On the Robustness Tradeoff in Fine-Tuning

Kunyang Li, Jean-Charles Noirot Ferrand, Ryan Sheatsley, Blaine Hoak,
Yohan Beugin, Eric Pauley, Patrick McDaniel
University of Wisconsin-Madison
{kli253, jcnf, sheatsley, bhoak, ybeugin, epauley, mcdaniel}@cs.wisc.edu

Abstract

Fine-tuning has become the standard practice for adapting pre-trained models to downstream tasks. However, the impact on model robustness is not well understood. In this work, we characterize the robustness-accuracy trade-off in fine-tuning. We evaluate the robustness and accuracy of fine-tuned models over 6 benchmark datasets and 7 different fine-tuning strategies. We observe a consistent trade-off between adversarial robustness and accuracy. Peripheral updates such as BitFit are more effective for simple tasks—over 75% above the average measured by the area under the Pareto frontiers on CIFAR-10 and CIFAR-100. In contrast, fine-tuning information-heavy layers, such as attention layers via Compacter, achieves a better Pareto frontier on more complex tasks—57.5% and 34.6% above the average on Caltech-256 and CUB-200, respectively. Lastly, we observe that the robustness of fine-tuning against out-of-distribution data closely tracks accuracy. These insights emphasize the need for robustness-aware fine-tuning to ensure reliable real-world deployments.

1. Introduction

Pre-training and fine-tuning can efficiently transfer knowledge from upstream data to downstream tasks [2, 44]. Models can be fine-tuned in various ways [27]—full fine-tuning, linear probing (training the classification head), and parameter-efficient fine-tuning (PEFT) [5, 10, 14, 16, 17, 28–30, 34, 49]. Specifically, PEFT strategies selectively update parameters to achieve high accuracy on target tasks with significantly reduced computation and storage costs.

While fine-tuning optimizes efficiency and accuracy, its impact on model robustness remains underexplored–and in fact relatively unknown. Attackers actively generate adversarial examples [36, 40] by adding crafted perturbations to cause model misclassification. A model's ability to resist these samples, as well as perform well when presented with out-of-distribution (OOD) data, is called model robustness.

Prior studies on adversarial robustness [3, 7, 11, 36, 40] focus on attacking models that are trained from scratch. Models learn to use high-dimensional features during training. Since test data have similar features to training data, models are able to achieve high accuracy at evaluation. However, attackers can generate adversarial examples to largely degrade accuracy by perturbing those features, which we refer to as highly predictive, non-robust features [20, 45] in this paper. In comparison, while attacks are targeted at downstream phenomena [4, 6, 18, 22, 39], two data phenomena are involved in pre-training and fine-tuning—upstream and downstream data, respectively. Here, a key question arises: *how does adversarial robustness vary as the model is fine-tuned?* We hypothesize that features learned from pre-trained data are more robust against downstream attacks. As models fit to downstream phenomena, they learn non-robust features to gain accuracy while sacrificing adversarial robustness. Additionally, the fundamental trade-off between accuracy and adversarial robustness [31, 45, 51] may still exist but is potentially shifted.

This work is the first to investigate deeply how robustness is impacted by fine-tuning. Here, we focus on three questions: (1) does the adversarial robustness-accuracy trade-off exist during fine-tuning? (2) how sensitive is this trade-off to different fine-tuning strategies and downstream data distributions? and (3) do findings on adversarial robustness generalize to OOD robustness? We begin by formally exploring the potential interaction between robustness and fine-tuning (*i.e.*, the impact of mechanisms and data phenomena). Thereafter, we evaluate trade-offs empirically by fine-tuning a test suite of models using a range of fine-tuning strategies (see Figure 1). The experiments are performed over 231 models, 7 fine-tuning methods, and 6 benchmark datasets (5 for adversarial robustness and 1 with 6 domains for OOD robustness), resulting in approximately 2,100 adversarial and 2,000 OOD robustness assessments.

The evaluation finds: (1) a consistent adversarial robustness-accuracy trade-off early (within the first 3 epochs) in fine-tuning across all methods—initially, robustness improves together with accuracy, then it peaks and declines as fine-tuning continues; (2) the Pareto frontiers (*i.e.*,

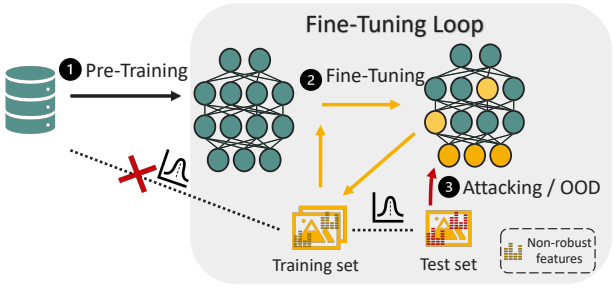

Figure 1. Continuous robustness evaluation during fine-tuning.

optimal trade-off curves) are sensitive to fine-tuning methods and downstream task complexity—fine-tuning methods modifying intermediate information-intense layers, such as attention layers (*e.g.*, Compacter [37]), achieve better balances than those updating excessively (*e.g.*, full fine-tuning) or only peripheral layers (*e.g.*, linear probing, BitFit [49]); and (3) OOD robustness does not exhibit a trade-off with accuracy but remains relatively stable and closely tracks accuracy, suggesting that different underlying mechanisms drive robustness in security and safety contexts. These findings deepen our understanding of designing robustness-aware fine-tuning strategies facing different tasks and risks.

2. Background

2.1. Fine-tuning Strategies

ViT backbone. The transformer architecture [46] has become the state-of-the-art across many fields [2, 8, 9]. It typically serves as the backbone structure in the pre-training and fine-tuning paradigm [44], where general knowledge of pre-trained models is transferred to solve specific tasks through fine-tuning on (often small) downstream datasets. In this study, we focus on vision transformers (ViT), thus on images. Here, an input image is divided into fixed-size patches, flattened, and projected into embeddings. Then attention scores are calculated to determine the relationship between patches to capture global dependencies. The outputs are then passed through a feedforward network (FFN), in-between two layer normalizations (LN) for stability.

