PRE-TRAINING ALSO TRANSFERS NON-ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-training has enabled state-of-the-art results on many tasks. In spite of its recognized contribution to generalization, we observed in this study that pre-training also transfers adversarial non-robustness from pre-trained model into fine-tuned model in the downstream tasks. Using image classification as an example, we first conducted experiments on various datasets and network backbones to uncover the adversarial non-robustness in fine-tuned model. Further analysis was conducted on examining the learned knowledge of fine-tuned model and standard model, and revealed that the reason leading to the non-robustness is the non-robust features transferred from pre-trained model. Finally, we analyzed the preference for feature learning of the pre-trained model, explored the factors influencing robustness, and introduced a simple robust pre-traning solution.

1 INTRODUCTION

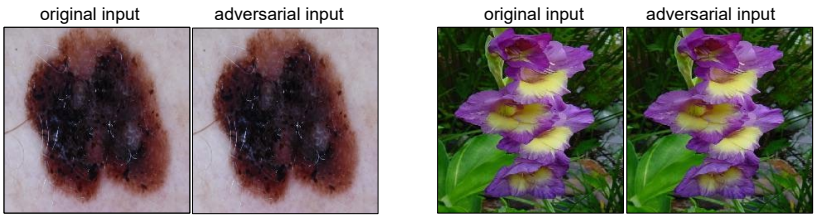
Benefited from both algorithmic development and adequate training data, deep neural networks have achieved state-of-the-art performance across a range of tasks. However, in many real-world applications, it is still expensive or impossible to label sufficient training data. In these cases, a well-established paradigm has been to pre-train a model using large-scale data (e.g., ImageNet) and then fine-tune it on target tasks[1]. Pre-training these days is becoming the default setting not only in researches Xie & Richmond (2018); Lee et al. (2020), but in many industry solutions Chen et al. (2019); Kolesnikov et al. (2020); Brown et al. (2020).

What's wrong with pre-training? With the gradual popularization of pre-training in addressing real-world tasks, it is vital to consider beyond the accuracy on experimental data, especially for tasks with high-reliability requirements. As illustrated in Figure 1, we find *in typical pre-training enabled scenarios, the fine-tuned models tend to have an unsatisfactory performance on robustness*[2]. While confidently recognizing the original input, the fine-tuned models are very sensitive to trivial perturbation and incorrectly classify the adversarial input. The success of pre-training in generalization improvement conceals its defect in decreasing robustness. In this work, we will investigate the robustness of pre-training by systematically demonstrating the performance on robustness, discuss how non-robustness emerges, and analyze what factors influence the robustness.

What accounts for the robustness decrease in the fine-tuned model? We then delve into the cause of the non-robustness by examining the learned knowledge of fine-tuned model. Even though the target tasks of fine-tuned model and standard model are the same, we find that they are quite different in terms of learned knowledge. Furthermore, we analyze what features learned by models lead to the differences and how these features affect robustness. *The non-robust features in the fine-tuned model are demonstrated to be mostly transferred from the pre-trained model and the mediators that derive non-robustness.* Finally, we attribute the preference for utilizing non-robust features to the difference between the source task and the target task. The difference positively correlates to the robustness decrease.

[1]Pre-training typically involves three models as **pre-trained model** trained on large-scale source dataset, **fine-tuned model** initialized with pre-trained model and then fine-tuned on target dataset, and **standard model** directly trained on target dataset (trained from scratch).

[2]Robustness in this paper refers to *adversarial robustness*. We mix these two terms when no ambiguity is caused.



Label	malignant		sword lily	
	Confidence on original input	Confidence on adversarial input	Confidence on original input	Confidence on adversarial input
Standard model	**0.99**	**0.80**	**0.89**	**0.75**
Fine-tuned model	**1.00**	**0.00**	**0.98**	**0.18**

Figure 1: Example of two typical scenarios using pre-training. Regarding the true label, the fine-tuned model obtains higher confidence on original input yet lower confidence on adversarial input than the standard model.

Why does the pre-trained model learn non-robust features? We hypothesize that model can both utilize robust features and non-robust features, and *the pre-trained model tends to rely more on non-robust features when the model capacity is too limited or the source task is too difficult*. Then we study how model capacity and task difficulty, the influencing factors on generalization in prior studies Vapnik & Chervonenkis (2015); Bartlett & Mendelson (2002), influence the learned features of pre-trained model and the robustness of fine-tuned model. It is observed that limited pre-trained model capacity and difficult source task basically lead to non-robust fine-tuned model.

Finally, with the observation that non-robust feature are transferred, a simple robust pre-training solution is introduced by adversarially training the pre-trained model and then regularizing steepness at the fine-tuning stage. Experimental comparison validates its effectiveness of regularizing the difference between target and source tasks.

2 Related Work

It is well-known that transfer learning with CNNs can improve generalization, and many researchers focus on achieving state-of-the-art generalization on downstream tasks Xie & Richmond (2018); Tajbakhsh et al. (2016); Lee et al. (2020). Works investigating the robustness of transfer learning has emerged in the recent years. Adversarial training Madry et al. (2018) provided an alternative way to improve robustness at the fine-tuning stage (denoted as *AT@stage-2*). Salman et al. (2020) introduced adversarial training into the pre-training stage for increasing downstream-task accuracy, and the increase in robustness is actually a byproduct (denoted as *AT@stage-1*). Hendrycks et al. (2019) investigated the gains of pre-training on label corruption, class imbalance, and out-of-distribution detection. They also found employing adversarial training both in pre-training stage and fine-tuning stage can improve adversarial robustness compared with adversarially standard training (denoted as *AT@stage-1&2*). Shafahi et al. (2020) implemented Knowledge Distillation, a defensive tool Papernot et al. (2016), at the fine-tuning stage to improve robustness (denoted as *KD@stage-2*). The authors were motivated by forgetting/un-inheriting knowledge from pre-trained model to the fine-tuned model. However, according to our observation, the non-robust features transferred/inherited from pre-trained model to the fine-tuned model results in non-robustness.

3 Pre-training Is Non-robust

3.1 Notations and Settings

Pre-training. Pre-training is commonly used to initialize the network for target task. We decompose the network for target task into two parts: feature extractor f with parameters θ_f, and classifier g with parameters θ_g. Given an original input x, $f(x;\theta_f)$ denotes the mapping from x to its embedding representation e_x, and $g(e_x;\theta_g)$ denotes the mapping from e_x to its predicted label. Typical pre-training involves with two fine-tuning settings: *partial fine-tuning*, in which only fully connected layer corresponding to the classifier $g(\cdot;\theta_g)$ is updated; and *full fine-tuning*, in which both $f(\cdot;\theta_f)$

Table 1: Comparison of generalization and robustness between standard model, partial fine-tuned model and full fine-tuned model. For each model, we report accuracy of original inputs (AOI), accuracy of adversarial inputs (AAI), and decline ratio (DR) on 7 different target datasets.

