TWINS: A Fine-Tuning Framework for Improved Transferability of Adversarial Robustness and Generalization

Ziquan Liu1, Yi Xu2*, Xiangyang Ji3 and Antoni B. Chan1*
1Department of Computer Science, City University of Hong Kong
2School of Artificial Intelligence, Dalian University of Technology
3Department of Automation, Tsinghua University

ziquanliu.cs@gmail.com, yxu@dlut.edu.cn, xyji@tsinghua.edu.cn, abchan@cityu.edu.hk

Abstract

Recent years have seen the ever-increasing importance of pre-trained models and their downstream training in deep learning research and applications. At the same time, the defense for adversarial examples has been mainly investigated in the context of training from random initialization on simple classification tasks. To better exploit the potential of pre-trained models in adversarial robustness, this paper focuses on the fine-tuning of an adversarially pre-trained model in various classification tasks. Existing research has shown that since the robust pre-trained model has already learned a robust feature extractor, the crucial question is how to maintain the robustness in the pre-trained model when learning the downstream task. We study the model-based and data-based approaches for this goal and find that the two common approaches cannot achieve the objective of improving both generalization and adversarial robustness. Thus, we propose a novel statistics-based approach, TWINS fine-tuning framework, which consists of two neural networks where one of them keeps the population means and variances of pre-training data in the batch normalization layers. Besides the robust information transfer, TWINS increases the effective learning rate without hurting the training stability since the relationship between a weight norm and its gradient norm in standard batch normalization layer is broken, resulting in a faster escape from the sub-optimal initialization and alleviating the robust overfitting. Finally, TWINS is shown to be effective on a wide range of image classification datasets in terms of both generalization and robustness.

1. Introduction

The adversarial vulnerability of deep neural networks (DNNs) [59] is one of the major obstacles for their wide

*Corresponding author
UCSD Birds-200-2011 or CUB200 [60], object detection [42] and semantic segmentation [49].

On the other hand, pre-trained models have been considered as the foundation models in deep learning as a result of their strong performance and wide employment in computer vision [17,24,25,45], as well as natural language processing [6,13,52]. Thus, how to better use the pre-trained model in downstream has emerged as a major research topic in many vision and language tasks, such as image classification under distribution shifts [47,63], object detection [36] and semantic segmentation [29,35]. There are a few papers that investigate the pre-trained model’s robustness in target tasks [7,8,15,31,32,57,62]. [7,57] mainly considers the transfer between small-scale datasets (e.g., CIFAR100 to CIFAR10), while [8,32] use adversarial robust pre-training and fine-tuning on the same dataset, without considering a large-scale and general pre-trained model. Finally, [15,62] investigate different kinds of robustness to corruption or out-of-distribution samples, and are not devoted to adversarial robustness.

In this paper, we consider how to transfer the adversarial robustness of a large-scale robust pre-trained model (e.g., a ResNet50 pre-trained on ImageNet [12] with adversarial training) on various downstream classification tasks when fine-tuning with adversarial training. This problem setting is becoming more important as the standard pre-trained models do not learn robust representations from the pre-training data and are substantially weaker than the robust pre-trained counterparts in some challenging downstream tasks, e.g., fine-grained classification as shown in our experiment. Meanwhile, more large-scale robust pre-trained models are released (e.g., ResNet [55] and ViT [4]), which makes the robust pre-trained models more accessible. However, naively applying adversarial training to fine-tune from the robustly pre-trained model will lead to suboptimal robustness, since the robust representations learned by the robust pre-trained model are not fully utilized. For example, [57] suggests that the robustness from a pre-trained model needs to be explicitly maintained for its better transfer to the downstream.

Following the idea that the key to improving the transferability of robustness is to maintain the robustness of the pre-training stage during fine-tuning [57], we first evaluate the data-based and model-based approach on two representative datasets, CIFAR10 and Caltech-256. The data-based approach uses pre-trained data in the fine-tuning and keeps their performance under adversarial attack, while the model-based approach regularizes the distance of features of the fine-tuned and pre-trained model. Our experiment shows that both methods fail to improve the robustness and generalization (Fig. 1), since the two methods are too aggressive in retaining the robustness and hurt the learning in downstream. Thus, we propose a subtle approach that keeps the batch-norm (BN) statistics of pre-training for preserving the robustness, which we call Two-Wing NormaliSation (TWINS) fine-tuning. TWINS has two neural networks with fixed and adaptive BN layers respectively, where the fixed BN layers use the population means and STDs of pre-training for normalization, while the adaptive BN layers use the standard BN normalization. Our experiment first demonstrates the importance of pre-trained BN statistics in the robust fine-tuning and then finds the benefit of TWINS in adversarial training dynamics. As the relationship between weight norm and its gradient norm no longer holds in TWINS, it is able to increase the gradient magnitude without increasing the gradient variance. At the initial training stage, TWINS has a faster escaping speed from the sub-optimal initialization than vanilla adversarial training [41]. At the final training stage, the gradient of TWINS is more stable than adversarial training, which alleviates the robust
overfitting effect [56]. In summary, the contributions of our paper are as follows:

1. We focus on the fine-tuning of large-scale robust pre-trained models as a result of their potential importance in various downstream tasks. We evaluate current approaches to retain the pre-training robustness in fine-tuning, and show that they cannot substantially improve the robustness.

2. We propose TWINS, a statistics-based approach for better transferability of robustness and generalization from the pre-training domain to the target domain. TWINS has two benefits: a) it keeps the robust statistics for downstream tasks, thus helps the transfer the robustness to downstream tasks and b) it enlarges the gradient magnitude without increasing gradient variance, thus helps the model escape from the initialization faster and mitigates robust overfitting. The mechanisms of these two benefits are validated by our empirical study.

3. The effectiveness of TWINS is corroborated on five downstream datasets by comparing with two popular adversarial training baselines, adversarial training (AT) [48] and TRADES [64]. On average, TWINS improves the clean and robust accuracy by 2.18% and 1.21% compared with AT, and by 1.46% and 0.69% compared with TRADES. The experiment shows the strong potential of robust pre-trained models in boosting downstream’s robustness and generalization when using more effective fine-tuning methods.

2. Related Work

Adversarial defense. There are several major approaches to improving the adversarial robustness of DNNs. The training of DNNs can be regularized to induce biases that are beneficial to adversarial robustness, such as locally linear regularization [51], margin maximization [14, 43] and Jacobian regularization [30]. The most commonly used adversarial defense is adversarial training (AT) [48], which directly trains the DNN on adversarial examples generated from PGD attack. Later, TRADES [64] is proposed to add a KL regularization to AT and achieves stronger adversarial robustness. Our paper proposes TWINS to improve adversarial training in the fine-tuning stage when the initial model is adversarially pre-trained. We compare TWINS-AT and TWINS-TRADES with vanilla AT and TRADES in our experiment and show the strong effectiveness of TWINS in the robust fine-tuning setting.

Fine-tuning for downstream robustness. Several aspects of robustness in pre-training and fine-tuning have been studied in existing works. Adversarial contrastive learning [8, 32] is proposed to pre-train on a dataset with contrastive learning and then fine-tune on the same dataset, without considering the transferability of robustness from a large-scale pre-trained model to a different downstream task. In contrast, our paper investigates a more general problem, where task-specific pre-training is not needed for a new task as we use one robust large-scale pre-trained model trained on ImageNet. [55] considers the robust pre-training on the large-scale ImageNet and its transfer to downstream tasks, but focuses on the performance on clean instead of adversarial images. The Learning-without-Forgetting (LwF) [39] approach for retaining robustness is shown to be effective in the small-scale transfer experiment [57], but is not effective in our experiment setting of transfer of large-scale models. [31] proposes a learning rate schedule to improve the adversarial robustness of fine-tuned models, and [16] proposes robust informative fine-tuning for pre-trained language models to robustly keep pre-training information in downstream. The difference between [16, 31] and our work is that they assume a standard pre-trained model instead of the adversarial pre-trained model. [15, 62] investigate the performance of pre-trained models in downstream tasks, but the focus is the robustness to out-of-distribution samples instead of adversarial perturbations.

Batch normalization. There are existing papers proposing the two-branch BN structure for different purposes with different technical details. [58] proposes dual normalization for a better trade-off between accuracy and robustness, where the normalization is a weighted sum of normalized clean and adversarial input. [61] proposes a similar two-branch BN structure, where one branch is for adversarial examples and the other is for clean examples. The major difference between our work and [58, 61] is that both BN branches in TWINS are for adversarial examples and one branch (Frozen Net) has fixed BN statistics from pre-training so as to better maintain the pre-trained robustness, whereas [58, 61] uses clean examples in BN and aims to improve the accuracy for clean images.