Fine-tuning Methods. Full fine-tuning, which updates all model parameters, is widely used to transfer knowledge from pre-trained models to downstream tasks in computer vision [27, 50]. While effective, it can be computationally expensive, especially for large-scale models. In contrast, linear probing, which fine-tunes only the final classification layer, is more efficient but often fails to match the performance of full fine-tuning [27]. To bridge this gap, parameter-efficient fine-tuning (PEFT) methods have been developed, aiming to achieve comparable or higher accuracy with fewer trainable parameters and reduced memory and storage overhead [15, 32].

PEFT techniques primarily retain the pre-trained parameters while introducing a small set of trainable parameters to adapt the model to the downstream task. A general formulation can be expressed as:

$$\hat{y} \leftarrow W_0 x + \Delta W x, \tag{1}$$

where W_0 represents the frozen pre-trained weights, and ΔW denotes the task-specific learnable parameters introduced by a given PEFT method. The input x is processed through the model, and the prediction $\hat{y}$ is computed based on all parameters. Several PEFT techniques have been adapted for ViTs from the original transformer models in NLP [1]. LoRA [17] reduces computational cost by factorizing weight updates into low-rank matrices within attention layers, effectively capturing task-specific adaptations. BitFit [49] takes a more selective approach by updating only bias terms, leaving all other weights unchanged. Adapter-based methods [16] introduce small trainable modules between transformer layers to inject task-specific information without modifying the backbone. Compacter [37] further improves efficiency by using Kronecker-based parameterization within adapters, reducing the number of additional parameters needed. Finally, $(IA)^3$ [34] fine-tunes the model by learning per-layer multiplicative reweighting factors, modifying activations without directly changing pre-trained weights. A detailed visualization of how they are applied to a ViT block is shown in Figure 2.

2.2. Model Robustness

Model robustness is crucial for evaluating the reliability of machine learning models under both adversarial manipulations and natural distribution shifts. Adversarial robustness [3, 11, 36, 40] focuses on a model's ability to defend against adversarial examples—carefully crafted perturbations that are imperceptible to humans but lead to incorrect model predictions. A widely used benchmark for security evaluation, projected gradient descent (PGD), iteratively modifies inputs to maximize model loss L while constraining the perturbed image $x + \delta$ within a predefined norm-ball $\mathcal{B}$ of radius ϵ centered at the original input x as shown in Equation 2.

$$x_{adv} = \text{argmax}_{x+\delta \in \mathcal{B}_\epsilon(x)} L(x + \delta, y) \tag{2}$$

Beyond adversarial threats, models are also expected to demonstrate robustness to out-of-distribution (OOD) data, which is important to ensure reliable performance in real-world settings [26, 41]. OOD shifts can vary in nature, from entirely novel objects absent in training data to more subtle domain variations, such as background changes or stylistic transformations (e.g., sketches versus real images). This study focuses on the latter, where the object of interest remains the same but appears in a different context.

3. Methodology

In this section, we construct two artifacts to study the relationship between robustness and fine-tuning. We begin by extending an existing model of training to explore the potential impacts of fine-tuning on robustness, and then construct an evaluation framework to (a) decompose the space of fine-tuning strategies and (b) analyze how robustness varies with fine-tuning methods (reported in Section 4).

3.1. Modeling Robustness

We establish the problem setup, based on prior work on model robustness [45], to understand how fine-tuning affects robustness. Consider a binary downstream classification task, where labels are uniformly distributed—$y \overset{u.a.r}{\sim} \{-1, +1\}$. Each input x consists of a feature, x_1, strongly-correlated to the corresponding ground-truth label (*i.e.*, robust), and d weakly-correlated (*i.e.*, non-robust) features:

$$x_2, ..., x_{d+1} \overset{i.i.d}{\sim} \mathcal{N}(\eta y, 1). \tag{3}$$

Here, η represents the mean shift of the weakly correlated features, quantifying their predictive power. A larger η implies that these features contribute more information toward classification, while a smaller η means they are less distinguishable from noise.

In fine-tuning, the classifier has frozen, pre-trained weights w_0 and adaptive weights Δw. Here, k parameters are updated, where $k = \|\Delta w\|_0$, $d = \|w_0\|_0$, and $k \ll d$. A simple linear classifier is defined as:

$$f_{FT}(x) := \text{sign}((w_0 + \Delta w)^\top x), \tag{4}$$

where

$$w_0 = [0, \frac{1}{d}, ..., \frac{1}{d}], \Delta w = [0, \frac{1}{d}, ..., 0, ..., \frac{1}{d}]. \tag{5}$$

Then, we derive a lower bound on η to analyze the robustness impact of fine-tuning:

$$\begin{aligned} \Pr[f_{FT}(x) = y] &= \Pr[\text{sign}((w_0 + \Delta w)^\top x) \cdot y > 0] \\ &= \Pr[\text{sign}(\sum_{i=1}^{d} \frac{1}{d} x_i + \sum_{i=1}^{k} \frac{1}{d} x_i) \cdot y > 0] \\ &= \Pr[\mathcal{N}(\frac{k+d}{d}\eta, \frac{k+d}{d^2}) > 0]. \end{aligned} \tag{6}$$

Assuming a fine-tuned classifier achieves 99% accuracy, we obtain (from the standard normal (Z) table):

$$\eta \geq \frac{2.33}{\sqrt{k+d}} \tag{7}$$

This shows that the required correlation strength η of non-robust features depends on both k and d. For full fine-tuning (*i.e.*, $k = d$), this simplifies to $\eta_{\text{full}} \geq \frac{2.33}{\sqrt{2d}}$, which relaxes its lower bound. Here, fine-tuning the entire model allows it to learn a larger number of non-robust features jointly to achieve high accuracy. But each feature is even less correlated to ground-truth, and thus, the model becomes more vulnerable. Additionally, the lower bound is tightened if the downstream task is simpler (*i.e.*, smaller d), such as tasks with well-separated classes or fewer features. In this case, the model requires those non-robust features to have comparatively higher correlations. Thus, they are less susceptible to adversarial perturbations.