Model		Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
Standard	AOI	60.62	81.29	61.29	74.54	74.52	**95.44**	**100.00**
	AAI	40.23	53.45	55.96	53.61	28.24	57.33	99.92
	DR	**33.63**	**34.24**	**8.69**	**28.07**	**62.10**	**39.93**	**0.06**
Partial Fine-tuned	AOI	86.45	91.03	87.98	41.90	58.59	78.48	59.60
	AAI	3.38	10.34	8.23	0.12	0.74	0.00	0.00
	DR	96.09	88.64	90.64	99.76	98.73	100.00	100.00
Full Fine-tuned	AOI	**89.78**	**94.27**	**91.98**	**81.25**	**78.93**	**95.54**	**99.94**
	AAI	15.7	28.33	27.86	18.57	22.30	1.34	2.90
	DR	82.51	69.95	69.71	77.14	71.74	98.59	97.10

and $g(\cdot;\theta_g)$ of pre-trained model are fine-tuned on the target dataset, and $f(\cdot;\theta_f)$ is usually assigned a smaller learning rate.

Adversarial robustness. Adversarial robustness, i.e., robustness, measures model's stability to adversarial example when small perturbation (often imperceptible) is added to the original input. To generate the adversarial example, given an original input x and the corresponding true label y, the goal is to maximize the loss $L(x+\delta, y)$ for input x, under the constraint that the generated adversarial example $x' = x + \delta$ should look visually similar to the original input x (by restricting $\|\delta\|_p \leq \epsilon$, in this work, we use $\|\delta\|_\infty \leq \epsilon$) and $g(f(x')) \neq y$.

Non-robust feature. According to the definition in Ilyas et al. (2019), who simplified classification into a binary case: input space $\mathcal{X} \to \{\pm 1\}$ which predicts a label $\hat{y}$ corresponding to an input x sampled from a dataset $\mathcal{D}$. Each feature $h : \mathcal{X} \to \mathbb{R}$ maps the input space $\mathcal{X}$ to the real number. A feature h is η-*useful* for a dataset $\mathcal{D}$ when there exists a $\eta > 0$ such that,

$$\mathbb{E}_{(x,\hat{y})\in\mathcal{D}}[\hat{y} \cdot h(x)] \geq \eta, \tag{1}$$

which represents h is correlated with the class $\hat{y}$ of an input x. For a given input under adversarial perturbation (for some specified set of valid perturbations Δ) $x + \delta$, a feature h is γ-*robust* when,

$$\mathbb{E}_{(x,\hat{y})\in\mathcal{D}}[\inf_{\delta\in\Delta} \hat{y} \cdot h(x)] \geq \gamma. \tag{2}$$

Otherwise, a feature h is *non-robust* when it is η-useful for some η bounded away from zero, but is not a γ-robust feature for any $\gamma \geq 0$. Therefore, a classifier C is comprised of a set of features $\mathcal{H}$, a weight vector w, and a bias b, such that,

$$C(x) = \text{sgn}(b + \sum_{h\in\mathcal{H}} w_h \cdot h(x)), \tag{3}$$

where w_h reflects the dependence of C on its corresponding feature h.

Datasets. We carry out our study on several widely-used image classification datasets including Pets Parkhi et al. (2012), NICO He et al. (2020), Flowers Nilsback & Zisserman (2008), Cars Krause et al. (2013), Food Bossard et al. (2014), and CIFAR10 Krizhevsky et al. (2009). In addition, we craft a new Alphabet dataset as a comparing example with low semantic complexity and relatively sufficient training data. The Alphabet dataset is constructed by offsetting the 26 English letters and adding random noise, resulting in $1,000$ training images and 200 testing images for each letter class. Example images of these datasets are illustrated in Figure 2.

3.2 Experimental Results on Robustness

To examine whether pre-training transfers non-robustness, we compare the performance of standard model, partial fine-tuned model and full fine-tuned model. Regarding adversarial robustness, we

Figure 2: Example images of Pets, NICO, Flowers, Cars, Food, CIFAR10 and Alphabet, respectively.

introduce decline ratio (DR) as an additional evaluation metric. Given the recognition accuracy of original inputs (AOI) and adversarial inputs (AAI), DR is defined as DR = (AOI-AAI)/AOI. DR serves as a more balanced indicator of model robustness than AAI, especially when two models perform quite differently on original inputs (i.e., AOI). Large DR indicates sharp accuracy decrease in case of input perturbation, and thus inferior robustness.

The pre-trained model is trained on ImageNet dataset with $224 \times 224 \times 3$ input resolution, then all target datasets are set to the same input resolution before feeding into the models. For each type of model, we report maximum accuracy (over different combinations of learning rates for θ_f and θ_g) based on ResNet-18 backbone in Table 1. The robustness is evaluated against PGD-10 attack Kurakin et al. (2017) under $\epsilon = 0.5$ and set step size $= 1.25 \cdot (\epsilon/step)$. The results based on ResNet-50 backbone and WideResNet-50-2 backbone are shown in Supplement B (Table 6 and Table 7).

Table 1 shows that: (1) For most of the examined datasets, fine-tuned models usually achieve better generalization (AOI), but worse robustness (AAI and DR) than standard model. This demonstrates that pre-training not only improves the ability to recognize original input of target tasks, but also transfers non-robustness and makes the fine-tuned model more sensitive to adversarial perturbation. (2) Within the two pre-training settings, full fine-tuning consistently obtains better robustness as well as generalization than partial fine-tuning setting. This suggests that full fine-tuning is preferable when employing pre-training in practical applications to alleviate the decrease in robustness. In the rest of the paper, we deploy full fine-tuning as the default pre-training setting. (3) For CIFAR10 and Alphabet when the standard models trained on target datasets already achieve good AOI, pre-training contributes to trivial improvement on generalization (even with decreased AOI when partially fine-tuned on CIFAR10) but severe non-robustness to the fine-tuned model. In this view, instead of improving fine-tuned model, pre-training actually plays a role as poisoning model Koh & Liang (2017); Liu et al. (2020) (The model behaves normally when encountering normal inputs, but anomalous patterns are activated for some specific inputs.). This further demonstrates the risk of arbitrarily employing pre-training and the necessity to explore the factors influencing the performance of pre-training in subsequent target tasks.

More robustness criteria. To solidly demonstrates the non-robustness of fine-tuned model, we evaluate AAI under stronger/more diverse attacks and different perturbation levels. Aside from the PGD-10#0.5 (i.e., PGD-10 attack under $\epsilon = 0.5$) used in Table 1, FGSM#0.5 (i.e., FGSM attack under $\epsilon = 0.5$), FGSM#2.0 (i.e., FGSM attack under $\epsilon = 2.0$) and AA#0.5 (i.e., Auto Attack Croce & Hein (2020) under $\epsilon = 0.5$) are employed as more robustness criteria. Table 2 shows the AAI of fine-tuned model is lower than standard model using every criterion. Especially when the AOI of the fine-tuned model is basically higher than that of the standard model, the gap between the DR of the fine-tuned model and the standard model is larger. This further demonstrates the fine-tuned model has lower robustness than standard model.

