Two Sides of the Same Coin: White-box and Black-box Attacks for Transfer Learning

Yinghua Zhang
yzhangdx@cse.ust.hk
CSE, HKUST, Hong Kong, China

Yangqiu Song
yqsong@cse.ust.hk
CSE, HKUST, Hong Kong, China
Peng Cheng Laboratory, Shenzhen, China

Jian Liang
joshualiang@tencent.com
Cloud and Smart Industries Group, Tencent, China

Kun Bai
kunbai@tencent.com
Cloud and Smart Industries Group, Tencent, China

Qiang Yang
qyang@cse.ust.hk
CSE, HKUST, Hong Kong, China
WeBank, China

ABSTRACT

Transfer learning has become a common practice for training deep learning models with limited labeled data in a target domain. On the other hand, deep models are vulnerable to adversarial attacks. Though transfer learning has been widely applied, its effect on model robustness is unclear. To figure out this problem, we conduct extensive empirical evaluations to show that fine-tuning effectively enhances model robustness under white-box FGSM attacks. We also propose a black-box attack method for transfer learning models which attacks the target model with the adversarial examples produced by its source model. To systematically measure the effect of both white-box and black-box attacks, we propose a new metric to evaluate how transferable are the adversarial examples produced by a source model to a target model. Empirical results show that the adversarial examples are more transferable when fine-tuning is used than they are when the two networks are trained independently.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Security and privacy**;

KEYWORDS

Transfer Learning, Neural Networks, Adversarial Attacks

ACM Reference Format:
Yinghua Zhang, Yangqiu Song, Jian Liang, Kun Bai, and Qiang Yang. 2020. Two Sides of the Same Coin: White-box and Black-box Attacks for Transfer Learning . In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403349

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403349

1 INTRODUCTION

Deep learning models achieve state-of-the-art performances on a wide range of computer vision tasks. Yet the performance is achieved at the cost of large scale labeled training data. In practice, there are many domains where labeled data are insufficient to train a deep model from scratch. In such cases, transfer learning techniques [16, 24] are usually adopted. Transfer learning uses the knowledge that is extracted from a well-annotated *source* domain to help learning in a *target* domain where only limited labeled data are available. One of the most successful and popular transfer learning techniques is fine-tuning. For example, it is demonstrated that the parameters in a convolutional neural network (CNN) are transferable [15, 25]. Nowadays, there are many pre-trained networks publicly available and developers often use them to save the efforts on data labeling and model training.

However, a perturbation that is imperceptible to humans can easily fool a deep learning models such as a well-performed complex CNN [20]. A typical example was given in [20] where a panda image is misclassified as "gibbon." Though there are a lot of successful stories of transfer learning, surprisingly, few studies consider the robustness of transfer learning models. It is found that adversarial examples can generalize across the networks with different architectures that are trained on the same dataset [12] or the networks that are trained with disjoint datasets [20]. These studies have proven that adversarial examples could be transferable. However, they are not directly dealing with transfer learning models. This motivates us to think about to which extent we can generate adversarial examples for transfer learning.

Generating adversarial examples for transfer learning is not a trivial problem. An adversarial attack can be either white-box or black-box. White-box attacks assume that the target of the attack is accessible while black-box attacks only allow querying the network output or even have no knowledge of the network. Thus, there are mainly two challenges to be considered. First, in the case of white-box attacks, although it has been proved that adversarial training can be transferred to the target domain [7], it is still unclear what is the effect of pure fine-tuning for the resultant model or whether there exists a general way of attacks when only a giant model trained on a large scale dataset (e.g., ImageNet) is available. Second, in the case of black-box attacks, to our best knowledge, there has been no study on how transfer learning would affect

model robustness under black-box attacks. A trivial solution may be directly using the adversarial examples in the source domain that is used for pre-training. However, in many target domains used for fine-tuning, the label sets are different from the source-domain labels. Therefore, it is difficult to apply this trivial solution.

In this paper, we study both white-box and black-box attacks for a simple transfer learning paradigm: the pre-training and fine-tuning procedure of domain adaptation of a CNN model. We find this simple transfer learning paradigm shows more robustness under the white-box FGSM attacks. For the black-box attack, we propose a simple attack method that attacks the fine-tuned model with the adversarial examples produced by the source model. Experimental results show that this method is simple yet effective and it hurts the robustness results. To systematically measure the effect of both white-box and black-box attacks, we propose a new metric to evaluate how transferable are the adversarial examples produced by a source domain network to a target domain network using both white-box and black-box attack results. Without loss of generality, we evaluate the following two transfer learning settings.

• The source domain is similar to the target domain. Then we directly transfer the source domain model to the target domain.

• There exists a giant model trained on a general large dataset. However, the similarity of source and target domains does not support to generate adversarial examples for the target domain. In this case, we introduce another source domain which is similar to the target domain and also fine-tuned from the giant model.

Empirical results show that the adversarial examples are more transferable when fine-tuning is used than they are when the two networks are trained independently.

In addition to improved transfer performance and robustness under white-box attacks when applying fine-tuning, our study suggests that the benefits are obtained at the cost of the potential risks of using untrusted pre-trained networks. A malicious attacker can take advantage of this phenomenon by releasing a pre-trained model and attack the downstream fine-tuned models. While most transfer learning methods only optimize for a low generalization error, we argue that the robustness of transfer learning models should be considered as well. Otherwise, we may expose transfer learning models under harmful attacks. Such risk has been overlooked which can be dangerous for safety-critical applications such as autonomous driving.