These preliminary results suggest that k, which is related to fine-tuning methods, and d, which is related to downstream task complexity, are connected to adversarial robustness. It motivates further exploration on measuring and studying those relationships.

3.2. Measuring Robustness

3.2.1. Decomposition of PEFTs

To investigate the impact of fine-tuning on robustness, we select seven state-of-the-art fine-tuning strategies for our decomposition, including five PEFT methods that span all three main categories [32]: addition-based (*i.e.*, inserting new parameters: Adapter [16], Compacter [37], and IA^3 [34]), reparametrization-based (*i.e.*, decomposing into low-rank matrices: LoRA [17]), and selection-based (*i.e.*, modifying pre-trained weights: BitFit [49]), as well as full fine-tuning and linear probing.

We decompose PEFT methods along two dimensions: a) the *type* of information extracted from the pre-trained model and b) the *mechanisms* used to fine-tune the extracted information. As illustrated in Figure 2, we map out where PEFT strategies are applied within a ViT block and the underlying mechanisms they use. Since the addition of full fine-tuning and linear probing to the visualization is straightforward, we focus on decomposing PEFT methods here.

The knowledge models gain during fine-tuning depends on what information PEFT methods extract from the pre-trained model. It includes the information type (*i.e.*, model weights or representations) and its location. Here, weights correspond to static parameters in the pre-trained model layers, whereas intermediate representations are dynamic and dependent on input data. For example, LoRA [17] explicitly modifies model weights by decomposing attention matrices into low-rank matrices, whereas $(\text{IA})^3$ [34] introduces vectors to scale intermediate representations after the attention layer. Beyond distinguishing between weights and representations, we also examine where these modifications occur within the model. Different PEFT strategies target specific layers, such as attention weights, feed-forward networks (FFNs), or biases as shown in Figure 2.

Once information is extracted, PEFT strategies use specific mechanisms to update parameters. We identify three

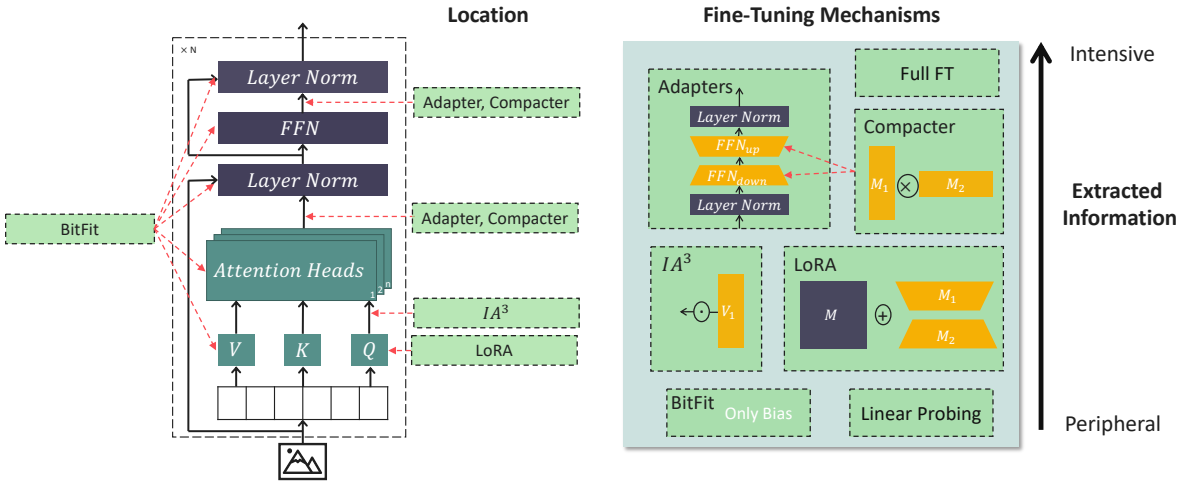

Figure 2. A graphical illustration of how 5 PEFT methods are applied to ViT (left) and a decomposition of PEFT mechanisms (right).

primary mechanisms used—(a) projection with neural layers, which introduces feed-forward layers or layer normalization to down-project and up-project intermediate representations; (b) matrix/vector computation, which applies matrix operations (*e.g.* multiplication) to rescale extracted parameters; and (c) direct update, which directly uses backpropagation to update selected parameters.

These two dimensions remain consistent for each PEFT method across the N blocks of a ViT. We provide a summary mapping table in Section A.6. Our decomposition offers a new perspective on studying the fine-tuning space. This enables us to have a solid foundation to further investigate how and why different fine-tuning strategies may have different degrees of robustness.

3.2.2. Sensitivity Analysis

Our framework, as shown in Figure 1, systematically analyzes how robustness evolves throughout fine-tuning. As opposed to focusing on the final model state, it captures the dynamics of the trade-off between accuracy and robustness as the model is adapted. This is particularly important for fine-tuning. Here, the model transitions from a general to a specialized state with different numbers of robust and non-robust features learned. Encountering new data phenomena (*i.e.*, step-level) and iterative updates (*i.e.*, epoch-level) lead to changing degrees of the robustness-accuracy trade-off.