4 Difference between Fine-tuned Model and Standard Model

4.1 On the Learned Knowledge

Knowledge measurement. To understand the performance difference between the fine-tuned model and standard model, we start from examining their learned knowledge. Motivated by many studies on model knowledge measurement Raghu et al. (2017); Morcos et al. (2018); Wang et al. (2018); Kornblith et al. (2019); Liang et al. (2019), we employ a recognized metric, Canonical Correlation Analysis (CCA) Raghu et al. (2017); Hardoon et al. (2004), to quantify the representation similarity between two networks. It is a statistical technique to determine the representational

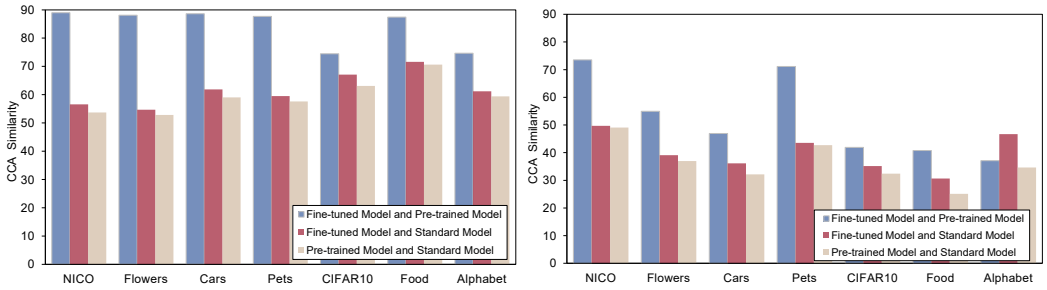
Table 2: The AAI of standard model and fine-tuned model under different adversarial attack and ϵ.

Criterion	Model	Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
PGD-10#2.0	Standard	5.45	3.89	39.31	6.90	0.61	1.14	95.73
	Fine-tuned	0.00	0.08	0.24	0.00	0.02	0.00	0.00
FGSM#0.5	Standard	38.18	51.88	55.20	52.20	28.86	63.10	99.87
	Fine-tuned	26.25	39.02	39.00	25.06	12.19	24.52	75.83
FGSM#2.0	Standard	6.54	9.74	37.94	13.79	4.29	22.02	97.33
	Fine-tuned	1.39	6.13	8.81	1.68	4.97	9.34	53.13
AA#0.5	Standard	5.61	13.06	3.48	3.46	2.45	18.61	99.69
	Fine-tuned	1.04	5.57	11.35	2.46	0.20	0.41	0.00

similarity between two layers L_1, L_2. We briefly explain the flow according to Raghu et al. (2017); Morcos et al. (2018). Let L_1, L_2 be $i \times j$ (i is the number of images, j is the number of neurons) dimensional matrices. To find vectors z, s in $\mathbb{R}^i$, such that the correlation coefficient ρ is maximized:

$$\rho = \frac{\langle z^T L_1, s^T L_2 \rangle}{\|z^T L_1\| \cdot \|s^T L_2\|}. \tag{4}$$

By calculating a series of pairwise orthogonal singular vectors, the mean correlation coefficient is used to represent the similarity of L_1, L_2: $\frac{1}{k}\sum_{a=1}^{k} \rho^{(a)}$, where $k = \min(i, j)$. Specifically, feature extractor $f(\cdot; \theta_f)$ consists of 4 layers, and we compare the similarity between fine-tuned model and standard model using the output of bottom-layer feature (only conv2_x) and the output of total feature extractor (considering features of all 4 layers) respectively. More detailed experimental settings are reported in Supplement C.



(a) Bottom-layer feature (b) All-layer feature

Figure 3: The CCA similarities between different models, which is normalized to [0, 100].

Experimental results. As shown in Figure 3, the fine-tuned model is more similar to the pre-trained model than to the standard model, both on bottom-layer and all-layer features for most of the examined datasets. Since the pre-trained model and standard model are trained on source dataset and target dataset separately, this result seems to tell that more knowledge learned in the fine-tuned model is transferred from the source task data, than from the fine-tuning target task data. By further comparing Figure 3(a) with Figure 3(b), we find that the bottom-layer features of the fine-tuned model and standard model are relatively more similar than all-layer features, suggesting that the bottom-layer features (e.g., edges, simple textures) extract some shared semantics between the source and target tasks. This justifies the role of initialization of pre-training and its contribution to generalization improvement.

4.2 On Non-robust Features

Universal adversarial perturbation. In this subsection, we investigate what features lead to the difference in learned knowledge and how these features affect robustness. Different from standard

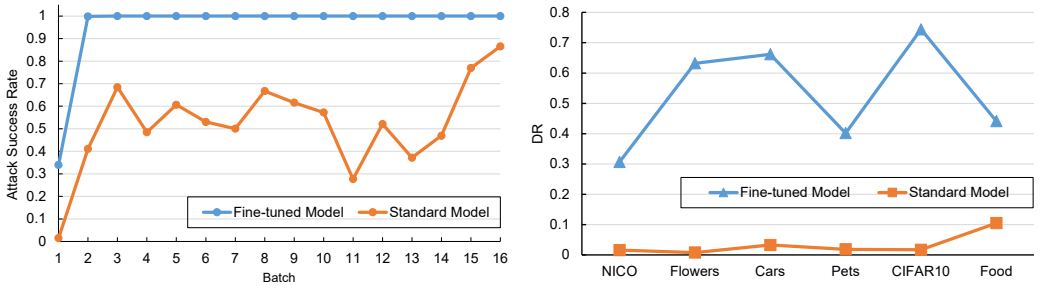
adversarial perturbation which is sample-specific, Universal Adversarial Perturbation (UAP) Moosavi-Dezfooli et al. (2017); Poursaeed et al. (2018) is fixed for a given model misleading most of the input samples Moosavi-Dezfooli et al. (2017). Let μ denotes a distribution of images x in $\mathbb{R}^d$ with corresponding true label y, the focus of UAP is to seek perturbation v that can fool the model by identifying almost all datapoints sampled from μ as the target class $\tilde{y}$ (targeted perturbation):

$$g(f(x+v)) = \tilde{y} \quad \text{for most } x \sim \mu \tag{5}$$

In this work, we mainly focus on targeted UAP and generate it by an encoder-decoder network Poursaeed et al. (2018). Rather than categorizing it as mere adversarial perturbation in the current understanding of a series of works Moosavi-Dezfooli et al. (2017); Poursaeed et al. (2018); Zhang et al. (2020), we find that it contains features that can also work independently. In other words, without adding into any images, the targeted UAP can be identified as the target class with 100.00% confidence, e.g., the first picture in Figure 4 is recognized as letter "A" by the standard model with 100.00% confidence. The two properties demonstrate that *UAP contains patterns that not only effectively cover native semantic features in images, but also can be independently recognized by the model as belonging to the target class*. Therefore, we employ UAP as the probe for features on which the model relies and to understand model behavior.