The rest of this paper is organized as follows. We review related works in Section 2. In Section 3, we introduce the problem settings and white-box and black-box attack methods for transfer learning models. The experiment setup is described in Section 4, and numerical results under white-box and black-box attacks are presented in Section 5. Ablation experiments are shown in Section 6. We summarize and discuss the empirical results in Section 7, and finally conclude the paper in Section 8. Our source code is available at https://github.com/HKUST-KnowComp/AttackTransferLearning.

2 RELATED WORKS

Transfer learning is necessary to overcome the data-hungry nature of neural networks [16, 24]. Fine-tuning is one of the most popular transfer learning methods. Using the networks that are pre-trained on large scale datasets such as ImageNet [18] can significantly boost the performance of downstream tasks, such as video classification, object detection, image/video captioning, etc [5, 9, 21–23].

In pursuit of machine learning models that are both robust and efficient, adversarial attacks and defenses have attracted attention in the past few years. Numerous attack and defense methods have been proposed [26]. Szegedy et al. [20] first propose an L-BFGS method to craft adversarial examples that are close to the original examples and misclassified by the network. L-BFGS attack is effective but slow. To improve efficiency, Goodfellow et al. propose a one-step attack method FGSM by moving along the direction of the gradient [6]. Madry et al. [13] formulate a min-max optimization problem to study adversarial robustness. They start from a random perturbation around the original input and strengthen the gradient-based attack by applying it iteratively with a small step size.

Previous works that study model robustness assume that the model is trained from scratch in an individual domain while transfer learning techniques are often used in practice and the assumption no longer holds in such settings. Though fine-tuning has been widely used, surprisingly, few research works pay attention to the robustness of transfer learning models. The most related work is [7] where the robustness of adversarial fine-tuned models under white-box attacks is evaluated. Our work differs with theirs in three aspects: (1) We study the robustness of fine-tuned models under both white-box and black-box attacks; (2) We focus on fine-tuning which is more widely used instead of adversarial fine-tuning proposed in [7]; (3) Ablation experiments are conducted to study the effect of a number of factors, including the number of labeled data, domain similarity, network architectures, etc., on the transfer performance and robustness.

3 ROBUSTNESS OF TRANSFER LEARNING MODELS

We first introduce the problem settings and the notations used, then present three model training strategies. We briefly describe how to generate adversarial examples in an individual domain, and then propose the method to attack transfer learning models. We finally introduce a new metric to measure the transferability of adversarial examples between transfer learning models.

3.1 Problem Setup

We focus on the classification task where both domains are labeled. To avoid the confusion with the "target" of attack, the source domain and the target domain are called `Domain A` and `Domain B`, and denoted by $\mathcal{D}_A$ and $\mathcal{D}_B$, respectively. `Domain A` is composed of n_A training samples which is denoted by $\mathcal{D}_A = \{(\mathbf{x}_{A_j}, y_{A_j})\}_{j=1}^{n_A}$ where there are n_A training samples $\mathbf{x}_A \in \mathcal{X}_A$ and their corresponding labels $y_A \in \mathcal{Y}_A$. Similarly, `Domain B` is denoted by $\mathcal{D}_B = \{(\mathbf{x}_{B_j}, y_{B_j})\}_{j=1}^{n_B}$ and there are $\mathbf{x}_B \in \mathcal{X}_B$ and $y_B \in \mathcal{Y}_B$. There are many more labeled data in `Domain A` than there are in `Domain B`, i.e., $n_A \gg n_B$. When there are $\mathcal{X}_A = \mathcal{X}_B$ and $\mathcal{Y}_A = \mathcal{Y}_B$, the setting is referred to as homogeneous transfer learning. Otherwise, it is referred to as heterogeneous transfer learning. In each domain, a neural network, denoted by f, is trained to learn the mapping from the input space to the label space, $f : \mathcal{X} \to \mathcal{Y}$. The output of the

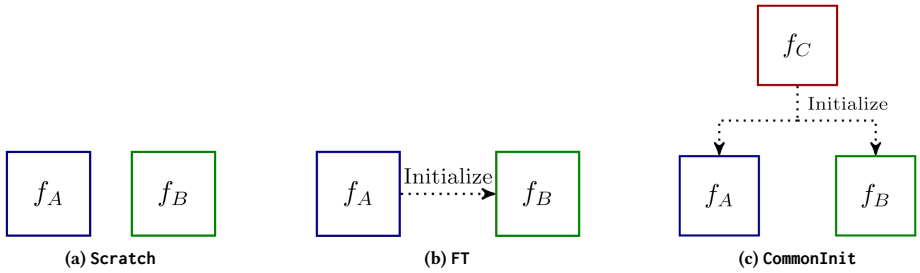


Figure 1: Three model training strategies. There is no transfer learning and `Model A` and `Model B` are independent if they are trained with the `Scratch` strategy. Transfer learning is involved and two models are correlated explicitly/implicitly when the `FT`/`CommonInit` strategy is used.

network, denoted by $f(\mathbf{x})$, predicts the probability distribution over the label space.

3.2 Model Training Strategies

As attacking transfer learning can be similar or different from attacking non-transfer learning models, we consider three different training strategies to study the adversarial example generation for comparing transfer learning with non-transfer learning. The three ways to train the models in the two domains, namely `Scratch`, `Fine-tune`, and `CommonInit`, are shown in Fig. 1. While `Scratch` is not a transfer learning setting, the other two strategies both involve transfer learning. The `Fine-tune` and `CommonInit` strategies address the homogeneous and heterogeneous transfer learning settings, respectively. The details of the three training strategies are described as follows.

