been compared before [Hua et al., 2024, Xu et al., 2023, Liu et al., 2023a], there is limited understanding of the compute-efficient setting (FFT-5) and of how design choice combination correlates with performance across the fine-tuning protocols.

Practical considerations. Our choice of 50 training epochs is motivated by prior robust fine-tuning works who employ 40 [Hua et al., 2024] to 60 epochs [Xu et al., 2023, Liu et al., 2023a]. Other (non-robust) fine-tuning benchmarks have considered more epochs (e.g., 100 epochs in Goldblum et al. [2024]) but they are not specifically focused in the low data regime setting. Additional technical details regarding the optimization of hyper-parameters are provided in Appendix B.

2.3 Loss objectives

We consider two loss objectives, namely Classic AT and TRADES [Zhang et al., 2019] which are widely popular [Wang et al., 2023, Croce et al., 2020]. There is no consensus as to which loss to choose to perform robust fine-tuning, as suggested by inconsistent design decisions in the literature (e.g., Classic AT in Hua et al. [2024], Singh et al. [2024], TRADES in Xu et al. [2023]). Although Liu et al. [2023b] identify TRADES as most effective, their findings are based on models trained from scratch, which differs from fine-tuning where pre-trained backbones play a central role.

Crafting synthetic adversarial perturbations Both Classic AT and TRADES require crafting synthetic adversarial perturbations throughout training. Given an observation (x, y) and a classifier f_θ, consider the perturbation x' given by the following maximization problem:

$$\arg \max_{x' \in \mathbb{B}(x,\epsilon,p)} \mathcal{L}_{\texttt{CE}} \left(f_\theta(x'), y \right), \tag{1}$$

where $\mathcal{L}_{\texttt{CE}}$ denotes the cross-entropy loss and $\mathbb{B}(x, \epsilon, p) = \{x \in X : \|x' - x\|_p \leq \epsilon\}$ is the ℓ_p-ball around x. Projected Gradient Descent (PGD-K) [Madry et al., 2018] with K iterations finds an approximate solution x'_K to the perturbation x' resulting from Eq. 1. Specifically, PGD-K corresponds to starting from $x'_0 = x$ and to iteratively apply the update rule

$$x'_{k+1} = \Pi_{\mathbb{B}(x,\epsilon,p)} \left(x'_k + \delta \texttt{sign} \left(\nabla_x \mathcal{L}_{\texttt{CE}} \left(f_\theta(x'_k), y \right) \right) \right), \quad k = 0, \ldots, K-1 \tag{2}$$

where $\delta \geq 0$ is the step size and $\Pi_{\mathbb{B}(x,\epsilon,p)}$ is the projection operator to ensure that the perturbed input remains within the ℓ_p-ball. The APGD-K perturbation [Croce and Hein, 2020] improves upon PGD-K by automatically adapting the step size δ, removing the need for manual tuning.

Classic adversarial training (Classic AT) Corresponds to training a classifier $f_\theta(\cdot)$ using the cross-entropy loss $\mathcal{L}_{\texttt{CE}}$ on observations perturbed by APGD-K. This corresponds to minimizing the loss $\mathcal{L}_{\texttt{AT}}(x, y) := \mathcal{L}_{\texttt{CE}}(f_\theta(x'_K), y)$ over θ.

TRADES Corresponds to training a classifier $f_\theta(\cdot)$ with the TRADES loss objective [Zhang et al., 2019]. This corresponds to minimizing the following loss over θ:

$$\mathcal{L}_{\texttt{TRADES}}(x, y) := \mathcal{L}_{\texttt{CE}}(f_\theta(x), y) + \beta \, \texttt{KL}(f_\theta(x) \| f_\theta(x'_K)), \tag{3}$$

where the scalar $\beta \geq 0$ controls the trade-off between cross-entropy and the Kullback–Leibler (KL) divergence of predictions on perturbed and unperturbed inputs.