Figure 4: Visualization of UAP for fine-tuned and standard models on Alphabet: UAPs with different attacking letter classes.

Visualization on features. Figure 4 illustrates the UAP for fine-tuned and standard models on the crafted Alphabet dataset. It is shown that UAP of fine-tuned model expresses no semantics, indicating fine-tuned model prefers to rely on non-robust features. Relying on these noise-alike features, fine-tuned models are vulnerable to adversarial attacks. In contrast, the UAP of standard model contains clear semantics related to the target class. We can see that misleading the standard model is non-trivial and needs to add more human-perceptible information (e.g., edge of "A"). This coincides with the superior robustness of standard model than fine-tuned model.



(a) (b)

Figure 5: UAP attack results: (a) Using UAP of fine-tuned model and standard model to attack themselves at different training batches (on Alphabet). (b) Using UAP of pre-trained model to attack the fine-tuned model and standard model (on other datasets).

The learning process for non-robust features. Next, we employ UAP to examine how the non-robust features are learned. Since the premise behind successful UAP attack is that the models

actually extract the corresponding features, we are motivated to use the above obtained UAP to attack model during its training process to observe when the non-robust features are learned. As shown in Figure 5(a), we record the attack success rate (i.e., the perturbated images are misclassified as the attack letter) at different training batches for the fine-tuned model and standard model respectively. It is easy to perceive that the attack success rate of fine-tuned model remains at a very high level at the very beginning of training. This indicates that these non-robust features are more likely to be transferred from the pre-trained model than learned from the target data. Other observation includes that the UAP of fine-tuned model has a much stronger attack ability than that of standard model, which again demonstrates the inferior robustness of fine-tuned model compared with standard model.

The transferred non-robust feature. We conduct further experiments to confirm whether the *specific* non-robust features (derived from the source task/pre-trained model instead of other ways) are transferred. The idea is to generate UAP on the pre-trained model, and then use this UAP to attack the fine-tuned and standard model on different target tasks. The DR value is reported in Figure 5(b), showing the obvious robustness decrease for the fine-tuned model and trivial influence on the standard model. Note that UAP is model-dependent, the fact that UAP of pre-trained model succeeds in attacking the fine-tuned model validates our assumption that pre-training transfers non-robust features.

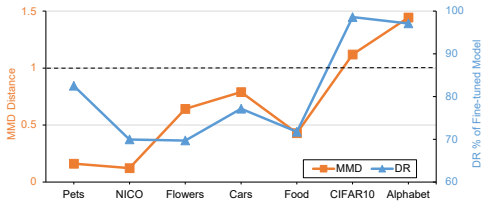


Figure 6: The MMD distance between source dataset and target dataset v.s. DR of fine-tuned model.

The factors affecting the transfer of non-robust features. To delve into the reason why non-robust features are transferred during fine-tuning, we then investigate how the difference between the source task and target task relates to the robustness decrease. We introduce maximum mean discrepancy (MMD) Gretton et al. (2012) to measure the similarity between embedding of source dataset $e_{x_1} \sim p$ and embedding of target dataset $e_{x_2} \sim q$:

$$\mathrm{MMD}(p, q, \mathcal{J}) := \sup_{J \in \mathcal{J}} (\mathbb{E}_{e_{x_1} \sim p}[J(e_{x_1})] - \mathbb{E}_{e_{x_2} \sim q}[J(e_{x_2})]), \tag{6}$$

where $\mathcal{J}$ is a set containing all continuous functions. To solve this problem, Gretton et al. (2012) restricted $\mathcal{J}$ to be a unit ball in the reproducing kernel Hilbert space. Figure 6 compares the DR value of fine-tuned model (from Table 1) with the MMD distance between source dataset and target dataset. We can see that basically DR and MMD distance are in a positive correlation, i.e., the more different target dataset is from source dataset, the more non-robust the fine-tuned model is. Specially, when MMD distance is greater than 1.0, the DR value is almost 100% (the worst case). We draw a rough conclusion that the deviation of the target task from the source task largely affects the robustness of fine-tuned model.

Discussion. It has been recognized that the knowledge and features pre-training transfers are semantic-oriented Yosinski et al. (2014); He et al. (2019). We find from the above analysis that pre-training transfers not only semantic but also non-robust features. Recent studies suggested that non-robust features can help to improve generalization Ilyas et al. (2019) and belong to so-called "shortcut" features Geirhos et al. (2020). We speculate that the transferred non-robust features in pre-training also contributes to the generalization improvement, but imposes robustness problem at the same time. In particular, the experimental results with excessive differences between the source dataset and target dataset (High MMD distance indicates that semantic features are hardly helpful for downstream tasks while the partial fine-tuned model can still achieve passable performance. E.g., in Table 1, the partial fine-tuned model has no robustness (AAI of 0%) but has an AOI of 59.6%.) suggest that *non-robust features seem to be the more transferable than semantic features*. The difference between the target task and source task encourages the non-robust features transfer and increases the risk for robustness decrease.

Table 3: The performance of fine-tuned model with different pre-training architectures (from left to right, the model size increases gradually). Results are averaged over all 7 datasets.

Model		RN-18	RN-50	RN-101	WRN-50-2	WRN-101-2
	Model Size	42.7MB	90.1MB	162.8MB	255.4MB	477.0MB
Full Fine-tuned	AOI	90.24	92.36	92.58	92.91	**93.01**
	AAI	16.71	21.07	23.43	26.73	**29.64**
	DR	81.47	78.11	75.53	72.01	**69.12**

Table 4: The performance of fine-tuned model using different source pre-training datasets.

Dataset	Pets			NICO		
	AOI	AAI	DR	AOI	AAI	DR
ImageNet-10animals	75.91	17.03	77.56	88.90	41.71	53.08
CIFAR10	62.85	26.49	57.85	77.76	47.72	38.63

5 THE NON-ROBUST FEATURE FROM PRE-TRAINED MODEL

The previous sections demonstrate that the non-robustness in pre-training is derived from the non-robust features originating from the pre-trained model. The issue is how pre-trained model gets the non-robust features during the pre-training phase. So this section investigates the feature preference of pre-trained model and the factors influencing the preference. A simple hypothesis: when the model capacity is too limited or the source task is too difficult, the pre-trained model itself tends to rely more on non-robust features and represents more risk to affect the robustness of fine-tuned models.

5.1 ANALYTICAL RESULTS ON MODEL CAPACITY AND TASK DIFFICULTY

The key to understanding the pre-trained model's preference for utilizing features is to observe their weights w_h^r (for robust feature) and w_h^n (for non-robust feature) during the training phase. For utilizing non-robust features h^n, we evaluate the weights w_h^n by perturbing the non-robust features and observing change of accuracy during the training phase, a larger change indicates that more non-robust features are utilized. For utilizing robust features h^r, since it is difficult to perturb them directly, we evaluate the weights w_h^r by penalizing non-robust weights w_h^n (approaching zero, via adversarial training) and observing accuracy during the training phase, a better accuracy indicates that more robust features are utilized. We find that: (1) The limited model lacks the ability to learn sufficient robust features and prefers to utilize more non-robust features; (2) The difficult task makes the model lack the ability to learn sufficient robust features (utilize more non-robust features). The detailed results are reported in Supplement D.