- `Scratch`: As shown in Fig. 1a, in the `Scratch` setting, the `Model B` is randomly initialized and is only trained with `Domain B`'s data. There is no transfer learning if the models are trained with the `Scratch` strategy, and `Model A` and `Model B` are independent.
- `Fine-tune (FT)`: The `FT` strategy is shown in Fig. 1b. To transfer the parameters from `Domain A` to `Domain B`, the two networks share an identical architecture. `Model A` (f_A) is first trained with `Domain A`'s data, then `Model B` is initialized with the parameters of `Model A`. Finally, `Model B` (f_B) is fine-tuned with `Domain B`'s data.
- `CommonInit`: Both `Model A` and `Model B` are initialized with another `Model C` and then fine-tuned with domain-specific data. This approach is useful for black-box attacking a heterogeneous transfer learning model. For example, to train a model on `STL10` where there are only 5,000 training images, a natural choice of the source domain is a downsampled variant of ImageNet [2], denoted by `ImageNet32`, where there are more than one million training images. However, the label spaces of the two domains do not agree. There are 10 classes and 1,000 classes in `STL10` and `ImageNet32`, respectively. We cannot attack an `STL10` model with the adversarial examples produced by an `ImageNet32` based model due to the mismatched label space. One solution to the problem is to use another domain such as `CIFAR10` as the source domain. `CIFAR10` has the same label space as `STL10` does. Thus, the adversarial examples generated based on `CIFAR10` can be transferred to attack models based on `STL10`. In our setting, both models for `CIFAR10` and `STL10` are fine-tuned from `ImageNet32` and the adversarial examples produced by the `CIFAR10` model can be more transferable to attack the `STL10` based model. In this example, `CIFAR10`, `STL10`, and `ImageNet32` correspond to `Domain A`, `B`, and `C` in Fig. 1c, respectively.

3.3 Generate Adversarial Examples

Before attacking transfer learning models, we first introduce adversarial example generation in an individual domain as preliminary knowledge. There are different ways to generate adversarial examples. As our study mainly focuses on different transfer learning settings and study the transferability of the generated examples in different settings, we choose the widely used Fast Sign Gradient Method (FGSM) [6] in our work.

In general, crafting adversarial examples for a model f can be formulated as an optimization problem:

$$\arg\min_{\|\hat{\mathbf{x}}-\mathbf{x}\|_p \le \epsilon} \ell(\hat{y}, f(\hat{\mathbf{x}})), \tag{1}$$

where $\hat{\mathbf{x}}$ denotes the adversarial example, $\hat{y}$ denotes a label that is different from the ground truth label y, $\ell(\cdot,\cdot)$ denotes a classification loss, $\|\cdot\|_p$ denotes the p-norm distance and ϵ denotes the perturbation budget. The adversarial example is optimized to mislead the network f within the p-norm ϵ-ball of the clean example $\mathbf{x}$. In this paper, we adopt the cross-entropy loss as the classification loss and the infinity norm as the distance measure.

Many methods to solve the optimization problem in Eq. (1) have been proposed [6, 13, 20]. The FGSM takes one step in the direction of the gradient:

$$\hat{\mathbf{x}} = \mathbf{x} + \epsilon \cdot sgn(\nabla_{\mathbf{x}} \ell(y, f(\mathbf{x}))), \tag{2}$$

where the *sgn* function extracts the sign of each dimention in the gradient $\nabla_{\mathbf{x}} \ell(y, f(\mathbf{x}))$ and uses that as the direction to slightly modify the given example. The FGSM update is believed to optimize Eq. (1) to generate some valid examples that are imperceptible to humans but may fool a deep learning model [6].

3.4 Attack Transfer Learning Models

We consider the robustness of transfer learning models under both white-box and black-box attacks in this paper. Particularly, we apply the FGSM attack method to generate adversarial examples

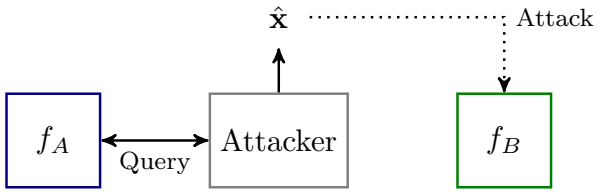


Figure 2: Black-box attack `Model B` with adversarial examples produced by `Model A`. The proposed method allows attack without any query to the target model.

for transfer learning models under both white-box and black-box settings.

3.4.1 White-box Attacks. For white-box attacks, it is assumed that everything related to the target of the attack is accessible. Thus, we can view `Model B` as a white box and attack it by applying FGSM. `Model B` is either trained from scratch in `Domain B` or fine-tuned from a source network `Model A`.

3.4.2 Black-box Attacks. The assumption of white-box attacks usually does not hold in reality. A more realistic setting is the *black-box* attack where the target of the attack is completely not accessible or only the output of the target can be queried. While most black-box attack methods are developed for the setting that allows querying the output of the target model [1, 17], we focus on a more restricted setting where no access to the target model is allowed.

We develop a simple black-box attack method for transfer learning models which first produces an adversarial example with `Model A` and then attacks `Model B` with the generated adversarial example. Hence we can attack `Model B` without any access to it. The procedure is illustrated in Fig. 2. The two models can be trained with three different strategies introduced in Section 3.2. We will apply the proposed black-box attack method and evaluate model robustness in Section 5.2.

3.5 Transferability of Adversarial Examples

Here, we propose a new metric to measure the transferability of adversarial examples between transfer learning models. Similar to the evaluation metrics for the robustness under white-box attacks, the robustness under black-box attacks can be measured by the adversarial accuracy. A lower adversarial accuracy indicates that the model is more vulnerable to the transferred adversarial examples.