Practical considerations To facilitate comparison between Classic AT and TRADES loss objectives, we always consider the same process to craft synthetic adversarial perturbations, namely APGD-K with $K = 10$, $\epsilon = 4/255$, and bounded with respect to the ℓ_∞-norm. Additionally, the training data is augmented regardless of the loss objective using standard augmentation techniques (see Appendix B for more details).

3 Threat definition and evaluation methods for robust generalization

Although we set a specific type of adversarial perturbation for the optimization strategy (i.e., APGD-K for ℓ_∞-norm), deploying machine-learning systems reliably and responsibly requires generalization to diverse, unknown and evolving types of perturbations. We now define additional perturbation types that the model will face at test time, to evaluate robust generalization.

3.1 Threat model at test time

To study robust generalization, we define the *threat model* which specifies the possible perturbation types faced by the model at test-time [Akhtar and Mian, 2018]. We use the notation $\mathcal{T}_X(z)$ to denote the distribution of observations drawn from X that contain a perturbation of type z. We consider a finite set of perturbation types noted τ, where each type can be categorized into *adversarial* and *common* perturbations.

The adversarial perturbations are bounded by a scalar ϵ with respect to the ℓ_p-norm, i.e. $x \sim X, x' \sim \mathcal{T}_X(z)$ such that $\|x - x'\|_p \leq \epsilon$. We include three adversarial perturbation types, generated from $p = 1, 2, \infty$, and $\epsilon = 75.0, 2.0, 4/255$, respectively. The values for ϵ are standard choices in robustness benchmarks [Croce et al., 2020, Singh et al., 2024]. In contrast, the common perturbations reflect unfortunate events that commonly occur in vision systems (e.g. noise, blur, contrasts, digital format compressions, etc) and that hamper the predictive performance [Jung, 2018].

In summary, the threat model is $\mathcal{T}_X(z), z \in \tau = \{\emptyset, \infty, 1, 2, \texttt{common}\}$, where τ comprises five perturbation types: no perturbations (i.e., clean observations, noted $\emptyset$), adversarial perturbations under the ℓ_1, ℓ_2, and ℓ_∞ norm (noted 1, 2 and ∞), and common perturbations (noted `common`). Appendix C describes how we generate these test-time perturbations using open-source software such as AutoAttack [Croce and Hein, 2020] and Jung [2018].

3.2 Evaluating robust generalization

We measure performance against the threat model using the *accuracy*, which corresponds to the total number of correct predictions over the total number of observations in the test dataset. We adopt accuracy for its interpretability and widespread use in prior works Tang et al. [2021], Liu et al. [2023b], Shao et al. [2021] and robustness competitions Croce et al. [2020]. Recall that a configuration is the combination of a pretrained backbone and a loss objective that results into a classifier model. For each of the three fine-tuning protocols considered (FFT-50, FFT-5 and LP-50), we evaluate the performance of each configuration as follows. For every configuration $i \in \{1, \ldots, I\}$ on dataset $d \in \{1, \ldots, D\}$ we obtain a predictive accuracy score $a_{i,d}(z) \in [0, 1]$ for each perturbation type $z \in \tau$. Let $\mathbf{a}_{i,d} := \left[a_{i,d}(z_1), \ldots, a_{i,d}(z_{|\tau|})\right]$ denote the vector of predictive accuracies of configuration i on dataset d.

Borda score We use the Borda score to compare the relative performance of various configurations on the same fine-tuning protocol. Consider any pair $v = (d, z)$, consisting of a dataset d and a perturbation type z as a voter. Let $V = D \times |\tau|$ be the set of all voters. To each voter $v = (d, z)$ corresponds a function $\texttt{rank}_v : I \to \{1, \ldots, |m_v|\}$ that ranks the configurations $i \in I$ based on their score $a_{i,d}(z)$, in decreasing order. The configuration i_{top} with best performance gets rank 1 (i.e., $\texttt{rank}_v(i_{\text{top}}) = 1$) and the worst one gets rank m_v. We have $m_v \leq |I|$ to account for the possibility of equal scores (and so equal ranks). Then, the Borda score for each configuration $i \in I$ is defined by $B(i) := \sum_{v=1}^{V} m_v - \texttt{rank}_v(i)$.