3. The Model-based and Data-based Approach to Retaining Adversarial Robustness

This section introduces the two common approaches for keeping adversarial robustness of pre-training in downstream, model-based and data-based approaches. Denote the feature vector output of a neural network as $g_\theta(x)$, the training sample in the downstream task as $\{(x_i, y_i)\}_{i=1}^{N} \sim P$, and the loss function as $L(w^T g_\theta(x) + b, y)$, where $(w, b)$ are the parameters of last classification layer. We assume that the pre-trained model is trained on adversarial examples generated from the PGD attack [48], where the $l_\infty$ norm of the adversarial attack is bounded by $\epsilon$, and during fine-tuning we use adversarial training with the same PGD attack to obtain adversarial robustness in downstream tasks. In short, we consider the robust pre-training and robust fine-tuning setting in this paper, if not specified otherwise.
Model-based approaches. We first introduce the model-based approach, which keeps the pre-trained model $\theta_{d}$ during fine-tuning so as to maintain its robustness. The objective function is

$$\sum_{(x_n, y_n) \sim P} L(w^T g_{\theta}(x_n) + b, y_n) + \lambda_{UOT} \sum_{(x_m, y_m) \sim Q} L(w_q^T g_{\theta}(x_m) + b_q, y_m),$$

(1)

where the adversarial example generated from $x_n$ is denoted as $\tilde{x}_n$. The regularization term of the loss aims to minimize the distance between the features from the pre-trained and the fine-tuned models, which is expected to maintain the robustness of the pre-trained model. This approach is originally proposed in [39] to prevent the catastrophic forgetting in continual learning, and is used in [57] to preserve adversarial robustness in transfer learning. Note that [57] uses the LwF method in standard fine-tuning instead of robust fine-tuning as in our paper.

Data-based approaches. The objective function of the data-based approach is

$$\sum_{(x_n, y_n) \sim P} L(w^T g_{\theta}(x_n) + b, y_n) + \lambda_{UOT} \sum_{(x_m, y_m) \sim Q} L(w_q^T g_{\theta}(x_m) + b_q, y_m),$$

(2)

where $P$ and $Q$ are data distribution of the target and the pre-training tasks, $w_q, b_q$ are the classification layer for pre-trained data from $Q$. This method regularizes the current fine-tuned model feature extractor so that its prediction is still robust on the pre-training data. The joint training method is proposed in [46] to improve the performance of fine-tuning in downstream tasks where the training data is not sufficient.

Next, we test the performance of these two approaches on two standard image classification datasets, CIFAR10 [34] and Caltech-256 [22], with results shown in Figure 1. We use a grid search for learning rate and $\lambda_{UOT}$ and report the result of the model with the best robust accuracy. See Section 5 and the supplemental material for the experiment setting. On CIFAR10, both approaches fail to improve either clean or robust accuracy; on Caltech-256, the two approaches improve the clean accuracy by a small margin but deteriorate the robustness. One reason why the model- and data-based approaches fail is that the regularization term might be too strong, thus hurting the learning in the downstream.

4. TWINS Fine-Tuning

The previous section demonstrates that both data- and model-based approaches cannot substantially improve the adversarial robustness in the downstream task. Thus we propose the TWINS for the better fine-tuning of robust pre-trained models for downstream adversarial robustness.

![Figure 3. The mean and normalized STD of gradient norms in AT and TWINS-AT on CIFAR10, CIFAR100 and Caltech-256. The averaged $\mu(||\nabla w||)$ and $\sigma(||\nabla w||)/\mu(||\nabla w||)$ over epochs are shown in each plot. The gradient magnitudes of TWINS-AT are substantially larger than those of AT, while the Normalized STDs of gradient norm in TWINS-AT are not obviously increased (CIFAR10) or even decreased (CIFAR100 and Caltech-256) compared with AT. This property leads to the faster escaping speed of TWINS-AT from the initial sub-optimum compared with AT (Fig. 4), and reduced robust overfitting (Tab. 1).](image-url)
networks and their BN statistics are initialized by the robust pre-trained models. During training, the Frozen Net uses the population means and STDs of pre-training data computed in the pre-training stage in the normalization operation, while the Adaptive Net uses the current batch’s mean and STD in the normalization and updates its running mean and STD with the target training data. Fig. 2 shows the general pipeline of TWINS training and the network structure.