As will be shown in Section 5.1, the model robustness can be enhanced after fine-tuning, and hence it may be unfair to directly compare the adversarial accuracy of the `Scratch` model and that of a transfer learning model. We introduce a new metric to evaluate the transferability of adversarial examples between `Model A` and `Model B`. Let a_w and a_b denote the adversarial accuracy of a network under the white-box attack and black-box attack, respectively. Let γ denote the transferability metric defined as:

$$\gamma = \frac{a_b - a_w}{a_w}, \tag{3}$$

which measures how much the adversarial accuracy under the black-box attack deviates from the one under the white-box attack. Usually, we have $\gamma > 0$ since less knowledge is available in the black-box setting and black-box attacks are not as effective as the white-box ones. If $\gamma \leq 0$, it means that the black-box attacks by transfer learning are even more effective than directly attacking the target model.

Domain			n_A	n_B	n_C
A	B	C			
M	U	S	$60K$	74	$604K$
U	M	S	$7.4K$	600	$604K$
S	M	NA	$604K$	600	NA
S	Syn	M	$604K$	$4.8K$	$60K$
CIFAR	STL	ImageNet32	$45K$	$4.5K$	$1.28M$

Table 1: Statistics of transfer tasks.

4 EXPERIMENT SETUP

In this section, we introduce the datasets, evaluation metrics and implementation details in the following.

4.1 Transfer Tasks

We use seven datasets for our evaluation, which are `MNIST (M)` [11], `USPS (U)` [8], `SVHN (S)` [14], `SynDigits (Syn)` [4], `CIFAR10` [10], `STL10` [3], and `ImageNet32` [2]. The first four datasets contain "0" to "9" digit images with various distributions. Both `M` and `U` are handwritten digit databases while `S` and `Syn` are digit images with colored backgrounds. The latter three datasets are composed of low-resolution natural images. Five transfer tasks in the form of (`Domain A`, `Domain B`, `Domain C`) are constructed[1]. Their statistics are described in Table 1. We follow the default train/test split of the datasets. As preprocessing, all the images are resized to 32×32 and they are rescaled to the range $[-1, 1]$.

4.2 Evaluation Metrics

Since the five transfer tasks are all classification tasks, the classification accuracy on clean examples is adopted to measure the transfer performance in `Domain B`. A higher classification accuracy indicates better transfer performance. In addition to the transfer performance, robustness under adversarial attacks is considered as well. The robustness of a neural network is measured by the classification accuracy on the adversarial examples, which is referred to as *adversarial accuracy* in the following. The adversarial examples are obtained on the test set of `Domain B`. The clean examples that are correctly predicted by the target model are attacked. The higher the adversarial accuracy is, the more robust the network is.

4.3 Implementation Details

All the experiments are implemented with the PyTorch deep learning framework. Two network architectures are adopted. For the digit classification tasks, a simple 5-layer CNN, denoted by `DTN`, is used. For the `CIFAR` → `STL` task, a more expressive architecture, the 28-10 wide residual network (`WideRes`) [27], is used. When

[1] If n_B is smaller than the size of the default training set, we randomly sample n_B examples from it. Usually, there are a large amount of data in `Domain C`. Since there are only $7.4K$ training examples in `U`, we do not use `U` as `Domain C`. For the `CIFAR` → `STL` task, 9 categories that are shared by the two domains are used.