Our approach builds on insights from overfitting studies. There, the model's test accuracy declines after prolonged training as it memorizes dataset-specific noises rather than generalizable patterns [38]. Similarly, fine-tuned models may learn *non-robust* features [20, 45] from downstream datasets to improve accuracy, but it degrades robustness. Here, our pipeline has two stages—(a) fine-tuning integration and (b) continuous evaluation. We first integrate PEFT modules by modifying the pre-trained model structure. Then, as the model is fine-tuned on downstream data, we use an adaptive tracking schedule to continuously evaluate model robustness and accuracy.

Intuitively, robust and non-robust features learned from upstream and downstream data at different points influence model robustness and accuracy on target tasks. To capture the full variances during training, we monitor them at key update steps during fine-tuning. A major challenge here is to determine the optimal tracking frequency. While monitoring per epoch is standard, it is too coarse for classification tasks where fine-tuned models converge within a few epochs [18]. Early-stage changes of robustness are missed with sparse tracking. Instead, we track robustness at the granularity of backpropagation steps. This captures how the robustness-accuracy trade-off evolves in two ways: (a) when new downstream-specific data is introduced, revealing immediate shifts in robustness as the model encounters new data phenomena, and (b) as the model iteratively updates after seeing the entire dataset, showing longer-term trends in robustness and accuracy trade-offs.

Furthermore, to balance efficiency while tracking at this granularity, we strategically sample model states at selected backpropagation intervals rather than at every step. Specifically, we increase tracking frequency (*i.e.*, every 50-200 steps) during early training, when models rapidly adapt to new data, and decrease it (*i.e.*, every 6,000 steps) in the later stage when performance stabilizes. This adaptive tracking approach ensures that we capture critical transitions in robustness while minimizing unnecessary overhead.

4. Evaluation

Building upon our framework of fine-tuning integration and continual robustness evaluation, we empirically investigate how robustness changes during fine-tuning. Specifically, we

seek to understand whether the shift from training a model from scratch to various fine-tuning strategies changes the robustness-accuracy trade-off. To this end, we address the following three key research questions:

RQ1: Does the established trade-off between accuracy and adversarial robustness persist in fine-tuning?

RQ2: How do different fine-tuning strategies and downstream task complexity affect the optimal trade-offs?

RQ3: Are the findings consistent with out-of-distribution (OOD) robustness?

4.1. Experimental Design

To facilitate our experiments, we use ViT-Base model, pre-trained on ImageNet-21k [43] (14 million images, 21,843 classes) at resolution 224x224 from HuggingFace [19], and AdapterHub [1] v1.0.0 to integrate PEFT modules. Experiments are performed across 12 A100 GPUs with 40 GB of VRAM and CUDA version 11.7 or greater. Fine-tuning models with tracking adversarial and OOD robustness takes approximately 2,100 and 3,780 GPU hours, respectively. Differences among PEFT strategies are negligible—they all update 0.07%-3.97% of the pre-trained model. Our source code is publicly available.[1]

Since fine-tuning strategies are often used in data-limited regimes, we evaluate them on six representative datasets (10k-60k samples) with varying complexity (10 vs. 256 classes, fine- vs. course-grained classes): CIFAR10 [42], CIFAR100 [25], CalTech256 [13], CUB200 [47], StanfordDogs [23], and DomainNet [41]. Dataset details are in Section A.1. To study scalability with data quantity, we also include Places365 [52] ($\sim$1.8M samples). However, as models converge to only $\sim$50% clean accuracy [52] and $\sim$5% adversarial robustness, their weak overall performance makes it difficult to draw meaningful conclusions about robustness trends as discussed in Section A.5. Additionally, fine-tuning configurations follow standard CV practice [15, 18] (Table 1), with grid search used to select training hyperparameters (Section A.4). We also conduct an ablation study on learning rate and the number and location of trainable parameters, which can be found in Section A.3.

For adversarial robustness evaluations, we use the state-of-the-art attack algorithm PGD [36] from TorchAttack [24] v3.5.1. Following standard practices [31], we set the attack budget to $\epsilon = 1/255$, step size $\alpha = 0.25/255$, and the number of steps to 15. Adversarial examples are generated from the test sets of each downstream dataset. Given the computational cost of attacks, we follow a structured evaluation tracking schedule: (1) in early fine-tuning (0-700 steps), we evaluate robustness and accuracy every 50 steps to capture robustness changes; (2) between $700 - 3,000$ steps, evaluations occur every $1,000$ steps; and (3) beyond $3,000$ steps, we evaluate every $6,000$ steps.

[1] https://github.com/kyangl/robustness-finetuning

PEFTs	Configs	Values
Adapter & Compacter	reduction factor	8
	non linearity	gelu
	locations	multi-heads attn, W_O
$(IA)^3$	locations	W_K, W_V, FFN
LoRA	locations	W_K, W_V, W_Q, W_O

Table 1. Standard configurations of PEFTs.

For OOD robustness, we evaluate the model's ability to generalize across distribution shifts using DomainNet [41]. Here, the model is fine-tuned on a single domain and tested on other unseen domains. Due to the large number of data with higher computational overhead, OOD evaluations are conducted at a coarser granularity: (1) every 200 steps for the first $1,000$ fine-tuning steps; (2) every $2,000$ steps from $1,000$ to $3,000$ steps; (3) every $4,000$ steps from $3,000$ to $10,000$ steps; (4) every $6,000$ steps from $10,000$ to $30,000$ steps; and (5) every $20,000$ beyond $30,000$ steps.

4.2. Accuracy and Robustness Trade-off

Prior studies attribute the trade-off between adversarial robustness and accuracy to models that rely on a large number of non-robust yet highly predictive features [20, 45]. This claim is based on the assumption that the training data and the data used to generate adversarial examples share the same distribution. However, this assumption breaks in fine-tuning. Here, distinct downstream datasets with different inter-class/domain separation, image resolution, and similarities with upstream data are involved during training. This shift raises a crucial question: does the trade-off phenomenon still persist in fine-tuning and how?

