Adversarial Robustness for Unsupervised Domain Adaptation

Muhammad Awais [1, 2*], Fengwei Zhou [1], Hang Xu [1], Lanqing Hong [1],
Ping Luo [3], Sung-Ho Bae [2†], Zhenguo Li [1]

[1]Huawei Noah's Ark Lab
[2]Dept. of Computer Science, Kyung-Hee University, South Korea
[3]Dept. of Computer Science, The University of Hong Kong
awais@khu.ac.kr, {zhoufengwei, xu.hang, honglanqing}@huawei.com, pluo@cs.hku.hk,
shbae@khu.ac.kr, li.zhenguo@huawei.com

Abstract

Extensive Unsupervised Domain Adaptation (UDA) studies have shown great success in practice by learning transferable representations across a labeled source domain and an unlabeled target domain with deep models. However, current work focuses on improving the generalization ability of UDA models on clean examples without considering the adversarial robustness, which is crucial in real-world applications. Conventional adversarial training methods are not suitable for the adversarial robustness on the unlabeled target domain of UDA since they train models with adversarial examples generated by the supervised loss function. In this work, we propose to leverage intermediate representations learned by robust ImageNet models to improve the robustness of UDA models. Our method works by aligning the features of the UDA model with the robust features learned by ImageNet pre-trained models along with domain adaptation training. It utilizes both labeled and unlabeled domains and instills robustness without any adversarial intervention or label requirement during domain adaptation training. Our experimental results show that our method significantly improves adversarial robustness compared to the baseline while keeping clean accuracy on various UDA benchmarks.

1. Introduction

Transferring knowledge from a labeled source domain to an unlabeled target domain is desirable in many real-world applications. However, deep learning models do not perform well in the presence of such domain shifts. For example, a model trained on synthetic data may fail to generalize well on real-world data. Unsupervised Domain Adaptation (UDA) seeks to solve this problem by learning domain-invariant features. Recent UDA methods harness transferable features learned by deep models pre-trained on large datasets like ImageNet [12, 17, 29, 28, 49, 26, 40, 15, 21, 22]. However, a large body of work has shown that these deep models are vulnerable to small adversarial changes in the input that can easily fool the trained models [5, 39, 14, 7]. The widespread use of these models in sensitive applications requires them to be robust against these changes.

Significant attention has been devoted to counter adversarial examples, and many defense methods have been devised [14, 16, 42, 30, 6, 25, 33, 37, 41, 46]. Supervised adversarial training is among the most successful approaches [30]. It is based on the simple idea of training a model on adversarial examples. It utilizes min-max optimization where adversarial examples are first generated by iterative maximization of the loss, and the model is then trained on these examples. However, the generation of these adversarial examples requires labels and adversarial training implicitly assumes inputs from a single domain. These issues limit the applicability of adversarial training in UDA.

In this paper, we propose a simple, unsupervised, and domain agnostic method for robustness in UDA. Our method does not require labels and utilizes data from both domains, making it feasible for UDA. Our work is motivated by the recent line of work on transferability of robustness [13, 9], and observation that adversarially trained models learn "fundamentally different" features from normally trained counterparts [43, 20, 36]. The first line of work has demonstrated the transferability of adversarial robustness from a pre-trained robust model. The authors in [18, 38] show that adversarially pre-trained models can improve robustness in transfer learning; [13] shows that adversarial robustness can be distilled by matching softened labels produced by robust pre-trained

*This work was done while being at Huawei Noah's Ark Lab. The webpage for the project: awaisrauf.github.io/robust_uda
†Corresponding Author

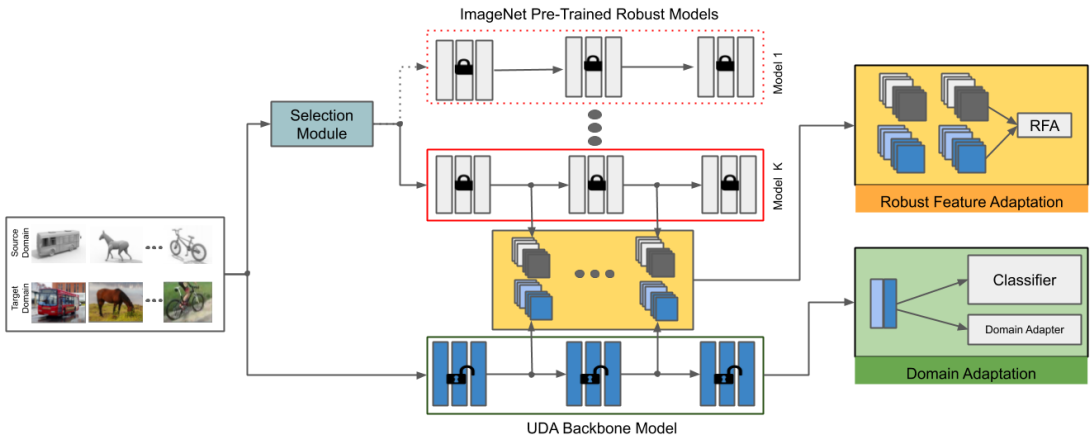

Figure 1. An overview of the proposed method. Source and target images are passed through the backbone model and robust teachers to get features at different blocks. The intermediate features are transferred to the robust feature adaptation (RFA) module, which adapts the robustness. The output of the backbone model goes through the domain adaptation module, which utilizes an unsupervised domain adaption algorithm. The parameters of the UDA feature extractor are updated to minimize both domain adaptation and robust feature adaptation loss. Light colors show the features extracted for source domain inputs and dark colors for target domain inputs.

models; [9] shows that robustness can be distilled by matching input gradients of robust models to those of a non-robust model. These works focus on cutting the computational cost of adversarial training for single domain classification and require labeled data.