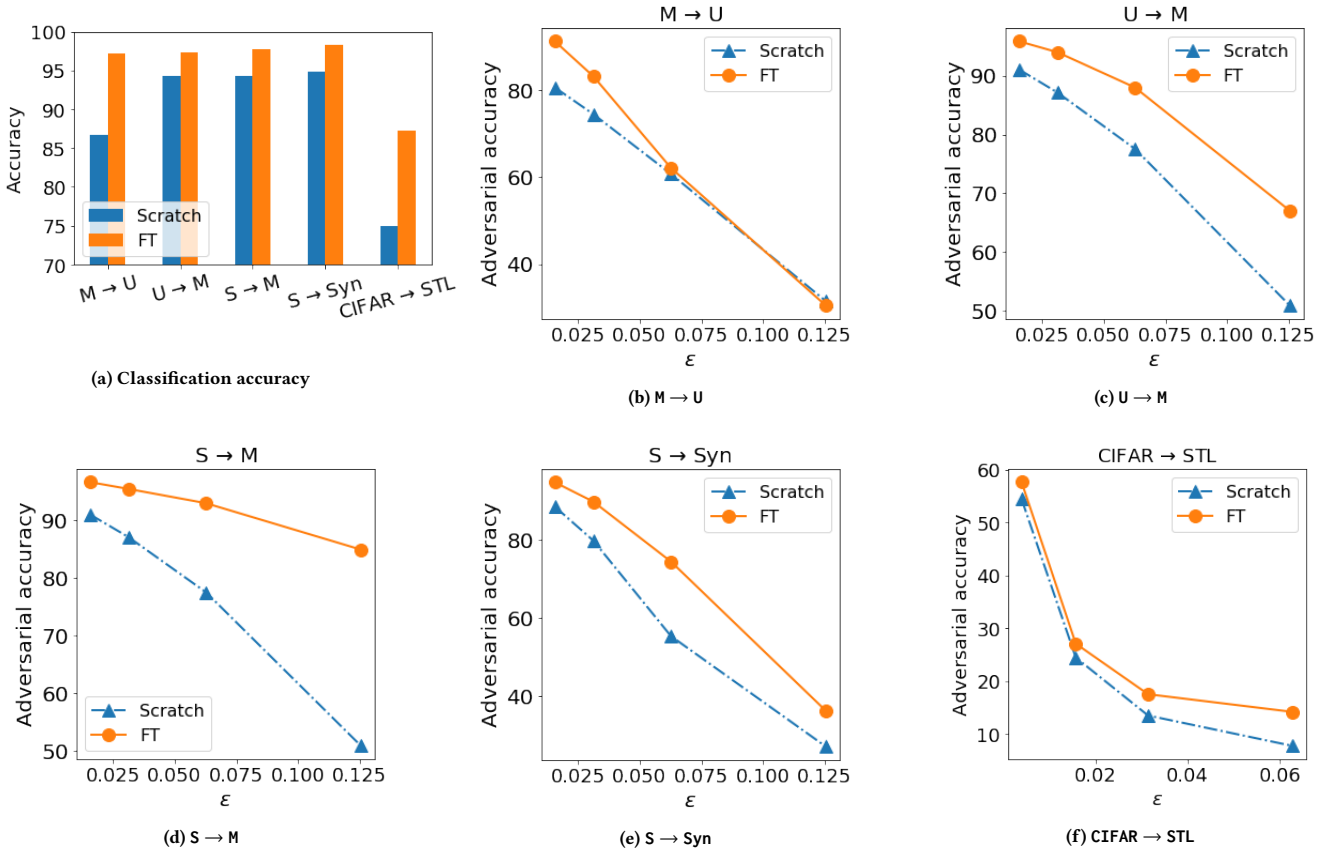


Figure 3: (a) Classification accuracy of the five transfer tasks. The FT models consistently outperform the Scratch baselines. (b-f) Robustness under white-box FGSM attacks. Compared to the Scratch models, the adversarial accuracy increases after fine-tuning, which indicates that the FT models are more robust than the Scratch ones.

ImageNet32 is used as Domain C, Model A and B are initialized with Model C except for the final classification layer. The neural networks are optimized with the mini-batch stochastic gradient descent with the momentum of 0.9. Early stopping is used, that is, if the network performance does not improve within 50 epochs, the training process is terminated. The batch size equals to 128. The learning rate is selected from $\{0.1, 0.01, 10^{-3}\}$ and weight decay is selected from $\{5 \times 10^{-4}, 2.5 \times 10^{-5}, 5 \times 10^{-6}\}$. We report the accuracy achieved with the optimal hyperparameters.

5 MAIN RESULTS

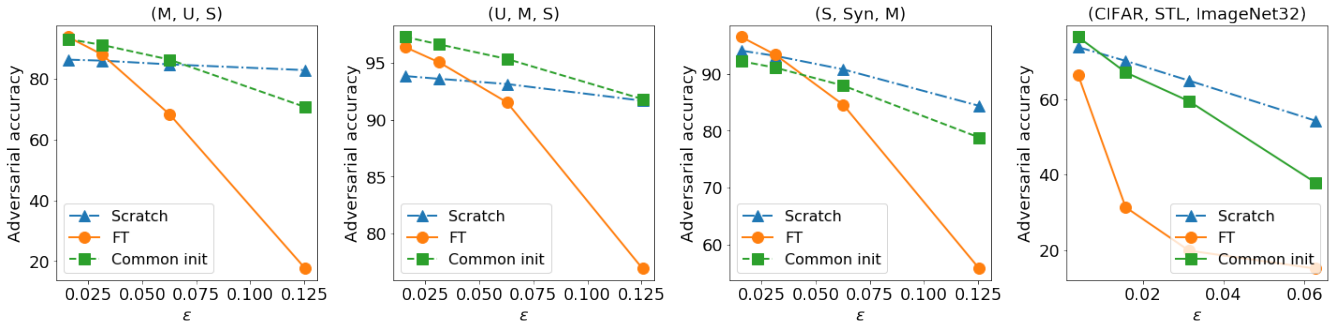
In this section, we study the effect of transfer learning on the robustness of Model B under attacks.

5.1 Under White-box Attacks

The classification results of the five transfer tasks are shown in Fig. 3a. On all transfer tasks, fine-tuning brings noticeable improvement over the Scratch baselines, which demonstrates the necessity and effectiveness of transfer learning. The adversarial accuracy of the five transfer tasks are shown in Fig. 3. We report adversarial accuracies under multiple perturbation budgets. Compared to the adversarial accuracy of Scratch models, the adversarial accuracy increases after fine-tuning. The improvement is more obvious when the network is attacked with a large perturbation budget. For example, the adversarial accuracy rises from 50.86% to 84.96% on the S → M task when $\epsilon = 0.125$. The results show that in addition to better transfer performance, another advantage of fine-tuning is the enhanced model robustness under white-box attacks.