5.2 FACTORS INFLUENCING ROBUSTNESS OF FINE-TUNED MODEL

Model capacity. We then employ model size to examine the influence on fine-tuned model. Table 3 shows the average results for 5 ResNet-based backbones as pre-training architecture (detailed results are reported in Supplement E (Table 8)): ResNet-18 (RN-18), ResNet-50 (RN-50), ResNet-101 (RN-101), WideResNet-50-2 (WRN-50-2), and WideResNet-101-2 (WRN-101-2). It is easy to find that as network size increases, both the generalization and robustness consistently improve (with DR value decreasing from 81.47 to 69.12). This indicates that the high capacity of the pre-trained model alleviates the non-robustness transfer to the fine-tuned models.

Task difficulty. Task difficulty largely depends on the dataset. In this work, we measure task difficulty as the amount of semantics in the dataset necessary to solve the task. Specifically, we select 2 source datasets for comparison: ImageNet-10animals (a subset of ImageNet, with sufficient semantics and containing images of animals) and CIFAR10 (with insufficient semantics and containing images of animals) with the equal number of training images ($50,000$ images of 10 classes). To ensure the scale of source domain to target domain, we select 2 target datasets that also contain images

Table 5: The AAI value based on different pre-training methods. Results are averaged over 7 datasets.

raw	*AT@stage-1*	*AT@stage-2*	*AT@stage-1&2*	*KD@stage-2*	*MD@stage-1&2*
16.71	65.07	70.39	74.67	34.19	76.36

of animals: Pets and NICO. The performance of fine-tuning on different target datasets is reported in Table 4. It is unsurprising that employing ImageNet-10animals as pre-training dataset gives rise to fine-tuned models with higher AOI. However, the fine-tuned models transferred from CIFAR10 achieves lower DR (better robustness), which indicates that the source dataset with more semantics improves generalization yet has more risk to transfer non-robustness. Therefore, the guideline in selecting source dataset for robust fine-tuned models seems not that straightforward. Uncovering the paradox between generalization improvement and robustness decrease for pre-training needs to further study the mechanism of feature learning.

6 Robust Pre-training

The related works have mainly focused on how to improve robustness of pre-training, but hardly any work has paid to how and why pre-training derives non-robustness. Ignoring it and simply using generic adversarial defenses may lead to a gap from the theoretically optimal robustness, while focusing on the difference between fine-tuned model and standard model has the potential to achieve better performance than the above generic adversarial defenses. E.g., feature space steepness is a characterizing factor for robustness, and we observe that using pre-training increases the feature space steepness of the model. With the *Local Lipschitzness* Yang et al. (2020) as the indicator for measuring feature space steepness, the model using pre-training shows significant discrepancy on the same target task. Therefore, we propose a method called *Discrepancy Mitigating* that regularizes the steepness of the feature space at (inspired by a smooth representation-based defense Zhang et al. (2019)) the two stages (denoted as *DM@stage-1&2*), and it outperforms existing methods in transfer learning as shown in Table 5. More details can be found in Supplement F. So the significance of understanding why pre-training transfer non-robustness goes beyond itself, and we hope this study can draw attention to delving into it.

7 Conclusion and Discussion

Conclusion In this work, we demonstrate that pre-training has risk to transfer non-robustness. Using image classification as an example, we first explore the influencing factors of model capacity and task difficulty to provide some practical guidelines for robust pre-training settings. Then we discuss the reason for robustness decrease that the difference between target and source tasks encourages the transfer of non-robust features from pre-trained model to fine-tuned model. Finally, we introduce a simple yet effective robust pre-training solution by regularizing the steepness of pre-trained feature space on target dataset. Experimental results further validate the role of target-source task difference in transferring non-robustness.

Discussion This paper studies pre-training as the example paradigm of transfer learning. Also of vital importance is to examine the reliability of other transfer learning paradigms like knowledge distillation and domain adaption. A particular phenomenon is the non-reliability accumulation in iterative transfer learning. For example, there has been an increasing attempt to automatically label data by a well-trained model Yalniz et al. (2019); Zoph et al. (2020); Xie et al. (2020); Kahn et al. (2020). Since it is difficult to tell whether the data is labeled by human or by model, there exists a risk to iteratively transfer the pseudo label from one to another model. Without human intervention to correct the potentially faulty knowledge, the continuous transfer of knowledge among models likely leads to the so-called "echo chamber" situation in sociology Barberá et al. (2015). As observed in this work, one single round of knowledge transfer can contribute to considerable reliability issues, and iterative transfer may result in catastrophic results. In summary, many works remain to explore the mechanisms behind non-reliability transfer, and we are working towards developing more reliable transfer learning solutions.

8 REPRODUCIBILITY STATEMENT

In Section 3.2, we described the details of the data processing and robustness evaluation. In Supplement A, we reported the details of the experimental settings. In Supplement B, we analyzed the effect of learning rate on robustness. In supplemental materials, we released easy-to-use code.

REFERENCES

Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10):1531–1542, 2015.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Proceedings of the European Conference on Computer Vision*, pp. 446–461, 2014.

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. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer learning for 3d medical image analysis. *arXiv preprint arXiv:1904.00625*, 2019.

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

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, pp. 107383, 2020.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *Proceedings of the International Conference on Machine Learning*, pp. 2712–2721, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2019.

Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 7084–7088, 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the International Conference on Machine Learning*, pp. 1885–1894, 2017.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Proceedings of the European Conference on Computer Vision*, pp. 491–507, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the International Conference on Machine Learning*, pp. 3519–3529, 2019.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *Proceedings of the International Conference on Learning Representations*, 2017.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Ruofan Liang, Tianlin Li, Longfei Li, Jing Wang, and Quanshi Zhang. Knowledge consistency between neural networks and beyond. In *Proceedings of the International Conference on Learning Representations*, 2019.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Proceedings of the European Conference on Computer Vision*, pp. 182–199, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773, 2017.

Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 5732–5741, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 582–597, 2016.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4422–4431, 2018.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 6078–6087, 2017.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. In *Proceedings of the International Conference on Learning Representations*, 2020.

Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pp. 11–30. 2015.

Liwei Wang, Lunjia Hu, Jiayuan Gu, Yue Wu, Zhiqiang Hu, Kun He, and John Hopcroft. Towards understanding learning representations: to what extent do different neural networks learn the same representation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 9607–9616, 2018.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Yiting Xie and David Richmond. Pre-training on grayscale imagenet improves medical image classification. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018.

I Zeki Yalniz, Herve Jegou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Proceedings of the International Conference on Neural Information Processing Systems*, 33, 2020.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 3320–3328, 2014.

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 International Conference on Machine Learning*, pp. 7472–7482, 2019.