Sum score To account for absolute performance and to compare configurations across different fine-tuning protocols we use the *Sum score*. For each configuration $i \in I$ the sum score is defined by $S(i) := \sum_{(d,z) \in V} a_{i,d}(z)$. By summing the accuracy scores across all perturbation types and datasets, the sum score rewards peak performance even when the accuracy is inconsistent. This contrasts with the Borda score that penalizes inconsistent performance through ranking degradation.

Mean Absolute Correlation For each dataset d, we define a $|\tau| \times |\tau|$ Spearman correlation matrix $\mathbf{C}^{(J,d)}$ computed over the accuracy vectors $\{\mathbf{a}_{i,d}\}_{j=1}^{J}$ associated to a subset of configurations $J \subseteq I$. The subset J can represent all the configurations ($J = I$) or a subset with a common

specific characteristic (e.g. architecture type, etc). The *mean absolute correlation* for dataset d on the subset of configurations J, is noted $\texttt{MAC}^{(d,J)}$, and is given by $\frac{1}{|\tau|(|\tau|-1)} \sum_{l \neq k} |C^{(J,d)}_{l,k}|$. The $\texttt{MAC}^{(d,J)}$ is the average absolute off-diagonal correlation between all pairs of perturbation types on dataset d for the subset of configurations in J. A high $\texttt{MAC}^{(d,J)}$, close to 1, indicates that, on average, the performance across all perturbation types is strongly related, suggesting more consistent or uniform robust generalization. Lower $\texttt{MAC}^{(d,J)}$ values indicate that on average the performance across all perturbation types is less correlated, implying that robustness may be specific to certain perturbations rather than uniform. We also compute a global average across datasets: $\texttt{MAC}^{(\texttt{avg},J)} = \frac{1}{D} \sum_{d=1}^{D} \texttt{MAC}^{(d,J)}$. This informs us on the strength of the robust generalization pattern across datasets.

4 Results

We select 6 datasets that fit in the *low data regime* (details in Appendix B). We consider five datasets from the natural image domain (**Caltech101** [Fei-Fei et al., 2004], **Aircraft** [Maji et al., 2013], **Flowers** [Nilsback and Zisserman, 2008], **Oxford pet** [Parkhi et al., 2012], **Stanford cars** [Krause et al., 2013]) and one from the satellite imagery domain (**Land-Use** [Yang and Newsam, 2010]).

We benchmark 240 design combinations (40 pre-trained backbones $\times$ 2 robust losses $\times$ 3 fine-tuning protocols) over 6 datasets, totaling $1,440$ evaluated configurations.

The hyper-parameters of each configuration are independently optimized (details are reported in Appendix B). Each configuration is tested against 5 perturbation types, unseen during (pre-) training (Section 3.1), resulting in $7,200$ robustness measurements. To our knowledge, this benchmark includes the most diverse and comprehensive set of design choices in the robust fine-tuning setting. Collected measurements, and code are open-sourced[1].

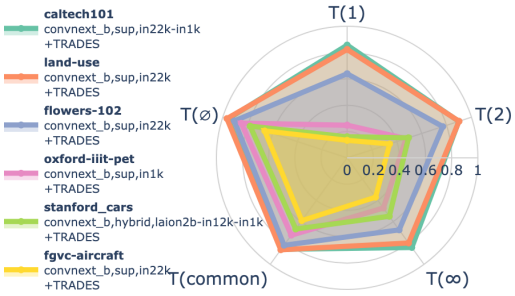


Figure 2: Robust generalization of the best configuration per dataset using FFT-50 (Borda score).

4.1 Which configurations perform best?

Best performing configurations overall. Table 3 reports the best performing configurations in FFT-50. We see that the best performing backbone is convolutional (*Convnext (base), with supervised pretraining on Imagenet-22k, using TRADES*). FFT-50 clearly outperforms other fine-tuning protocols, with a best sum score of 19.79, which is 61% higher than FFT-5 and 53% higher than LP-50 (see Table 8 and 10 in Appendix). In the Appendix, Tables 7, 9, and 11 report the ranking of all the configurations across FFT-50, FFT-5, and LP-50 respectively.