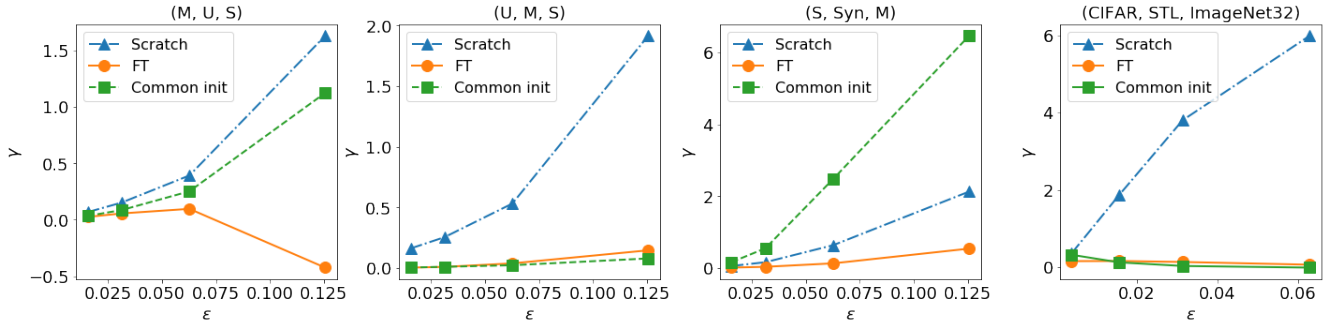
5.2 Under Black-box Attacks

The adversarial accuracy and the transferability of the four transfer tasks are shown in Figs. 4 and 5. In terms of the absolute adversarial accuracy values, the adversarial accuracy drops as the perturbation budget ϵ increases. While the adversarial accuracy of the Scratch model remains rather stable under multiple ϵ values, the adversarial accuracies of CommonInit and FT models drop significantly. Consequently, when the perturbation budget ϵ is large, the adversarial accuracies of CommonInit and FT models are much lower than those of the Scratch model. When there is Domain (A, B, C) = (M, U, S), the adversarial accuracy of the Scratch model remains larger than 80% while the adversarial accuracies of the CommonInit and FT model are only 70.59% and 17.53% when $\epsilon = 0.125$, respectively.



(a) Domain (A, B, C) = (M, U, S) (b) Domain (A, B, C) = (U, M, S) (c) Domain (A, B, C) = (S, Syn, M) (d) Domain (A, B, C) = (CIFAR, STL, ImageNet32)

Figure 4: Robustness (adversarial accuracy) under black-box attacks. The adversarial accuracies of the FT and CommonInit models drop drastically when the perturbation budget ϵ increases. They are much lower than those obtained with the Scratch models, which indicates that the fine-tuned models are likely to be attacked by the adversarial examples produced by their source models.



(a) Domain (A, B, C) = (M, U, S) (b) Domain (A, B, C) = (U, M, S) (c) Domain (A, B, C) = (S, Syn, M) (d) Domain (A, B, C) = (CIFAR, STL, ImageNet32)

Figure 5: Robustness (transferability γ) under black-box attacks. The new metric γ considers both white-box and black-box attack results and evaluates how transferable are the adversarial examples produced by Model A to Model B. When γ drops below 0, it means that Model B is more vulnerable to the adversarial examples transferred from Model A than those crafted directly with Model B as a white-box.

When the model robustness is measured by the transferability γ, the γ values of the CommonInit and FT models are usually smaller than those of the Scratch model, which indicates that Model B is likely to be successfully attacked by the adversarial examples produced by Model A if the parameters of the two models are correlated either explicitly or implicitly. An exception is found on the Domain (A, B, C) = (S, Syn, M) task where the γ values of the CommonInit model are larger than those of the Scratch model. We hypothesize that this is because the source domain M which is composed of handwritten digits is quite different from the two target domains S and Syn. The γ value of the FT model drops below 0 on the Domain (A, B, C) = (M, U, S) task when $\epsilon = 0.125$, which means that Model B is more vulnerable to the adversarial examples transferred from Model A than those crafted directly with Model B as a white-box. The results of the black-box attacks show that fine-tuning might introduce potential risks of being attacked by its source model, which are unaware of previously.

6 ABLATIONS

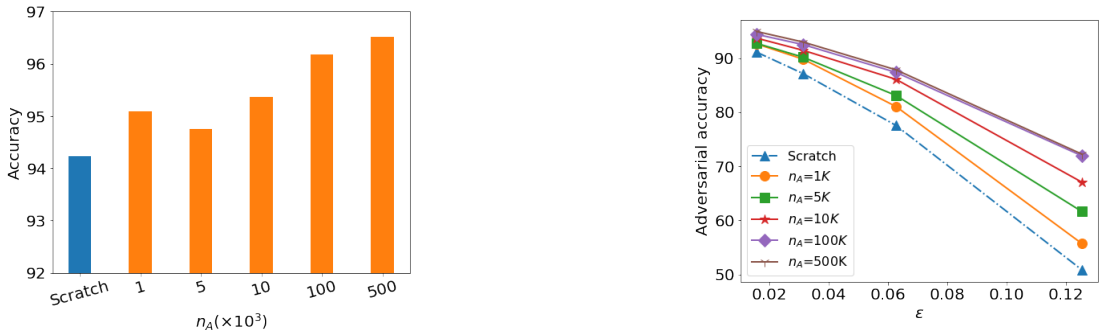
There are a number of factors that might affect the transfer performance and model robustness. To provide further insights into the effect of transfer learning, we conduct ablation experiments to study the following factors: (1) The number of training examples in Domain B. (2) The number of training examples in Domain A. (3) The choice of Domain A. (4) Network architectures. The ablation experiments are conducted on the S → M task and the network architecture is DTN if not specified.

6.1 Number of Training Examples in Domain B

We first show the impact of varying n_B on the transfer performance and model robustness in Fig. 6. The Scratch models and FT models

(a) Transfer performance (b) Under white-box attacks

Figure 6: The effect of n_B. As more labeled data are available in `Domain B`, both the transfer performance and the model robustness are improved. We report the adversarial accuracy of the white-box attacks with $\epsilon = 16/255$ in (b).