The training objective of TWINS with adversarial training (TWINS-AT) in one mini-batch is:

$$\sum_{j=1}^{B/2} \mathcal{L}(w^T g_{\theta_j}(\tilde{x}^{(a)}_j) + b, y_j) + \lambda_{\text{twins}} \sum_{i=B/2+1}^{B} \mathcal{L}(w^T g_{\theta_i}(\tilde{x}^{(a)}_i) + b, y_i),$$

where $\theta_f$ and $\theta_a$ denote the Frozen and Adaptive Nets respectively, and $\tilde{x}^{(a)}$ is the adversarial image for $x$ when attacking the Adaptive Net $\theta_a$. We split the batch into two different sub-batches to avoid doubled batch sizes in TWINS training. Since the Frozen and Adaptive Nets share weight parameters, the number of parameters in TWINS is only increased by a very small amount (i.e., BN parameters). Thus, TWINS-AT only has a negligible cost in terms of memory and training time compared with vanilla AT. The TWINS structure can also be used in the TRADES in a similar way. Similar to [38], we can use the target training set to update the BN statistics so that they are more relevant to the downstream task. We call this procedure warmup in TWINS fine-tuning. The pseudo codes of TWINS-AT and TWINS-TRADES are given in Alg. 1 of the supplemental.

### 4.2. The mechanism of TWINS

The first benefit of TWINS is mentioned in the motivation of TWINS, i.e., the BN statistics have robustness information in the pre-training domain that can be leveraged by robust fine-tuning to improve the downstream robustness. This argument is validated by our ablation study in Section 5, where we initialize the means and STDs with 1.0 and 0.0 for Frozen Net instead of the pre-trained means and STDs and check the accuracy and robustness. Figure 5 shows that the TWINS with (1,0) initialization cannot match the performance with TWINS with pre-trained statistics, indicating that the robustness information in pre-training is essential to the effectiveness of TWINS.

It is intriguing that even with the (1,0) initialization, TWINS still outperforms the AT baseline in terms of robustness or accuracy on some datasets, e.g., Stanford Dogs and CIFAR100. This suggests that besides retaining the pre-training information, TWINS provides some other benefits during robust fine-tuning. By analyzing the gradient of TWINS, we find that TWINS implicitly increases the effective learning rate without increasing the oscillation and empirically validate this finding. We detail this analysis next.

**Effective learning rate** We first write the gradient of a weight vector for TWINS training. Consider the $l$-th layer's weight $w_j^{(l)}$ and its output after BN layer

$$\tilde{h}_{ij}^{(l)} = \frac{\tilde{f}_{ij}^{(l)} - 2 \sum_{k=1}^{B/2} \tilde{h}_{kj}^{(l)}}{\sqrt{2/B} \sum_{k=1}^{B/2} (\tilde{h}_{kj}^{(l)} - \mu^{(l-1)})^2}$$

$$= \frac{w_j^{(l)} (h_i^{(l-1)} - \mu^{(l-1)})}{\sqrt{2/B} \sum_{k=1}^{B/2} (w_j^{(l)} h_i^{(l-1)} - \mu^{(l-1)})^2},$$

where we denote $h_{ij}^{(l)}$, $\tilde{h}_{ij}^{(l)}$ and $\tilde{h}_{ij}^{(l)}$ as the output of ReLU, convolution or fully connected layer and BN layer respectively, for $i$-th sample, $j$-th output variable at $l$-th layer. The gradient with respect to $w_j^{(l)}$ is

$$\nabla_w w_j^{(l)} = \frac{\partial \mathcal{L}_a(\tilde{x}_j)}{\partial w_j^{(l)}} = \frac{\nabla \tilde{h}_{ij}^{(l)} (h_i^{(l-1)} - \mu^{(l-1)})}{\sqrt{2/B} \sum_{k=1}^{B/2} (w_j^{(l)} h_i^{(l-1)} - \mu^{(l-1)})^2}$$

$$- \frac{\nabla \tilde{h}_{ij}^{(l)} w_j^{(l)} (h_i^{(l-1)} - \mu^{(l-1)})}{(2/B) \sum_{k=1}^{B/2} (w_j^{(l)} h_i^{(l-1)} - \mu^{(l-1)})^2} \Sigma^{(l-1)} w_j^{(l)},$$