To explore this, we measure the variances of robustness and accuracy during fine-tuning for 7 fine-tuning methods and 5 datasets. Figure 3 shows results on `Caltech256`, where models trained with BitFit, LoRA, Adapter, and full fine-tuning all exhibit rapid improvements in standard accuracy, reaching $\approx$ 90% within $1,000$ steps. Meanwhile, adversarial robustness follows a different trajectory: it initially increases, then reaches $\approx$ 25% at around step 400, and finally steadily declines to $\approx$ 10% at convergence. Specifically, PEFT methods that fine-tune parameters in or around attention layers (*e.g.*, LoRA, Adapter) demonstrate a slightly more gradual decline in robustness, suggesting that they preserve robustness better than full fine-tuning or bias-only tuning. We further investigate differences among fine-tuning strategies in the next section.

The distinct trends of accuracy and robustness here highlight that models learn features for different purposes at

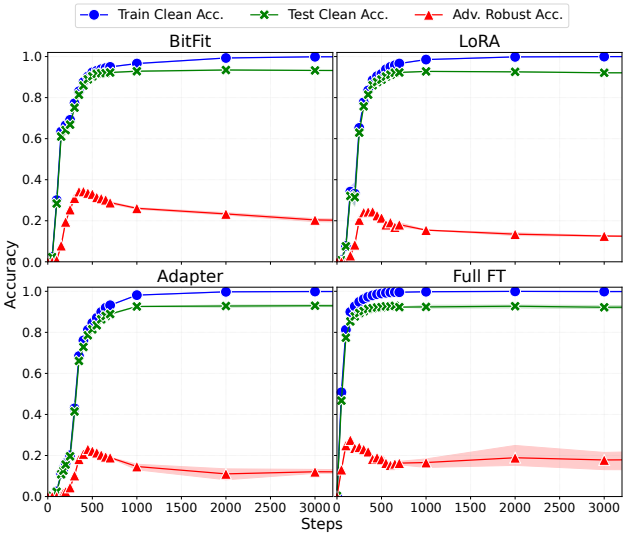

Figure 3. Continuous evaluation of training accuracy (blue), test accuracy (green), and adversarial robustness (red) across back-propagation steps (truncated at 3000 steps) on Caltech256.

different stages. Early fine-tuning steps improve both robustness and accuracy by leveraging pre-trained representations. We attribute this to the model's ability to effectively adapt the *randomly initialized* trainable parameters to downstream tasks. Then, as the model is increasingly fitted to the downstream data, it begins to exploit predictive but non-robust features to gain accuracy while sacrificing robustness. The findings strongly confirm the persistence of the trade-off in fine-tuning and imply that learning robust features with defense mechanisms may decrease accuracy.

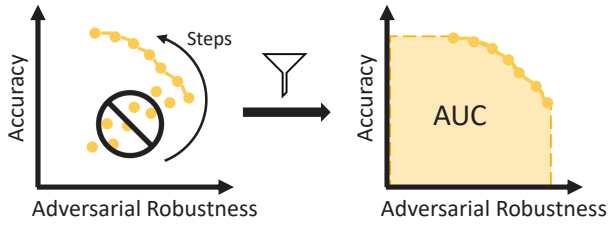

Figure 4. Pareto frontiers are extracted by filtering out the suboptimal points in the trade-off space. The AUC is then computed by extending the two endpoints and integrating the enclosed area.

4.3. Pareto Frontiers in the Trade-off Space

After analyzing the *consistent* adversarial robustness-accuracy trade-off, we further ask how *sensitive* the trade-off is to downstream phenomena and fine-tuning methods.

Across downstream distributions. First, we extract Pareto frontiers by identifying points in the trade-off space that no other point has higher accuracy and robustness simultaneously. As shown in Figure 4, these frontiers represent the optimal trade-offs achieved during fine-tuning. Figure 5 illustrates the results of the fine-tuning methods on the five datasets. It shows significant differences across downstream distributions. CIFAR-10 exhibits the flattest Pareto frontiers (*i.e.*, the most gradual trade-offs). Adversarial robustness remains relatively stable as accuracy improves in the last 2% before convergence (leftmost points of each frontier). Similarly, CIFAR-100 shows less gradual trade-offs, characterized by smaller gradients (in absolute values) of the frontiers. In comparison, the Pareto frontiers are steeper, indicating a more evident trade-off, for Caltech256, CUB200, and Stanford Dogs. Here, the robustness peaks (the rightmost points of each curve) and then sharply declines as accuracy approaches the final 10% before convergence.

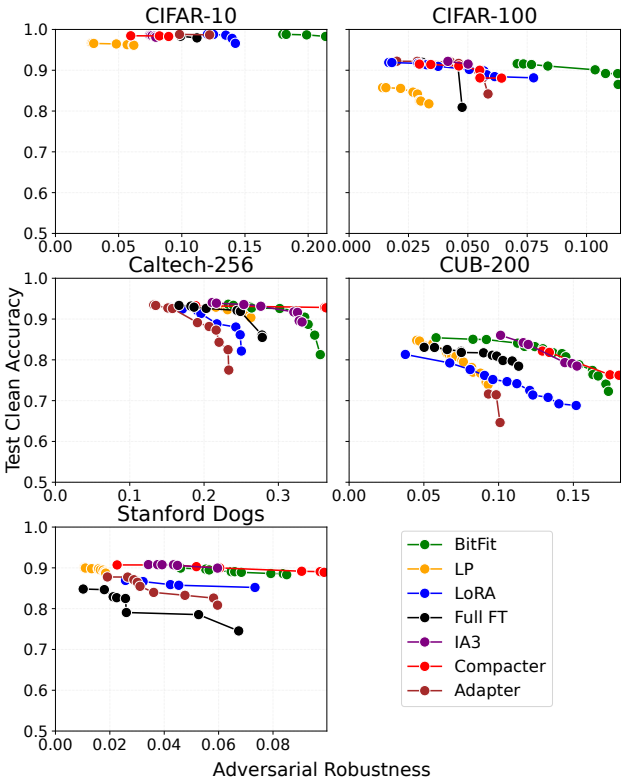

Figure 5. Pareto frontiers of the trade-off between accuracy and robustness on five downstream datasets.