(a) Transfer performance (b) Under white-box attacks

Figure 7: The effect of n_A. The `FT` models consistently outstrip the `Scratch` model. When n_A increases, both the transfer performance and the model robustness are improved.

are trained with 50, 100, 200, 500 labeled samples in `Domain B`. We report the adversarial accuracy when $\epsilon = 16/255$. The classification accuracy and the adversarial accuracy of both models increase as more labeled data are available. The transfer performance and the model robustness of the `FT` model remain stable given multiple values of n_B while those of the `Scratch` model significantly drops as n_B decreases. For example, when there are only 50 examples in `Domain B`, the classification accuracy and the adversarial accuracy of the `Scratch` model is 91.59% and 71.45%, respectively, which lags behind those of the `FT` model by 4.99% and 14.04%. The results show that fine-tuning can be particularly beneficial when there are very few labeled samples in `Domain B`.

6.2 Number of Training Examples in `Domain A`

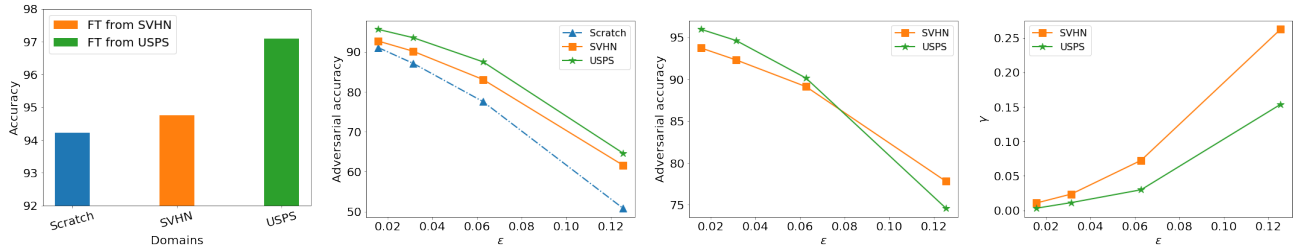
We vary the number of training examples in `Domain A` and report the classification accuracy and adversarial accuracy in Fig. 7. `Model A` is trained with $1K$, $5K$, $10K$, $100K$ and $500K$ examples. Similar to the results in Section 6.1, more labeled data in `Domain A` yield better transfer performance and improved robustness. The `FT` models obtained with different n_A values consistently outperforms the `Scratch` model.

6.3 Choice of `Domain A`

To study the effect of different source domains, we fix `M` as the `Domain B` and use `S` and `U` as `Domain A`, respectively. The number of training samples in `Domain A` and `Domain B` are 5,000 and 600, respectively. Thus we have two transfer tasks, `S` → `M` and `U` → `M`. Both `M` and `U` are handwritten digits and hence they are more visually similar than the other task `S` → `M`. The results are shown in Fig. 8. With the same amount of labeled data, when `Domain A` is more similar to `Domain B`, the classification accuracy is higher and `Model B` is more robust under white-box attacks. At the same time, `Model B` is more vulnerable to adversarial examples transferred from `Model A`.

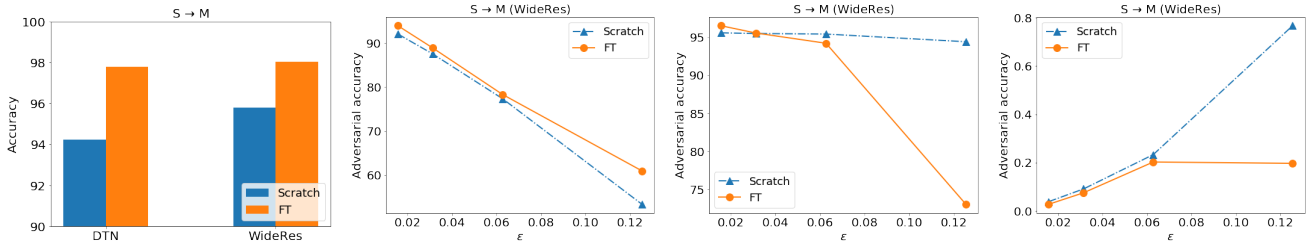
6.4 Network Architectures

We examine whether the observations generalize to other network architectures. We repeat the experiment with the `WideRes` network and the results are shown in Fig. 9. `WideRes` networks use widened residual blocks that improve both model performance and training efficiency. There are 28 convolutional layers in a `WideRes` network while there are only 5 layers in a `DTN` network, and the `WideRes` network has more representational power. This is demonstrated by the fact that the classification accuracy is improved for both the `Scratch` and `FT` model (Fig. 9a). Moreover, the adversarial accuracy of the `FT` model consistently outperforms that of the `Scratch` model disregard of the network architecture that is used. In terms of the robustness under black-box attacks, the conclusions are the same as those drawn when the `DTN` architecture is used: the `FT` model



(a) Transfer performance (b) Under white-box attacks (c) Under black-box attacks (d) Transferability γ

Figure 8: The effect of the choice of `Domain A` on transfer performance and robustness: `S` → `M` vs. `U` → `M`. `U` is more similar to `M` compared to `S`. Fine-tuning from a more visually similar domain yields more performance gain under white-box attacks, but the adversarial examples are more transferable if they are produced by the source model that is trained on a more similar domain.



(a) Transfer performance (b) Under white-box attacks (c) Under black-box attacks (d) Transferability γ

Figure 9: The effect of network architectures on transfer performance and robustness. The results of `WideRes` are consistent with those of the `DTN` architecture. The `FT` model is more advantageous than the `Scratch` model when the `WideRes` architecture is adopted under white-box attacks. On the other hand, the `FT` model is more likely to be attacked by the adversarial examples produced by its source model than the `Scratch` model under black-box attacks.

is again more likely to be attacked by the adversarial examples produced by its source model than the `Scratch` model.