where $\mu^{(l-1)}$ and $\Sigma^{(l-1)}$ are the mean and covariance matrix from $(l-1)$-th layer, $\nabla \tilde{h}_{ij}^{(l)}$ represents the gradient of loss with respect to $\tilde{h}_{ij}^{(l)}$. If we write the weight as its norm $\|w_j^{(l)}\|$ multiplied by the unit vector $u_j^{(l)}$, there is a relationship between $\nabla w_j^{(l)}$ and $\|w_j^{(l)}\|$,

$$\|\nabla_w w_j^{(l)}\| = \frac{1}{\|w_j^{(l)}\|} \|\nabla_u u_j^{(l)}\|$$

Table 1. The robust accuracy drop of AT and TWINS-AT, where the adversarial attack is PGD10. TWINS-AT has smaller accuracy drop compared with AT, indicating that the TWINS-AT is less prone to robust overfitting as a result of reduced variance of gradient norms and stable training as shown in Fig. 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Rob. Acc.</th>
<th>CIF10</th>
<th>CIF100</th>
<th>Caltech</th>
<th>CUB</th>
<th>Dogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>31.84</td>
<td>31.38</td>
<td>49.09</td>
<td>27.08</td>
<td>21.19</td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td>49.41</td>
<td>28.52</td>
<td>48.37</td>
<td>26.60</td>
<td>19.80</td>
<td></td>
</tr>
<tr>
<td>Gap ↓</td>
<td>2.43</td>
<td>2.86</td>
<td>0.73</td>
<td>0.48</td>
<td>1.39</td>
<td></td>
</tr>
<tr>
<td>TWINS-AT</td>
<td>53.23</td>
<td>31.60</td>
<td>48.80</td>
<td>29.24</td>
<td>20.89</td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td>52.40</td>
<td>31.08</td>
<td>48.40</td>
<td>29.24</td>
<td>20.58</td>
<td></td>
</tr>
<tr>
<td>Gap ↓</td>
<td>0.83</td>
<td>0.52</td>
<td>0.40</td>
<td>0.00</td>
<td>0.29</td>
<td></td>
</tr>
</tbody>
</table>
This relationship has been found in [2, 27] and extended to any scale-invariant layers such as layer normalization [3] by [40]. It means that there are two ways to increase the gradient magnitude or the effective learning rate: 1) find a steeper descent direction where \( \| \nabla \alpha w_j^{(l)} \| \) is increased and 2) decrease the weight norm so that \( 1/\| w_j^{(l)} \| \) is increased. For a DNN with standard BN layers, the training will exploit this property to increase the gradient magnitude by reducing the weight norms with the help of weight decay regularization, which leads to a spurious increase in gradient magnitude and a larger variance of gradient estimation. In contrast, for a DNN with fixed BN layers such as the Frozen Net, the gradient norm is not correlated with weight norm,

\[
\nabla_f w_j^{(l)} = \frac{\partial L_f(x_i)}{\partial w_j^{(l)}} = \nabla h_i^{(l)} \frac{h_i^{(l-1)}}{\sigma_j^{(l)pt}}.
\]  

(8)

In this gradient, the only way to increase the gradient magnitude is to find the actual steeper direction. The overall gradient for the weight is

\[
\Delta w_j^{(l)} = \nabla \alpha w_j^{(l)} + \lambda_{\text{twins}} \nabla_f w_j^{(l)}. 
\]  

(9)

**Empirical Study**

To see the difference between the gradients in AT and TWINS-AT, we record the gradient of all weight parameters in each step of the two training methods and compute the mean and STD of the gradient norm in each epoch. The model parameters and their gradients are treated as long vectors and we compute the \( l_2 \) norm of the weight and gradient vector. Figure 3 shows the mean and normalized STD of gradient norms in 60 training epochs for three datasets. Here we show the normalized STD, i.e., dividing the STD by the mean, to see the relative effect of variance. The major finding is that the gradient magnitude of TWINS-AT is substantially larger than that of AT, while the variance of TWINS-AT is lower than AT in most epochs. Note that in theory, the ratio between STD and mean should remain the same after down-scaling the weight norm in standard BN, but in practice we do observe the high normalized variance of DNNs with standard BNs since we use one epoch’s gradients to approximate the variance and mean.