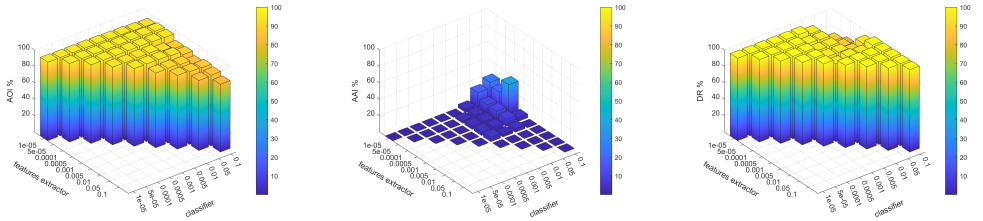
Jiaming Zhang, Jitao Sang, Xian Zhao, Xiaowen Huang, Yanfeng Sun, and Yongli Hu. Adversarial privacy-preserving filter. In *Proceedings of the ACM International Conference on Multimedia*, pp. 1423–1431, 2020.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Proceedings of the International Conference on Neural Information Processing Systems*, 33, 2020.

A Dataset and Experimental Setting

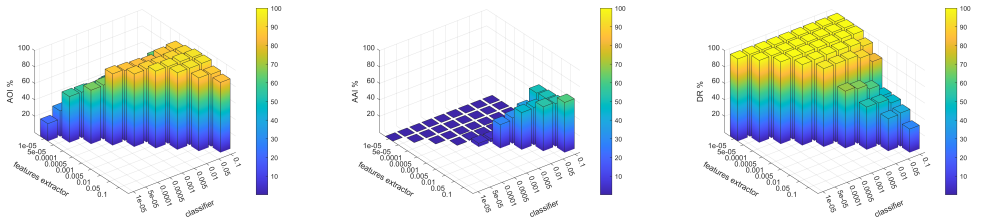
Experimental setting All analyses and experiments are conducted on NVIDIA RTX 2080Ti GPUs. The code for the analyses and experiments uses PyTorch framework. The pre-trained models using ImageNet as training set are provided by PyTorch package. We run 100 epochs for training fine-tuned model and standard model including ResNet-18, ResNet-50, ResNet-101, WideResNet-50-2 and WideResNet-101-2 (batch size is 128, 64, 64, 32 and 32, respectively). The data augmentation pipeline is first resize to $256 \times 256 \times 3$ input resolution input resolution, then randomly cropped to $224 \times 224 \times 3$ resolution.

Dataset We carry out our study on several widely used image classification datasets including Pets, NICO, Flowers, Cars, Food, and CIFAR10. **Pets** has 37 categories, and contains $3,680$ training images and $3,669$ testing images. **NICO** has 10 categories, and contains $10,491$ training images and $2,496$ testing images. **Flowers** has 102 categories, and contains $2,040$ training images and $6,149$ testing images. **Cars** has 196 categories, and contains $8,144$ training images and $8,041$ testing images. **Food** has 101 categories, and contains $75,750$ training images and $25,250$ testing images. **CIFAR10** has 10 categories, and contains $50,000$ training images and $10,000$ testing images.



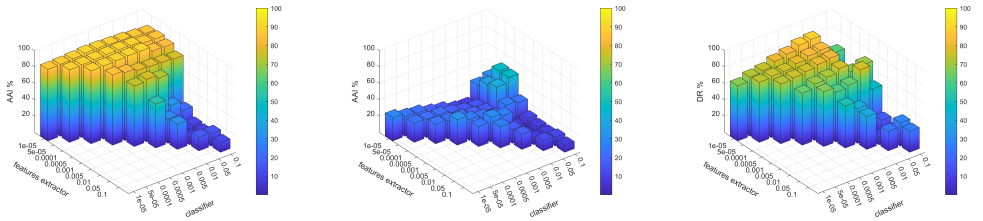
(a) AOI (b) AAI (c) DR

Figure 7: The performance of fine-tuned model on CIFAR10 dataset using ADAM optimizer.



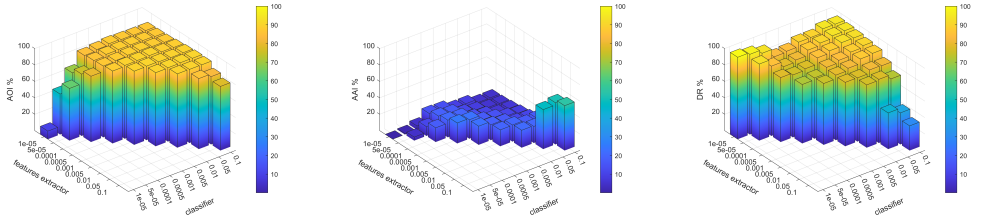
(a) AOI (b) AAI (c) DR

Figure 8: The performance of fine-tuned model on CIFAR10 dataset using SGD optimizer.



(a) AOI (b) AAI (c) DR

Figure 9: The performance of fine-tuned model on Pets dataset using ADAM optimizer.



(a) AOI (b) AAI (c) DR

Figure 10: The performance of fine-tuned model on Pets dataset using SGD optimizer.

Table 6: Comparison of generalization and robustness between standard model, partial fine-tuned model and full fine-tuned model based on ResNet-50 backbone.

Model		Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
Standard	AOI	67.94	84.41	57.87	71.35	82.80	94.38	99.96
	AAI	37.63	44.23	49.45	39.51	22.29	54.20	99.65
	DR	44.60	47.60	14.55	44.63	73.07	42.57	0.30
Partial Fine-tuned	AOI	91.93	95.39	91.73	50.31	69.51	82.62	64.21
	AAI	4.38	12.33	2.94	0.00	0.08	0.00	0.00
	DR	95.22	87.06	96.79	100.00	99.87	100.00	100.00
Full Fine-tuned	AOI	93.45	96.63	95.47	82.02	82.20	96.78	100.00
	AAI	15.72	25.52	24.75	2.40	9.04	0.58	69.53
	DR	83.17	73.59	74.07	97.07	89.00	99.40	30.46

B Extra Experimental Results on Robustness

We provide the performance of standard model, partial fine-tuned model and full fine-tuned model based on other architectures. Table 6 shows the result of ResNet-50, Table 7 shows the result of WideResNet-50-2. The experimental results are consistent with the previous observation that the fine-tuning model is basically less robust than the standard model.

Influence of learning rates. To exclude the interference of learning rates on model robustness, we evaluate different combinations of learning rates. Taking CIFAR10 and Pets as examples, the learning rate is searched from 0.1 until 0.00001 with ADAM and SGD optimizers (each decay is half of the previous one, the learning rate of $f(\cdot;\theta_f)$ is less than or equal to $g(\cdot;\theta_f)$). The Figure 7 8 9 10 show that the fine-tuned models are generally less robust than the standard model, regardless of the learning rates or optimizers used.