7 DISCUSSION

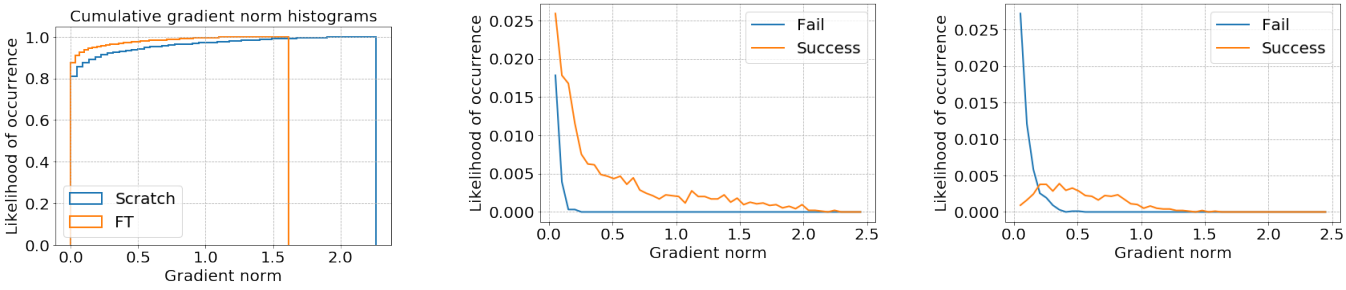
As demonstrated by the experimental results and ablation studies, fine-tuning can effectively improve both transfer performance and robustness under white-box attacks. On the other hand, we can successfully attack a fine-tuned target model in a restricted black-box manner by utilizing its source model, which is a downside of fine-tuning. The observations generalize across different transfer tasks and network architectures. We hypothesize that the success of fine-tuning can be attributed to the following two reasons.

• The improvements may benefit from more training data. Our empirical discoveries in Sections 6.1 and 6.2 show that increasing the number of training samples in either `Domain A` or `Domain B` yields enhanced robustness. They are in accordance with the theoretical results in [19], which postulate that training a robust classifier requires more data.

• Fine-tuning improves the Lipschitzness of the loss landscape and hence makes the model more robust. We visualize the histograms of the gradient norms $\|\nabla_{\mathbf{x}}\ell(y, f(\mathbf{x}))\|_2$ in Fig. 10. Fig. 10a shows that the gradient norms of the `FT` model is more likely to have a small value while the maximum value of the gradient norms of the `Scratch` model can be larger than 2. The histograms of the gradient norms of the `Scratch` and `FT` models are shown in Figs. 10b and 10c, respectively. The adversarial examples that successfully fool the target model are more likely to have large gradient norm values. The gradient norms of the `FT` model is suppressed, which might improve model robustness.

8 CONCLUSION

Though fine-tuning is a successful and popular transfer learning technique, its effect on model robustness has been almost ignored. To figure out this problem, extensive experiments are conducted in this paper. The results show that fine-tuning can enhance model robustness under white-box FGSM attacks. We also propose a simple and effective black-box attack method for transfer learning models. Results suggest that fine-tuning might introduce potential risks since a fine-tuned model is more likely to be successfully attacked by the adversarial examples crafted from its source model than a model that is learned from scratch. Our study convinces another advantage of fine-tuning and reveals that there are underlying risks that have been overlooked. We hope that the findings can serve as step stones towards transfer learning models that are both robust and effective. In addition, we also developed a new evaluation metric to measure how transferable are the produced adversarial



(a) Cumulative histograms of gradient norms

(b) Histogram of gradient norms of the Scratch model

(c) Histogram of gradient norms of the FT model

Figure 10: The histograms of the gradient norms. The FT model is more likely to produce small gradient norms, which indicates an improved Lipschitzness of the loss landscape and hence makes the model more robust. (For better viewing experience, the gradient norm starts from 0.05 in (b) and (c).)

examples to attack transfer learning models. We also believe this metric will be useful for future study of the vulnerability of transfer learning models.

ACKNOWLEDGMENTS

This paper was supported by the Early Career Scheme (ECS, No. 26206717), the Research Impact Fund (RIF, No. R6020-19), the General Research Fund (GRF, No. 16209715 and No. 16244616) from the Research Grants Council (RGC) of Hong Kong, China, and national project 2018AAA0101100.

REFERENCES

[1] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 15–26.

[2] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. 2017. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819* (2017).

[3] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 215–223.

[4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 580–587.

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. (2015). http://arxiv.org/abs/1412.6572

[7] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *International Conference on Machine Learning*. 2712–2721.

[8] Jonathan J. Hull. 1994. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 5 (1994), 550–554.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[12] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into transferable adversarial examples and black-box attacks. (2017). https://openreview.net/forum?id=Sys6GJqxl

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. (2018). https://openreview.net/forum?id=rJzIBfZAb

[14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[15] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1717–1724.

[16] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[19] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*. 5014–5026.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. (2014). http://arxiv.org/abs/1312.6199

[21] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 4534–4542.

[22] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1494–1504.

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 3156–3164.

[24] Karl Weiss, Taghi M Khoshgoftaar, and Dingding Wang. 2016. A survey of transfer learning. *Journal of Big Data* 3, 1 (2016), 1–40.

[25] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems*. 3320–3328.

[26] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2805–2824.

[27] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).