One benefit of the larger gradient magnitude is that the model can escape from the initial sub-optimal point faster and find a better local optimum than the small gradient optimization [41]. We validate this hypothesis by recording the distance between the current model and the initial model during training. Figure 4 shows these weight distances on three datasets, where TWINS-AT moves away from the initial model much faster than the AT baseline. The robust overfitting effect of adversarial training [54] is partially a result of large gradient variance, especially at the final stage of training [9]. We compare the robust accuracy drop of AT and TWINS-AT in Table 1 and find that the small relative variance of TWINS-AT has the effect of reducing robust overfitting. See the experiment details in Sec. 5 and Supp.

5. Experiment

This section presents our experiment with TWINS. We first introduce our experiment setting and then show our main result and ablation study.

5.1. Experiment Setting

**Dataset.** We use five datasets in our experiment. CIFAR10 and CIFAR100 [34] are low-resolution image datasets, where the training and validation sets have 50,000 and 10,000 images, and CIFAR10 has 10 classes, while CIFAR100 has 100 classes. Caltech-256 [22] is a high-resolution dataset with 30,607 images and 257 classes, which is split into training and validation set with a ratio of 9:1. Caltech-UCSD Birds-200-2011 (CUB200) [60] is a high-resolution bird image dataset for fine-grained image classification, which contains 200 classes of birds, 5,994 training images and 5,794 validation images. Stanford Dogs [33] has high-resolution dog images from 120 dog categories, where the training and validation set has 12,000 and 8,580 images. For both low-resolution image datasets (CIFAR10 and CIFAR100) and high-resolution datasets (Caltech-256, CUB200 and Stanford Dogs), we resize the image to 224×224 so that the input sizes are the same for pre-training and fine-tuning. As with pre-training, the input image is normalized by the mean and STD of the pre-training set. Note that the resizing and normalization function is integrated into the model so we can attack the input image with the [0,1] bounds for pixel values as in standard adversarial attacks. We use the standard ImageNet data augmentation for high-resolution datasets [26]. For CIFAR datasets, we use random cropping with padding=4 and random horizontal flipping.

**Adversarial Pre-Training.** Large-scale adversarial pre-training on ImageNet is time-consuming, and thus [55] has released adversarially pre-trained ResNet50 and WideResNet50-2, trained with \( l_2 \) and \( l_\infty \) norm bounded attacks. In this paper, we adopt the pre-trained ResNet50 models, trained with \( l_\infty \) attack with bound \( \epsilon_{pt} = 4/255 \). We test other robust pre-trained models in our ablation study.

**Training Setting.** For baselines and our method, we train all parameters of the pre-trained model, i.e., full fine-tuning instead of linear probing [57], with PGD attacks of \( l_\infty \) norm. The PGD step is 10, \( \epsilon_{f1} = 8/255 \) and stepsize \( \alpha = 2/255 \). We set the batch size as 128 and train the model for 60 epochs and divide the learning rate by 0.1 at 30th and 50th epoch. In TWINS with warmup, we initialize the means and STDs with their pre-trained values, and update the means and STDs using the target training set.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>Caltech256</th>
<th>CUB200</th>
<th>Stanford Dogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Acc.</td>
<td>AT</td>
<td>89.77</td>
<td>69.48</td>
<td>75.90</td>
<td>65.74</td>
<td>60.09</td>
</tr>
<tr>
<td></td>
<td>TWINS-AT</td>
<td>91.24 (+1.47)</td>
<td>70.72 (+1.24)</td>
<td>76.86 (+0.96)</td>
<td>68.09 (+2.35)</td>
<td>64.98 (+4.89)</td>
</tr>
<tr>
<td></td>
<td>TWINS-AT+warmup</td>
<td>91.95 (+2.18)</td>
<td>72.12 (+2.64)</td>
<td>77.35 (+1.45)</td>
<td>67.64 (+1.90)</td>
<td>66.12 (+6.03)</td>
</tr>
<tr>
<td>TRADES</td>
<td>AT</td>
<td>87.06</td>
<td>62.76</td>
<td>69.70</td>
<td>58.92</td>
<td>59.99</td>
</tr>
<tr>
<td></td>
<td>TWINS-TRADES</td>
<td>86.61 (-0.45)</td>
<td>66.72 (+3.96)</td>
<td>71.12 (+1.42)</td>
<td>60.72 (+1.80)</td>
<td>60.58 (+0.59)</td>
</tr>
<tr>
<td></td>
<td>TWINS-TRADES+warmup</td>
<td>86.60 (-0.46)</td>
<td>65.91 (+3.15)</td>
<td>73.39 (+3.69)</td>
<td>61.05 (+2.13)</td>
<td>63.96 (+3.97)</td>
</tr>
</tbody>
</table>