Best performing configurations per dataset. Figure 2 displays the best performing configuration per dataset, when using FFT-50. We observe that convolutional architectures outperform other options on all considered datasets. Despite growing attention on the robustness of attention-based architectures [Bai et al., 2021, Liu et al., 2023b, Shao et al., 2021], our findings show that the robust generalization capacity of well tuned convolutional architectures should not be underestimated. Additionally, on two datasets (Caltech101, and Land-Use), the best configurations achieve accuracy above 0.8 on all perturbation types, demonstrating strong robust generalization. This performance is remarkably high for the field [Croce et al., 2020], demonstrating the practical potential of robust fine-tuning and the importance of carefully identifying best design choices.

Low-cost proxies exist in robust fine-tuning. Given the evolving set of available design choices, practitioners need to be equipped with low-cost tools to rapidly identify design choices that are more promising than others. The identification of low-cost proxies helped practitioners in natural language

[1] https://anonymous.4open.science/r/robust_training-636C/README.md

modeling [Zhu et al., 2022], and can also benefit costly protocols such as robust learning. Between LP-50 and FFT-5, we find that LP-50 is the most reliable low-cost proxy to FFT-50 (see Figure 8a), especially when using TRADES over Classic AT. Indeed, the correlation between LP-50 and FFT-50 using TRADES is the highest.

Size	Gold (1st)	Silver (2nd)	Bronze (3rd)
small	coat_t,sup,in1k, TRADES (GR:18, BS:1653, SS:15.83)	edgenetx_s,sup,in1k, TRADES (GR:23, BS:1552, SS:14.66)	edgenetx_s,sup,in1k, Classic AT (GR:33, BS:1356, SS:12.88)
medium	convnext_t,sup,in22k-in1k, TRADES (GR:14, BS:1773, SS:16.49)	convnext_t,sup,in1k, TRADES (GR:15, BS:1681, SS:15.6)	convnext_t,sup,in22k, TRADES (GR:20, BS:1650, SS:15.07)
large	convnext_b,sup,in22k, TRADES (GR:1, BS:2281, SS:19.79)	coatnet_2,sup,in12k-in1k, TRADES (GR:2, BS:2127, SS:18.74)	coatnet_2,sup,in12k, TRADES (GR:3, BS:2116, SS:18.87)

Table 3: Top FFT-50 configurations, with global ranking (GR) based on Borda score (BS), sum score (SS) also reported below.

4.2 Design Choices Favoring TRADES in Robust Fine-Tuning.

Overall, TRADES outperforms Classic AT. It has been shown previously that TRADES outperforms Classic AT when training from scratch [Liu et al., 2023b]. Our results show that these conclusions hold in the fine tuning setting as well (see Figure 4 in the Appendix). We next extend these results by identifying strong interactions between the loss and other design choices in the FFT-50 setting (see Figure 3). The identification of such interactions with TRADES is particularly valuable, given its frequent association with state-of-the-art performance on robustness benchmarks [Wang et al., 2023, Croce et al., 2020].

TRADES interacts positively with architecture size. On average, TRADES achieves higher returns compared to Classic AT when architecture size grows (see Figure 3a). Additionally, the odds ratio of TRADES outperforming Classic AT increases steeply with architecture scale, which is a significant effect in FFT-50 (see Figure 5 in Appendix E). These results suggest that TRADES is a promising approach to improve the robustness of large systems, a setting where Classic AT is currently the preferred approach [Wang et al., 2024]. Existing implementations of TRADES require the storage of two forward passes in memory, which motivates an avenue to improve this algorithmic limitation to fully reveal the potential of TRADES on large architectures.