We attribute this variation to task complexity. CIFAR-10 consists of 10 classes that fully overlap with the pre-trained data. This enables a smoother adaptation. In contrast, CUB-200 requires distinguishing 200 bird species, all falling under the broad "Bird" category of the upstream data. The complexity with smaller inter-class separation forces the model to learn finer details, increasing its dependence on fragile, non-robust features. In summary, greater downstream task complexity with less similarity with the

upstream phenomena leads to steeper Pareto frontiers.

Across fine-tuning strategies. Furthermore, to quantify the quality of the trade-off across fine-tuning strategies, we compute a simple scalar—area under the Pareto frontiers [33]—as our metric. Since Pareto frontiers capture the optimal balances that are achievable by each fine-tuning strategy, we aggregate all "suboptimal" points in that trade-off space by extending the two end points of the Pareto frontier and integrating the enclosed area, as shown in Figure 4. We refer to this metric as AUC in the paper. Here, a larger value indicates a better balance between robustness and accuracy.

BitFit achieves the highest AUC for CIFAR10 (75% above the average) and CIFAR100 (81.5%), while Compacter outperforms others on Caltech256 by 57.5%, Stanford Dogs by 24%, and CUB200 by 34.6%. This suggests that BitFit excels on simpler datasets, where modifying only bias terms efficiently adapts pre-trained knowledge while retaining robustness. In contrast, Compacter is better for complex tasks, where its low-rank reparameterization in intermediate layers balances adaptation and robustness inheritance better. Notably, linear probing (LP) and full fine-tuning (Full FT) underperform across all datasets, with the lowest AUCs for CIFAR100 (47.5% and 20% lower than the average, respectively) and CUB200 (28.2% and 19.2%, respectively). As shown in Figure 5, LP and Full FT result in shorter Pareto frontiers, failing to achieve either high robustness or high accuracy compared to others. Fine-tuning all parameters destabilizes robustness, while freezing the entire network except for the classification layer limits adaptation, both resulting in suboptimal trade-offs.

The observed trends align well with our preliminary results (Section 3.1). Here, the fine-tuning methods, corresponding to k, and downstream data phenomena, corresponding to d, lead to different degrees of model robustness. Furthermore, we attribute the different results of different fine-tuning methods to where (*i.e.*, information location) and how (*i.e.*, underlying mechanisms) the models are fine-tuned, as described in Section 3.2.1. For example, BitFit and LP are applied at the periphery of the model, adjusting minimal information, while LoRA, Adapter, and Compacter update deeper layers (*e.g.*, attention mechanisms), introducing information-intense updates. Additionally, full fine-tuning adapts all parameters, leading to a more suboptimal trade-off. Here, methods that adapt intermediate representations maintain better robustness while improving accuracy, whereas peripheral or excessive updates largely degrade the balance between robustness and accuracy.

4.4. On Out-of-Distribution Robustness

Here, we address our third research question—does the robustness-accuracy trade-off observed in adversarial settings extend to real-world out-of-distribution (OOD) scenarios? Unlike adversarial robustness, where non-robust features exploited by attacks contribute to the trade-off, OOD robustness depends on a model's ability to generalize beyond its training distribution. This fundamental difference suggests that fine-tuning strategies may show different behaviors in OOD settings.

	C10	C100	Cal	CUB	Dogs
BitFit	**0.21**	**0.10**	0.33	0.14	0.08
Adapter	0.12	0.05	0.21	0.07	0.05
LoRA	0.14	0.07	0.23	0.12	0.06
Compacter	0.09	0.06	**0.34**	**0.15**	**0.09**
IA3	0.08	0.05	0.31	0.13	0.05
LP	0.06	0.03	0.24	0.08	0.02
Full FT	0.11	0.04	0.26	0.09	0.05

Table 2. Area under the curve (AUC) of the Pareto frontiers.

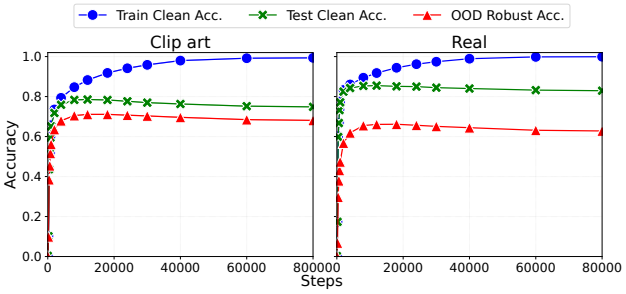

Figure 6. Continuous evaluation on training accuracy (blue), test accuracy (green), and OOD robustness (red) across backpropagation steps on clip art and real images from DomainNet.

Similarly, we track OOD robustness during fine-tuning and compare it to in-domain (test and training) accuracy, as shown in Figure 6. In comparison to adversarial robustness, which often deteriorates after peaking, OOD robustness *plateaus* after the initial improvement at a lower level than standard accuracy. Both in-domain and OOD robustness slightly decline after convergence, likely due to overfitting rather than the accuracy-robustness conflict in adversarial settings. This behavior highlights the different mechanisms behind adversarial and OOD robustness. While adversarial robustness is affected by low-level, non-robust features, OOD robustness depends on transferable features that are less sensitive to fine-tuning-induced degradation.