Table 2. The performance of our TWINS-AT and TWINS-TRADES on five image classification tasks compared with AT and TRADES. The clean accuracy means the accuracy when testing images are input without adversarial perturbations. PGD10 and AA denote the robust test accuracy under PGD10 and AutoAttack. The increase and decrease in performance are denoted with green and red numbers. The bold numbers denote the best robust accuracy under AA. The proposed TWINS achieves better robustness and clean accuracy compared with the baseline. Averaged over the datasets, the clean and robust accuracy of TWINS are increased by 2.18% and 1.21% compared with AT, and 1.46% and 0.69% compared with TRADES. The means and STDs of the performance are in the supplemental.

<table>
<thead>
<tr>
<th>Metric</th>
<th>PT Model</th>
<th>Caltech256</th>
<th>CUB200</th>
<th>Dogs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Acc.</td>
<td>Random</td>
<td>48.99</td>
<td>12.38</td>
<td>7.27</td>
</tr>
<tr>
<td></td>
<td>Non-Robust</td>
<td>64.66</td>
<td>53.30</td>
<td>41.99</td>
</tr>
<tr>
<td></td>
<td>Robust</td>
<td>75.90</td>
<td>65.74</td>
<td>60.09</td>
</tr>
<tr>
<td>PGD10</td>
<td>Random</td>
<td>31.78</td>
<td>3.728</td>
<td>3.59</td>
</tr>
<tr>
<td></td>
<td>Non-Robust</td>
<td>39.86</td>
<td>19.68</td>
<td>13.32</td>
</tr>
<tr>
<td></td>
<td>Robust</td>
<td>48.37</td>
<td>26.60</td>
<td>19.80</td>
</tr>
</tbody>
</table>

Table 3. Comparison of random initialization, non-robust and robust pre-trained model on three difficult classification tasks, when fine-tuned with AT. The robust pre-trained model is indispensable to downstream robustness.

The momentum of updating statistics is 0.1, the batch size is 128, and the warmup only lasts one epoch. Note that in the warmup stage, the input samples are added with adversarial perturbations generated by the PGD attack, which has the same setting as the attack in training, and the classifier layer is the pre-trained classifier for the adversarial attack. Our pilot experiment shows that using adversarial examples as input is more effective than using clean examples in the warmup. The optimizer is SGD with momentum in all of our experiments. The learning rate, weight decay and regularization hyperparameter are determined by grid search, which is described in detail in the supplemental.

Adversarial robustness evaluation. Two standard adversarial attacks are used in our experiment, i.e., PGD and AutoAttack [11]. The adversarial perturbation is $\ell_\infty$ norm bounded in our evaluation. The setting of PGD attack for validation set is the same as the PGD attack in training. The AutoAttack (AA) is a more reliable adversarial attack and more often used for evaluation than PGD in recent years.

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>$\lambda_{WD}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>Clean Acc.</td>
<td>89.92</td>
</tr>
<tr>
<td></td>
<td>Robust Acc.</td>
<td>46.34</td>
</tr>
<tr>
<td>TWINS-AT</td>
<td>Clean Acc.</td>
<td>91.90</td>
</tr>
<tr>
<td></td>
<td>Robust Acc.</td>
<td>47.19</td>
</tr>
</tbody>
</table>

Table 4. The performance of TWINS-AT and AT on CIFAR10 when the hyperparameter for weight decay $\lambda_{WD}$ is changed. The robust accuracy is evaluated using AutoAttack. Our TWINS-AT achieves better adversarial robustness than AT for different $\lambda_{WD}$.

We use the standard attacks of AA, i.e., untargeted APGD-CE, targeted APGD-DLR, targeted FAB [10] and Square Attack [1], with $\epsilon = 8/255$. The robust accuracy in our experiment result denotes the accuracy under AA.

5.2. Experimental Result

On the three high-resolution datasets, we compare the performance of fine-tuning different initialization models, i.e., random initialization, standard pre-trained ResNet50 and robust pre-trained ResNet50. Table 3 shows that the pre-trained models are essential to the accuracy and robustness in challenging downstream tasks, since the random initialization is much worse than the two pre-trained models. The robust pre-trained model has a clear benefit over the standard pre-trained one, indicating that robust pre-training is indispensable to downstream robustness.