TRADES interacts best with convolutional and hybrid architectures. While TRADES and Classic AT yield equivalent outcomes (similar mean Borda score) for attention-based architectures, convolutional and hybrid architectures benefit most from using TRADES over Classic AT (see Figure 3b). Since convolutional architectures tend to overfit more local features and patterns [Bhojanapalli et al., 2021], this result suggests that TRADES regularizes convolutional architectures more efficiently than Classic AT in computer vision tasks.

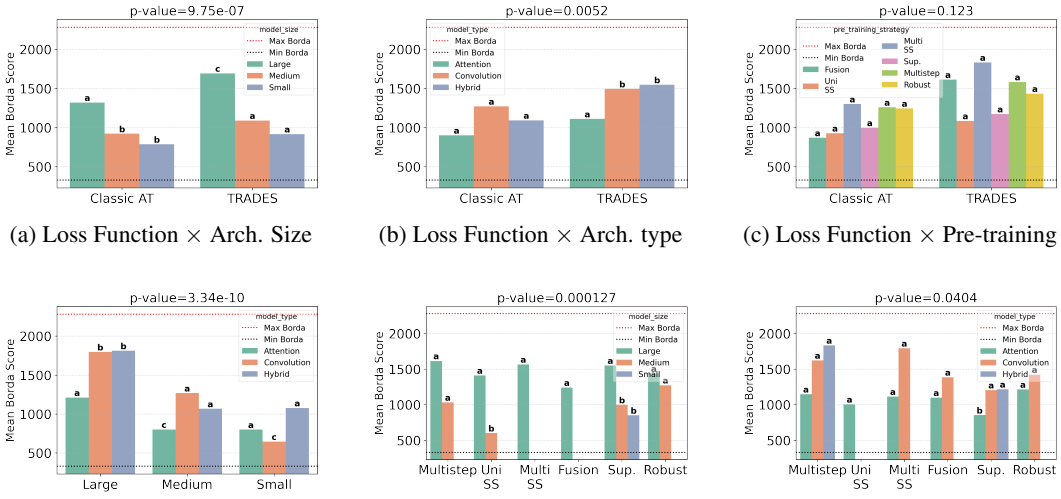
4.3 Distinct robust generalization patterns across architectures sizes and types.

Larger architectures are consistently better. Large architectures clearly outperform medium and small architectures in FFT-50 (see Table 3) and generalize better (see Table 7). Large convolutional and hybrid architectures outperform attention-based ones on average (see Figure 3d), though attention models may show their full potential at larger scales [Wang et al., 2024]. Because model scale is often subject to limitations in practice, we also provide analysis at specific architecture sizes to guide practitioners with such limitations.

If constrained to small architectures, hybrid architectures are the best option. Among small architectures, hybrid architectures achieve significantly higher scores than fully convolutional ones (see Figure 3d). Using TRADES loss, *Coat (tiny)* and *EdgeNetx* rival larger architectures and achieve impressive rankings of GR:18 and GR:23, corresponding to tier-1 performance among 80 configurations (see Table 3). Prior works on robustness are generally focused on larger architectures [Liu et al., 2023b], but this result extends knowledge by highlighting the practical potential of hybrid architectures for robust fine-tuning using small architectures on data scarce regimes. The result also

constitutes valuable motivations for the community that supports hybrid architectures [Dai et al., 2021, Maaz et al., 2022, Dai et al., 2021].

Promising generalization properties of hybrid architectures. Table 4 shows that hybrid architectures achieve the highest MAC values compared to attention and convolutional architectures in FFT-50. This finding complements prior work demonstrating strong robustness of hybrid architectures trained from scratch [Liu et al., 2023b], extending these conclusions to the robust fine-tuning setting. We also generalize the observation across diverse scales and types of hybrid architectures, while the previous observation held only for CoatNet (16M). Finally, Table 4 provides a precise characterization on the robust generalization capability of hybrid architectures, beyond accuracy score.