C Extra Experimental Results on Knowledge

We break the ResNet-18 into 4 layers, the illustration is shown in Figure 11. In Section 4.1, we report the results of Layer1 (bottom-layer feature) and Layer4 (all-layer feature) of ResNet-18. In this section, we report the results of Layer2 and Layer3 of ResNet-18 as shown in Figure 12. We can find that the results are consistent with our hypotheses: The knowledge learned by the fine-tuned model is similar to that learned by the pre-trained model regardless of the layer.

D Extra Analytical Results on Model Capacity and Task Difficulty

Model capacity. Since the computational complexity of adversarial training on ImageNet dataset is too high, we use CIFAR10 as the source dataset, ResNet-50 (RN-50) and WideResNet-50-2 (WRN-50-2) as variable (ResNet-50 has smaller model size) to observe how model capacity influences

Table 7: Comparison of generalization and robustness between standard model, partial fine-tuned model and full fine-tuned model based on WideResNet-50-2 backbone.

Model		Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
Standard	AOI	69.80	82.69	58.09	75.12	83.54	95.09	99.98
	AAI	31.42	37.25	47.92	28.45	25.07	48.38	99.69
	DR	54.97	54.94	17.49	62.12	69.98	49.12	2.88
Partial Fine-tuned	AOI	91.11	95.23	88.51	41.69	62.68	81.27	59.13
	AAI	3.46	9.33	1.20	0.00	0.12	0.00	0.00
	DR	96.20	90.19	98.64	100.00	99.79	100.00	100.00
Full Fine-tuned	AOI	93.70	97.31	94.73	81.90	83.11	97.35	100.00
	AAI	26.08	35.41	29.43	1.77	11.29	5.89	54.11
	DR	72.16	63.60	68.92	97.82	86.41	93.94	45.88

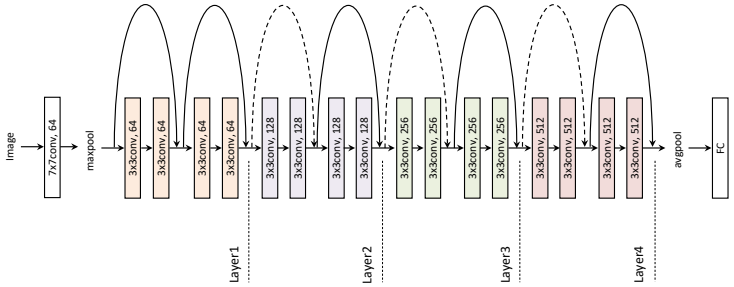


Figure 11: Illustration of the ResNet-18. The output of Layer1 is the corresponding bottom-layer feature, the output of Layer4 is the corresponding all-layer feature.

utilizing features. Figure 13 (a) shows that ResNet-50 requires more training epochs to reach near-zero error and converges more slowly than WideResNet-50-2 on the adversarial training images, demonstrating the limited model lacks the ability to learn sufficient robust features; Figure 13 (b) and (c) show that the ResNet-50's gap (gray area) between adversarial images and original images is narrower than WideResNet-50-2, demonstrating limited model utilizes more non-robust features.

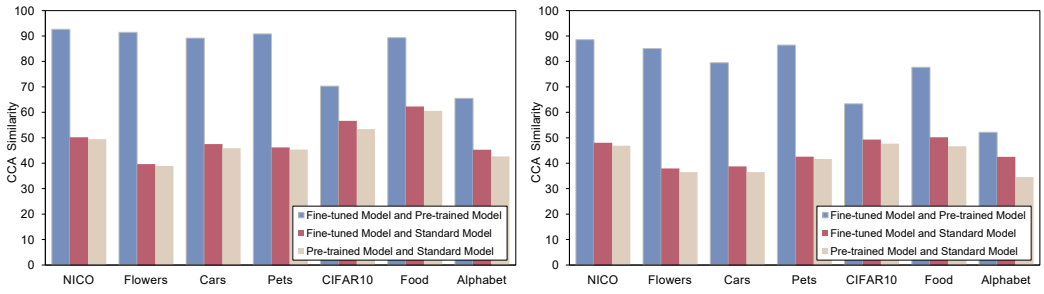
Source task difficulty. We use ResNet-18 as backbone, CIFAR-10 and Alphabet as variable (Alphabet has lower semantic) to observe how source task difficulty influences utilizing features. Figure 13 (d) shows that CIFAR10 converges more harder than Alphabet on the adversarial training images, demonstrating difficult tasks make the model lack the ability to learn sufficient robust features; Figure 13 (e) and (f) show that the Alphabet's gap (gray area) between adversarial images and original images is narrower than CIFAR10, demonstrating the difficult tasks make the model utilize more non-robust features.

E Extra Experimental Results on Model Capacity

Table 8 shows the detailed results for 5 ResNet-based backbones as pre-training architecture: ResNet-18 (RN-18), ResNet-50 (RN-50), ResNet-101 (RN-101), WideResNet-50-2 (WRN-50-2), and WideResNet-101-2 (WRN-101-2). It is easy to find that as network size increases, both the generalization and robustness consistently improve.

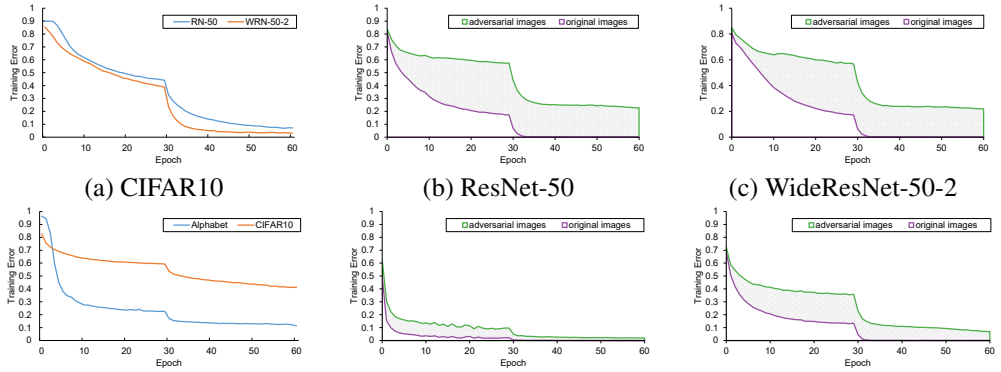
F Details on Robust Pre-training

In this paper, we demonstrate that pre-training not only improves generalization but also transfers non-robustness. As recognized in previous studies Yosinski et al. (2014); He et al. (2019); Hendrycks et al. (2019), the fine-tuned model tends to obtain good initialization and generalization when the



(a) Layer2 feature (b) Layer3 feature

Figure 12: The CCA similarities between different models, which is normalized to [0, 100].



(a) CIFAR10 (b) ResNet-50 (c) WideResNet-50-2

(d) ResNet-18 (e) Alphabet (f) CIFAR10

Figure 13: The analytical results on model capacity and task difficulty. The top three figures are the analytical results about model capacity, and the bottom three figures are the analytical results about task complexity.

target and source tasks are similar. In our view, the improvement probably owes to the similarity that transfers semantic-oriented features to the fine-tuned model. Next, we observe that the difference between target and source tasks encourages the transfer of non-robust features which largely account for the robustness decrease in fine-tuned model. Inspired by this, if we can constrict the difference between target and source tasks, it is expected that non-robust features are discouraged to alleviate robustness decrease and at the same time semantic features are reserved to guarantee generalization. In this section, we first introduce a metric to quantify the difference between target and source tasks, and then design a simple method to employ the metric towards robust pre-training.