As shown in Figure 7, we further analyze OOD robustness across the seven fine-tuning strategies and six training domains. Linear probing consistently yields the lowest OOD robustness ($61\% \pm 5\%$), while full fine-tuning achieves the highest ($73\% \pm 2\%$). In addition, models fine-tuned on the "real" domain, which is closest to the pre-training distribution, exhibit lower OOD robustness ($64\% \pm 5\%$) compared to more shifted domains such as "infograph" ($73\% \pm 4\%$) and "quickdraw" ($72\% \pm 3\%$). In-

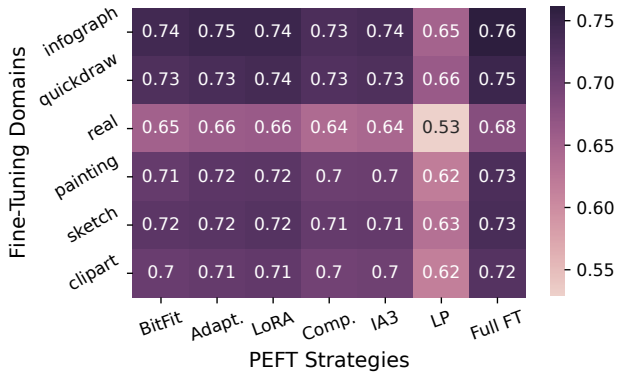

Figure 7. Heatmap representation of peak OOD robustness.

terestingly, OOD robustness stays stable across fine-tuning strategies, excluding linear probing. This suggests that fine-tuning methods support model generalization to other domains similarly. The results reinforce that OOD robustness is primarily dependent on domain shifts and the extent of parameter updates (*e.g.* linear probing vs. full fine-tuning), rather than the specific underlying mechanisms.

5. Related Work

The fundamental trade-off between robustness and accuracy has been extensively studied [20, 45, 51]. Tsipras et al. [45] argue that this trade-off arises from the data distribution, where feature representations learned for high clean accuracy often rely on weakly-related (to ground-truth label) features that degrade under adversarial conditions. Ilyas et al. [20] further highlight that adversarial vulnerability is a consequence of models exploiting highly predictive yet non-robust features. Zhang et al. [51] propose a theoretically justified adversarial training framework to mitigate this trade-off but acknowledge the inherent tension between robustness and accuracy. However, these studies primarily focus on (a) full model training and (b) attacking with data (test set) similar to the training data, whereas our work examines this trade-off within the context of pre-training and fine-tuning, considering various fine-tuning methods and distinct upstream-downstream data phenomena shifts.

Furthermore, with the rise of fine-tuning, robustness considerations of them have become increasingly important. Prior research has explored robustness from a broader perspective, focusing on studying adversarial robustness either during (a) pre-training robustness [4, 6, 22] or (b) fine-tuning [21, 35, 48]. Specifically, these studies primarily investigate full fine-tuning or linear probing, without considering various PEFT strategies. Recent studies have attempted to improve PEFT robustness. Hua et al. [18] propose a robustness-aware initialization strategy. AdapterMixup [39] integrates adversarial training with adapters using mixup. CLAT [12] studies layer-level robustness under adversarial training, while we focus on method-level evaluation across fine-tuning strategies. Additionally, they study the final model state instead of investigating with a finer granularity on how the robustness-accuracy tension changes fundamentally throughout fine-tuning before introducing robustness-improving mechanisms.

6. Discussion & Limitations

Non-robust features & Robustness. As adversarial robustness is closely tied to non-robust features from downstream data, one natural question arises: *are there non-robust features from upstream data?* The "robustness" of features here is relative. Since the goal is to apply models to solve downstream tasks, we only consider model accuracy and robustness on downstream phenomena. Thus, non-robust features specific to upstream data may be robust to adversarial perturbations targeted at downstream data. However, it is convoluted to distinguish *when* the model learns *what* features from different data phenomena when models can also learn them simultaneously. One potential way to further investigate this is to track robustness using adversarial examples generated on both upstream and downstream data.
Limitations & future work. Our study focuses on understanding robustness dynamics of SOTA PEFT strategies before applying defense mechanisms. Due to runtime constraints in continuous evaluation, we use PGD, which has been shown to reliably estimate robustness for clean models [7]. Future work can extend our framework to additional model types (*e.g.*, CNN-ViT hybrid models, adversarially-trained models) and broader threat scenarios (*e.g.*, black-box, adaptive, and data corruption attacks).

7. Conclusion

Our study systematically examines the trade-off between robustness and accuracy in the fine-tuning paradigm, revealing that adversarial robustness initially improves but declines as fine-tuning continues. Across 231 fine-tuned models with 7 state-of-the-art fine-tuning strategies and our independent security and safety evaluation framework, we find that the balance of the trade-off varies with fine-tuning strategies, downstream data complexity, and its similarity to pre-training data. Specifically, methods updating attention-related layers (*e.g.*, LoRA, Compacter) tend to better balance robustness and accuracy, while simpler adaptation techniques (*e.g.*, BitFit) achieve higher robustness peaks but degrade faster. However, this trade-off does not extend to safety (OOD) robustness, which instead depends on the model's ability to generalize across domains. These findings suggest that the design of robustness-aware fine-tuning strategies should consider both adversarial and OOD robustness independently.