(a) Loss Function × Arch. Size (b) Loss Function × Arch. type (c) Loss Function × Pre-training

(d) Arch. Type × Size (e) Pre-training × Arch. Size (f) Pre-training × Arc. Type

Figure 3: Nested Welch's ANOVA of the form A × B testing the main effect of A and how the effect of B varies within each level of A (p-value on top). Post-hoc groupings from Tukey HSD tests are annotated with letters above the bars: bars with different letters belong to significantly different groups, at the 90% confidence level. Results for FFT-50.

4.4 Influence of the pre-training strategy on robust generalization

Multi-modal self-supervised pretraining is beneficial for convolutional architectures. In FFT-50, the *Convnext (base)* architecture with multimodal self-supervised pre-training using the TRADES loss achieves the fourth best ranking in terms of Borda score, and top-2 in terms of sum score (see Table 7 in Appendix E). This performance is surprising given prior results [Hua et al., 2024] showing that supervised and robust pre-training often outperform Multi-SS. However, prior works focus on Multi-SS of attention-based architectures, while the reported performance improvement targets Multi-SS on convolutional architectures (see Figure 3e). The potential of Multi-SS pre-training is further evidenced in Figure 3c, where Fusion (which is based on Multi-SS) pre-training achieves second best average performance, behind Multi-SS.

Robust pre-training performs best in constrained fine-tuning protocols. In FFT-5, the global gold (global rank GR:1), silver (GR:2), and bronze (GR:3) are achieved with robust pretraining (see Table 8 in Appendix E). Similarly, the top-3 in LP-50 (GR:1,2 and 3) are also achieved with robust pre-training (see Table 10 in Appendix E). Note that this competitive performance does not hold in the less constrained FFT-50 protocol, where the best configuration based on robust pre-training achieves a global ranking of 9 (see Table 7). Our results are aligned with prior results showing that robust pre-training helps in parameter-efficient settings such as low-rank adaptation [Xu et al., 2023, Liu et al., 2023a] and linear probing [Hua et al., 2024]. We further show that robust pre-training is also beneficial in fine-tuning protocols constrained on the number of updates, a setting not covered by prior works.

Robust pre-training of larger architectures (may) have limited returns. Given its high computational cost, robust pre-training prompts critical evaluation of its return on investment relative to alternative pre-training protocols. The performance gains from medium to large architectures are relatively modest for robust pre-training (see Figure 3e). The rate of improvement [2] is $+12\%$ for robust pre-training, whereas it reaches $+43\%$ for supervised, $+44\%$ for multi-step, and $+80\%$ for Uni-SS in FFT-50. This relatively low gain is due to already high performance of robust pre-training at the medium scale, leaving less room for improvement at the larger scale. Although recent works have emphasized scaling robust pretraining to large architectures Singh et al. [2024], there are currently no robust pre-trained small architectures. This finding suggests that robust pre-training at smaller architecture scales could be a promising and underexplored direction for future research.

Influence of loss objective switches between robust pre-training and robust fine-tuning. When specifically considering robust pre-trained architectures, a question for practitioners is: *should we use the same robust loss objective for fine-tuning as for pre-training?* With FFT-50, configurations that use a different loss objective for robust fine-tuning than for robust pre-training significantly outperform configurations that use the same loss for both phases. While the global mean between both choices are not statistically different (e.g., $p = 0.42$ with the Mann–Whitney test), we observe that switched configurations are strongly overrepresented among top performers across the 6 datasets considered: 5 out of 6 top-1 configurations use a loss switch, with a binomial p-value of $0.041 < 0.05$. Previous works have used switching [Xu et al., 2023, Liu et al., 2023a] and non-switching strategies [Hua et al., 2024]. Our finding provides the first evidence that switching losses between pre-training and fine-tuning can be beneficial with enough compute (result holds only in FFT-50).