Table 2 shows the result of TWINS-AT and TWINS-TRADES compared with the baselines. Since AA is a more reliable attack, we highlight the best robust accuracy un-
under AA on each dataset. The TWINS fine-tuning learns more robust DNNs, as well as achieves a better clean accuracy on all five datasets, demonstrating the strong effectiveness of TWINS in the robust pre-training and robust fine-tuning setting. On CIFAR10 and CIFAR100, both TWINS-AT and TWINS-TRADES achieve better robustness and clean accuracy than their baselines, while the warming only improves the clean accuracy and hurts the robustness sometimes. On Caltech-256, TWINS-AT improves upon the vanilla AT in both robustness and accuracy, but TWINS-TRADES does not perform better than vanilla TRADES. However, the warming helps boost the performance of TWINS-TRADES as well as TWINS-AT and makes the TWINS with warming perform better than the baselines.

On the two fine-grained image classification datasets, TWINS-AT and TWINS-TRADES generally perform better than baselines in terms of accuracy and robustness, if we look at the robust accuracy under AA, where only TWINS-AT on CUB200 has a slightly worse robust accuracy than its baseline. We find that the warming improves the clean accuracy but hurts the robustness in most cases, except for CIFAR100 and Caltech-256. This can be a result of noisy adversarial perturbation, since we use the pre-trained classifier layer in the adversarial attack, or the insufficient update steps. Nevertheless, we note that the warming can be considered as an operation for achieving a trade-off between robustness and accuracy.

5.3. Ablation Study

TWINS initialization. We use the pre-trained BN statistics in the Frozen Net to keep the robust information learned during pre-training. To show the importance of the pre-trained BN statistics, we use the standard initialization (mean=0 and STD=1) for BN statistics in the Frozen Net, denoted as TWINS-Init, and show the result on the five datasets in Figure 5. Both clean and robust accuracy drop when the (0,1) initialization is used in TWINS-Init, demonstrating the crucial role of pre-trained statistics in TWINS. The fact that TWINS-Init sometimes improves upon the AT baseline motivates us to investigate the effect of TWINS on gradient norms in Section 4.

Effect of weight decay. Weight decay is the reason for the decreasing weight norm in DNNs with BN layers, so increasing the hyperparameter of weight decay $\lambda_{WD}$ is also a way to increase the gradient magnitude. Table 4 show the result of TWINS-AT and AT when different $\lambda_{WD}$ are used when fine-tuning the robust pre-trained model on CIFAR10. The robust accuracy of TWINS-AT is consistently better than that of AT across different $\lambda_{WD}$'s. We draw the same conclusion on CIFAR100 (see supplemental). Note that the clean accuracy of TWINS-AT drops when a large $\lambda_{WD}$ is used, suggesting that we should not use a too large $\lambda_{WD}$ for TWINS-AT.

Different robust pre-trained models. The main experiment uses the robust pre-trained ResNet50 with $\epsilon_{pt}=4/255$ as the initial model. We try different robust pre-trained models with different $\epsilon_{pt}$ in Table 5, which shows that a larger $\epsilon_{pt}$ is beneficial to both clean and robust accuracy in the downstream, and the proposed TWINS-AT is better than AT in both metrics with different pre-trained models.

6. Conclusion

This paper investigates the utility of robust pre-trained models in various downstream classification tasks. We first find that the commonly used data- and model-based approaches to maintain pre-training information do not work in the adversarial robust fine-tuning. We then propose a subtle statistics-based method, TWINS, for retaining the pre-training robustness in the downstream. In addition to the robustness preserving effect, we find that TWINS increases the gradient magnitude without sacrificing the training stability and improves the training dynamics of AT. Finally, the performance of TWINS is shown to be stronger than that of AT and TRADES on the five datasets. One limitation of our work is that we only evaluate the robust supervised pre-trained ResNet50. Recently, robust pre-trained ViT's on ImageNet [4] have been released. Our statistics-based approach can be extended to the layer normalization, on which the increasing gradient magnitude argument also holds, and thus future work will extend TWINS to ViT.

Acknowledgement This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11215820) and the Fundamental Research Funds for the Central University of China (DUT No. 82232031).
References


ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 1, 6


[46] Ziquan Liu, Yi Xu, Yuhong Xu, Qi Qian, Hao Li, Antoni Chan, and Rong Jin. Improved fine-tuning by better leveraging pre-training data. In Thirty-Sixth Conference on Neural Information Processing Systems, 2022. 1, 4