Acknowledgements. We would like to thank Kassem Fawaz and Rahul Chatterjee for their helpful comments on earlier iterations of this work. This material is based upon work supported by the National Science Foundation under Grant No. CNS-2343611 and by the U.S. Army Research Office under MURI grant No. W911NF-21-1-0317. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the U.S. Army Research Office. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

References

[1] AdapterHub. AdapterHub Documentation — AdapterHub documentation, 2025. 2, 5

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, 2020. arXiv:2005.14165 [cs]. 1, 2

[3] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks, 2017. arXiv:1608.04644 [cs]. 1, 2

[4] Dian Chen, Hongxin Hu, Qian Wang, Yinli Li, Cong Wang, Chao Shen, and Qi Li. CARTL: Cooperative Adversarially-Robust Transfer Learning, 2021. arXiv:2106.06667 [cs]. 1, 8

[5] Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. Parameter-Efficient Fine-Tuning Design Spaces, 2023. arXiv:2301.01821 [cs]. 1

[6] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning, 2020. arXiv:2003.12862 [cs]. 1, 8

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020. arXiv:2003.01690 [cs]. 1, 8

[8] Linhao Dong, Shuang Xu, and Bo Xu. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888, 2018. ISSN: 2379-190X. 2

[9] 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, 2021. arXiv:2010.11929 [cs]. 2

[10] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J. Clark, and Mehdi Rezagholizadeh. KronA: Parameter Efficient Tuning with Kronecker Adapter, 2022. arXiv:2212.10650 [cs]. 1

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples, 2015. arXiv:1412.6572 [stat]. 1, 2

[12] Bhavna Gopal, Huanrui Yang, Jingyang Zhang, Mark Horton, and Yiran Chen. Criticality Leveraged Adversarial Training (CLAT) for Boosted Performance via Parameter Efficiency, 2024. arXiv:2408.10204 [cs]. 8

[13] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. 5, 11

[14] Demi Guo, Alexander M. Rush, and Yoon Kim. Parameter-Efficient Transfer Learning with Diff Pruning, 2021. arXiv:2012.07463 [cs]. 1

[15] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient Model Adaptation for Vision Transformers, 2023. arXiv:2203.16329 [cs]. 2, 5

[16] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP, 2019. arXiv:1902.00751 [cs]. 1, 2, 3, 12

[17] 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, 2021. arXiv:2106.09685 [cs]. 1, 2, 3, 12

[18] Andong Hua, Jindong Gu, Zhiyu Xue, Nicholas Carlini, Eric Wong, and Yao Qin. Initialization Matters for Adversarial Transfer Learning, 2024. arXiv:2312.05716 [cs]. 1, 4, 5, 8

[19] HuggingFace. google/vit-base-patch16-224-in21k · Hugging Face, 2025. 5

[20] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples Are Not Bugs, They Are Features, 2019. arXiv:1905.02175 [stat]. 1, 4, 5, 8

[21] Ahmadreza Jeddi, Mohammad Javad Shafiee, and Alexander Wong. A Simple Fine-tuning Is All You Need: Towards Robust Deep Learning Via Adversarial Fine-tuning, 2020. arXiv:2012.13628 [cs]. 8

[22] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust Pre-Training by Adversarial Contrastive Learning. In *Advances in Neural Information Processing Systems*, pages 16199–16210. Curran Associates, Inc., 2020. 1, 8

[23] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs. In *Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 5, 11

[24] Hoki Kim. Torchattacks: A PyTorch Repository for Adversarial Attacks, 2021. arXiv:2010.01950 [cs]. 5

[25] Alex Krizhevsky. CIFAR-10 and CIFAR-100 datasets, 2009. 5, 11

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. 2

[27] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution, 2022. arXiv:2202.10054 [cs]. 1, 2

[28] Quoc Viet Le, Tamas Sarlos, and Alexander Johannes Smola. Fastfood: Approximate Kernel Expansions in Loglinear Time, 2014. arXiv:1408.3060 [cs]. 1

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning, 2021. arXiv:2104.08691 [cs].

[30] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. 1

[31] Yanxi Li and Chang Xu. Trade-off between Robustness and Accuracy of Vision Transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7558–7568, Vancouver, BC, Canada, 2023. IEEE. 1, 5

[32] Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning, 2024. arXiv:2303.15647 [cs]. 2, 3

[33] Camille Olivia Little, Michael Weylandt, and Genevera I. Allen. To the Fairness Frontier and Beyond: Identifying, Quantifying, and Optimizing the Fairness-Accuracy Pareto Frontier, 2022. arXiv:2206.00074 [stat]. 7

[34] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning, 2022. arXiv:2205.05638 [cs]. 1, 2, 3

[35] 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)*, pages 16436–16446, Vancouver, BC, Canada, 2023. IEEE. 8

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, 2019. arXiv:1706.06083 [stat]. 1, 2, 5

[37] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient Low-Rank Hypercomplex Adapter Layers, 2021. arXiv:2106.04647 [cs]. 2, 3

[38] Andrew Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *Twenty-first international conference on Machine learning - ICML '04*, page 78, Banff, Alberta, Canada, 2004. ACM Press. 4

[39] Tuc Nguyen and Thai Le. Adapters Mixup: Mixing Parameter-Efficient Adapters to Enhance the Adversarial Robustness of Fine-tuned Pre-trained Text Classifiers, 2024. arXiv:2401.10111 [cs]. 1, 8

[40] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings, 2015. arXiv:1511.07528 [cs]. 1, 2

[41] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment Matching for Multi-Source Domain Adaptation, 2019. arXiv:1812.01754 [cs]. 2, 5, 11, 12

[42] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 Classifiers Generalize to CIFAR-10?, 2018. arXiv:1806.00451 [cs]. 5, 11

[43] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses, 2021. arXiv:2104.10972 [cs]. 5

[44] Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale Efficiently: Insights from Pre-training and Fine-tuning Transformers, 2022. arXiv:2109.10686 [cs]. 1, 2

[45] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy, 2019. arXiv:1805.12152 [stat]. 1, 3, 4, 5, 8

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2023. arXiv:1706.03762 [cs]. 2

[47] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset, 2011. 5, 11

[48] Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. AutoLoRa: A Parameter-Free Automated Robust Fine-Tuning Framework, 2023. arXiv:2310.01818 [cs]. 8

[49] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models, 2022. arXiv:2106.10199 [cs]. 1, 2, 3, 12

[50] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark, 2020. arXiv:1910.04867 [cs]. 2

[51] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. ISSN: 2640-3498. 1, 8

[52] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 5, 12