		Caltech101	Aircraft	Flowers-102	Oxford-pet	Stanford-cars	Land-use	Global $\texttt{MAC}_{\text{avg}}$
MAC per dataset $\texttt{MAC}_d$ over the 80 configs.		0.847	0.782	0.805	0.681	0.849	0.807	0.795
Loss objective	Classic AT	0.876	0.702	0.842	0.778	0.844	0.890	0.822
	TRADES	0.823	0.884	0.830	0.669	0.896	0.777	0.813
Architecture size	Large	0.825	0.882	0.860	0.859	0.871	0.803	0.850
	Medium	0.743	0.628	0.743	0.436	0.701	0.833	0.681
	Small	0.911	0.531	0.503	0.467	0.575	0.678	0.611
Architecture type	Attention	0.884	0.800	0.761	0.713	0.856	0.882	0.816
	Convolutional	0.762	0.756	0.811	0.574	0.818	0.719	0.740
	Hybrid	0.951	0.815	0.892	0.798	0.889	0.879	0.871
Pre-training protocol	Fusion	0.760	0.760	0.920	0.660	0.920	1.000	0.837
	Uni-SS	0.821	0.902	0.860	0.619	0.878	0.668	0.791
	Multi-SS	0.911	0.742	0.760	0.589	0.855	0.703	0.760
	Supervised	0.937	0.760	0.727	0.682	0.806	0.789	0.783
	Multistep	0.867	0.809	0.846	0.893	0.856	0.798	0.845
	Robust	0.568	0.641	0.841	0.670	0.685	0.877	0.714

Table 4: Summary table of the Mean Absolute Correlation in FFT-50, measured over the 80 configurations as well as on subsets of configurations based on design choices.

5 Conclusion

Among other findings, we find that convolutional architectures perform best for robust fine-tuning in the low-data regime. Despite growing interest in the robustness of attention-based architectures [Bhojanapalli et al., 2021], our study suggests they are more difficult to fine-tune for robustness in practice. Our findings have broader design impacts for vision systems: for example, vision-language models predominantly rely on attention-based backbones [Radford et al., 2021].

Limitations The insights from this study are contingent to the set of datasets, backbones, and optimization strategies considered. We acknowledge that such insights need to continually evolve with the development of new design choices. In this study, the configurations were optimized based on a total compute budget, rather than on an equal number of trials across backbones. This choice reflects the practical reality that some backbones are more challenging to tune due to their compute requirements. This compute-aware tuning approach reflects real-world deployment constraints and promotes energy-conscious model selection [Courty et al., 2024].

[2] Rate measured using the relative change w.r.t. the average of the two scores, and computed as $\frac{\text{large}-\text{medium}}{(\text{large}+\text{medium})/2}$. This rate to ensures a symmetric and unbiased comparison that does not privilege either model scale.

References

Alexander B. Jung. imgaug. https://github.com/aleju/imgaug, 2018. [Online; accessed 30-Oct-2018].

Christian Szegedy et al. Intriguing properties of neural networks. *In Proc. ICLR*, 2014.

David Sculley et al. Hidden technical debt in machine learning systems. *In Proc. NeurIPS*, 2015.

Francesco Croce and Matthias Hein. Adversarial robustness against multiple and single l_p-threat models via quick fine-tuning of robust classifiers. In *International Conference on Machine Learning*, pages 4436–4454. PMLR, 2022.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations, 2019. URL https://arxiv.org/abs/1904.13000.

Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pages 6640–6650. PMLR, 2020.

Pouya Bashivan, Reza Bayat, Adam Ibrahim, Kartik Ahuja, Mojtaba Faramarzi, Touraj Laleh, Blake Richards, and Irina Rish. Adversarial feature desensitization. *Advances in Neural Information Processing Systems*, 34:10665–10677, 2021.

Adam Ibrahim, Charles Guille-Escuret, Ioannis Mitliagkas, Irina Rish, David Krueger, and Pouya Bashivan. Towards out-of-distribution adversarial robustness. *arXiv preprint arXiv:2210.03150*, 2022.

Saba Rahimi, Ozan Oktay, Javier Alvarez-Valle, and Sujeeth Bharadwaj. Addressing the exorbitant cost of labeling medical images with active learning. In *International Conference on Machine Learning in Medical Imaging and Analysis*, volume 1, 2021.