F.1 Steepness of Feature Space

Since pre-training essentially serves as a feature extractor for the target task, we propose to measure the difference by examining how the features extracted from pre-trained model fit to the images of target task. Specifically, steepness of feature space is a recognized property closely related to model robustness Yang et al. (2020). Local Lipschitzness (LL) is typically used to calculate steepness as following:

$$\mathrm{LL}(f(X)) = \frac{1}{n}\sum_{i=1}^{n} \max_{x_i' \in \mathbb{B}_\infty(x_i,\epsilon)} \frac{\|f(x_i) - f(x_i')\|_1}{\|x_i - x_i'\|_\infty}, \tag{7}$$

where n denotes the number of images in dataset X, x is original image from dataset X, and x' is the corresponding adversarial image.

A lower value of LL implies smoother feature space which is usually with good robustness. We use ImageNet and Alphabet datasets as examples to respectively train pre-trained models $f^I(\cdot)$ and $f^A(\cdot)$, and then use them to extract features for images from Alphabet dataset X^A. $\mathrm{LLF}(f^I(X^A))$

Table 8: The detailed performance of fine-tuned model with different pre-training architectures (from top to bottom, the model size increases gradually).

Model		Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
RN-18	AOI	89.78	94.27	91.98	81.25	78.93	95.54	99.94
	AAI	15.7	28.33	27.86	18.57	22.30	1.34	2.90
	DR	82.51	69.95	69.71	77.14	71.74	98.59	97.10
RN-50	AOI	93.45	96.63	95.47	82.02	82.20	96.78	100.00
	AAI	15.72	25.52	24.75	2.40	9.04	0.58	69.53
	DR	83.17	73.59	74.07	97.07	89.00	99.40	30.46
WRN-50-2	AOI	93.70	97.31	94.73	81.90	83.11	97.35	100.00
	AAI	26.08	35.41	29.43	1.77	11.29	5.89	54.11
	DR	72.16	63.60	68.92	97.82	86.41	93.94	45.88
RN-101	AOI	93.40	97.23	95.31	83.16	83.76	97.52	100.00
	AAI	18.42	36.17	31.17	9.38	13.69	14.34	63.94
	DR	80.27	62.79	67.29	88.70	83.65	85.29	36.05
WRN-101-2	AOI	93.64	97.67	95.16	83.32	83.90	97.41	100.00
	AAI	29.62	33.45	31.82	4.78	15.09	12.71	80.00
	DR	68.36	65.75	66.55	94.25	82.00	86.95	20.00

Table 9: Comparison of pre-training methods on ResNet18.

Method		Pets	NICO	Flowers	Cars	Food	CIFAR10	Alphabet
Full Fine-tuned	AOI	89.78	94.27	91.98	81.25	78.93	95.54	99.94
	AAI	15.7	28.33	27.86	18.57	22.30	1.34	2.90
	DR	82.51	69.95	69.71	77.14	71.74	98.59	97.10
AT@stage-1	AOI	86.02	92.31	86.23	61.87	70.48	95.78	99.94
	AAI	75.44	83.93	77.98	45.49	44.50	66.10	99.31
	DR	12.29	9.07	9.56	26.47	36.85	30.98	0.63
AT@stage-2	AOI	40.28	91.55	90.55	70.38	70.35	80.68	99.92
	AAI	31.56	80.89	71.93	44.04	52.77	74.53	99.79
	DR	21.65	11.64	20.56	37.42	24.99	7.62	0.13
AT@stage-1&2	AOI	71.52	82.61	81.31	65.39	67.54	90.88	99.88
	AAI	64.49	76.68	77.28	**59.02**	**58.48**	86.93	99.83
	DR	9.83	7.17	4.96	9.73	13.42	4.34	0.05
KD@stage-2	AOI	87.74	91.87	90.71	68.09	74.41	94.56	99.98
	AAI	21.37	31.97	42.06	4.75	5.43	44.28	89.50
	DR	75.64	65.19	53.63	93.02	92.71	53.17	10.48
MD@stage-1&2	AOI	86.48	91.71	87.17	64.83	70.04	95.62	99.96
	AAI	**77.73**	**85.50**	**81.41**	53.46	47.93	**88.63**	**99.90**
	DR	10.11	6.77	6.60	17.53	31.56	7.31	0.05

and $\mathrm{LLF}(f^A(X^A))$ thus represent how ImageNet-trained and Alphabet-trained features fit to the Alphabet images. The result is $\mathrm{LL}(f^I(X^A)) = 367.4$ and $\mathrm{LL}(f^A(X^A)) = 32.9$, indicating that the features pre-trained from ImageNet fail to fit to the Alphabet images.

F.2 Steepness Regularization

We propose to reduce the steepness of pre-trained feature space on the target samples to mitigate the influence of the discrepancy between target and source tasks (called Discrepancy Mitigating). Specifically, in addition to the traditional fine-tuning loss, LLF regularization term is added to derive

the following objective function:

$$\min_{\theta_f,\theta_g} \frac{1}{m} \sum_{i=1}^{m} \mathcal{C}(y, g(f(x_i))) + \lambda \cdot \mathrm{LL}(f(X)), \tag{8}$$

where $\mathcal{C}$ is the cross-entropy classification loss, $\mathrm{LL}(f(X))$ is the steepness regularization term as defined in equation 7, x_i is original image from dataset X, m denotes the number of images in dataset X, and λ is the balancing parameter to control the trade-off between generalization and robustness. The hyperparameter λ in this work is set to be 500. The above optimization problem can be easily solved by PGD-like procedures.

To evaluate the effectiveness of steepness regularization in robust pre-training, we consider several baselines (listed in Section 2) for comparison. Basically speaking, to improve the robustness of fine-tuned model involves with the two stages of fine-tuning and pre-training. Our proposed robust pre-training solution (denoted as *DM@stage-1&2*) combines the two stages: at the pre-training stage we employ adversarial training as in Salman et al. (2020) to obtain a robust pre-trained model, and at the fine-tuning stage we fine-tune on the target dataset according to equation 8 to reduce the feature space steepness caused by the discrepancy between target and source tasks.

The experimental results of ResNet-18 backbone are shown in Table 9. We have the following main findings: (1) Regarding robustness, *MD@stage-1&2* achieves superior AAI and DR in most examined datasets, owing to regularizing the transferred feature space steepness; (2) Regarding generalization, *MD@stage-1&2* guarantees performance compared to the original fine-tuned model, and achieves comparable if not better performance than the baseline methods. This demonstrates the feasibility of regularizing the difference between target and source tasks in addressing the paradox between pre-training robustness and generalization.