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

Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: A parameter-free automated robust fine-tuning framework, 2023.

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

Ziquan Liu, Yi Xu, Xiangyang Ji, and Antoni B. Chan. Twins: A fine-tuning framework for improved transferability of adversarial robustness and generalization, 2023a.

Aleksander Madry et al. Towards deep learning models resistant to adversarial attacks. *In Proc. ICLR*, 2018.

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

Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, et al. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv preprint arXiv:2109.05211*, 2021.

Aishan Liu, Shiyu Tang, Siyuan Liang, Ruihao Gong, Boxi Wu, Xianglong Liu, and Dacheng Tao. Exploring the relationship between architectural design and adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4096–4107, 2023b.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

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*, pages 2661–2671, 2019.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Rishika Bhagwatkar, Shravan Nayak, Pouya Bashivan, and Irina Rish. Improving adversarial robustness in vision-language models with architecture and prompt design. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17003–17020, 2024.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Zeyu Wang, Xianhang Li, Hongru Zhu, and Cihang Xie. Revisiting adversarial training at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24675–24685, 2024.

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.

Ahmadreza Jeddi, Mohammad Javad Shafiee, and Alexander Wong. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *arXiv preprint arXiv:2012.13628*, 2020.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pages 36246–36263. PMLR, 2023.

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.

Naman Deep Singh, Francesco Croce, and Matthias Hein. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. doi: 10.1109/CVPR.2004.383.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in neural information processing systems*, 34:26831–26843, 2021.

Zining Zhu, Soroosh Shahtalebi, and Frank Rudzicz. Predicting fine-tuning performance with probing. *arXiv preprint arXiv:2210.07352*, 2022.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10231–10241, 2021.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=dUk5Foj5CLf.

Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *International Workshop on Computational Aspects of Deep Learning at 17th European Conference on Computer Vision (CADL2022)*. Springer, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. URL https://doi.org/10.5281/zenodo.11171501.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773. If you use this software, please cite it as below.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=4nPswr1KcP.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

Christoph et al. Schuhmann. Laion-5b: An open large-scale dataset for training next generation image–text models. *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022. arXiv:2210.08402.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

Ross Wightman, Hugo Touvron, and Herve Jegou. Resnet strikes back: An improved training procedure in timm. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021. URL https://openreview.net/forum?id=NG6MJnVl6M5.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vh-0sUt8HlG.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

Tal Ridnik, Hussam Lawen, Emanuel Ben-Baruch, and Asaf Noy. Solving imagenet: a unified scheme for training any backbone to top results. *arXiv preprint arXiv:2204.03475*, 2022.

Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9981–9990, October 2021.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL http://jmlr.org/papers/v18/16-558.html.

S. Marcel and R. Rodriguez. Torchvision image transformations. https://pytorch.org/vision/stable/transforms.html, 2016. Accessed: 2025-05-04.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The topic and claims in the abstract are accurately reflected in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the Limitations paragraph in the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper explains the design choices made to obtain the results. The code and data is open-sourced. All the code to generate paper's figures is open-sourced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released anonymized Github, with all the instructions to re-create the environment, download datasets and models, and their preparation, and how to launch experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the code to reproduce the data splits. All hyper-parameters used to train each configuration are also open-sourced in the configs folder on the codebase.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The results in the paper have been tested for statistical significance using p-values whenever possible. The thresholds for statistical confidence are also reported in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details on the compute setup are reported in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research addresses the safety and reliability of machine learning models under distribution shifts and adversarial perturbations—key concerns under the NeurIPS Code of Ethics. By analyzing robust generalization across architectures and pretraining strategies, our work contributes to the development of more trustworthy and secure AI systems.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impacts are discussed in the first paragraph of the introduction. The technical impacts are discussed in the Conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper is based on already open-sourced and widely used tools.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have bibliographical references for all the datasets, and backbones, and open source software used for this study.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We provide all the code base to reproduce the study and the dataset of the collected results. These support the study but are not new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper is based on open sourced datasets and backbones.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No research with human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM use does not impact core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.