Our proposed method, Robust Feature Adaptation (RFA), embeds the adaptation of robustness in the domain adaptation training by leveraging the feature space of robust pre-trained models. RFA uses ImageNet adversarially pre-trained models to extract robust features for inputs of source and target domains. It then instills robustness in UDA's feature extractor by minimizing its discrepancy with robust features. RFA enables the model to learn both domain invariant and robust features.

Unlike previous works on transferability, our method does not require labeled data as it only uses intermediate features of the robust models and a label-free distance measure between the feature spaces of the two models. Similarly, RFA does not require any adversarial intervention during the domain adaptation training as it does not generate adversarial examples. These characteristics make it possible to harnesses both labeled source and unlabeled target domains. Moreover, the RFA is a plug-in method that can be used with any UDA method to enhance its robustness. It only requires adversarially pre-trained models similar to the UDA methods that need normally pre-trained models. Our experiments show that RFA can equip UDA models with high adversarial robustness while keeping good generalization ability. Our contributions can be summarized as follows:

- We propose a plug-in method that aligns the features of a UDA model with the robust features of multiple adversarially pre-trained ImageNet models. This way, it instills robustness in UDA models without adversarial intervention or label requirement.
- To the best of our knowledge, we are the first to show that the adversarial robustness for a target task can be distilled from intermediate representations of robust models adversarially trained on a different task without any fine-tuning.
- Comprehensive experimental results show that our method consistently improves the robustness of various UDA algorithms on widely-used benchmark datasets. For instance, it improves the adversarial robustness from 0% to 43.49% while maintaining the clean accuracy for CDAN as the UDA algorithm on challenging simulation-to-real adaptation task of the VisDA-2017 dataset.

2. Related Work

Unsupervised Domain Adaptation. Most of the unsupervised domain adaptation methods are motivated by the theoretical results in [4, 3]. These results suggested learning representations invariant across domains. In deep learning, this is often achieved by min-max training where a pre-trained deep neural network is fine-tuned such that not only does it minimize the loss on labeled data from the source domain but also fool a discriminator. This discriminator is simultaneously trained to distinguish between source and target domains [12]. In recent works, it has also been shown that large models, pre-trained on large-scale datasets such as ImageNet, improve unsupervised domain adaptation [27, 12, 17, 29, 28, 49, 26, 40, 15, 21]. Several unsupervised domain adaptation algorithms have been proposed that leverage pre-trained models [27, 28, 49, 26]. However,

Dataset	Robust PT	Source-only	DANN [12]	DAN [27]	CDAN [28]	JAN [29]	MDD [49]
VisDA-17	×	43.05 / 0	71.34 / 0	61.79 / 0.01	74.23 / 0	63.70 / 0	72.20 / 4.03
	✓	25.67 / 6.64	65.79 / 38.21	42.24 / 22.11	68.00 / 41.67	55.08 / 32.15	67.72 / 39.50
Office-31	×	77.80 / 0.02	85.79 / 0	81.72 / 0	86.90 / 0	85.68 / 0	88.31 / 1.70
	✓	69.51 / 41.11	77.30 / 62.38	73.71 / 42.29	79.67 / 65.53	75.12 / 60.24	80.72 / 67.54
Office-Home	×	58.29 / 0.06	63.39 / 0.05	59.64 / 0.23	67.03 / 0.04	64.61 / 0.07	67.91 / 5.81
	✓	53.89 / 31.46	58.10 / 37.25	55.18 / 24.21	63.04 / 43.81	60.74 / 33.09	63.30 / 43.42

×: Normally Pre-Trained Model, ✓: Adversarially Pre-Trained Model, PT: Pre-Training.

Table 1. Can Robust Pre-Training (PT) instill robustness in unsupervised domain adaptation setting? Comparison between normally and adversarially pre-trained models for clean accuracy / adversarial robustness (%) with six UDA algorithms. Adversarial pre-training improves adversarial robustness but also causes a drop in clean accuracy.

these works do not consider robustness. Our work is complementary to these works as it improves the robustness of these methods.

Adversarial Training and Robust Features Adversarial attacks are considered security risk [5, 39, 14, 7]. Numerous methods have been proposed to defend against such examples [16, 42, 30, 6, 25, 33, 37, 41, 46, 1]. Adversarial training – the most effective defense mechanism – is devised to defend against ℓ_p bounded adversarial perturbations [14, 30] in the inputs. However, adversarial training requires labels and therefore is not suitable for UDA training. In another direction, recent work has also shown that adversarially trained models learn 'fundamentally different" representations [43, 20, 11]. Our work is motivated by this observation, and we proposed an algorithm to leverage these robust features.

Knowledge and Robustness Transfer The main purpose of knowledge distillation is to decrease the size of a large model. It works by distilling the knowledge of a big pre-trained teacher model to a compact *randomly initialized* student model for the same dataset [19]. Many different settings have been explored to achieve this objective [32, 47, 48, 44]. Our work is different from these works as we want to only adapt robustness from the teacher without labels while also learning domain invariant features that perform well on two domains.

Our work is motivated by [13, 9, 18, 38] that showed transferability of robustness. However, these methods are primarily motivated to decrease the computational cost of adversarial training and require labels. In [13], the authors showed that the robustness can be distilled from a large pre-trained model (e.g., ResNet) to a compact model (e.g., MobileNet) by utilizing soft class scores produced by the teacher model. Compared to the work in [13], our method distills robustness from the intermediate representations only. Furthermore, the distillation is performed from teachers trained on one task (i.e., supervised classification) to a student needed to be trained on another task (i.e., unsupervised domain adaptation), which has not been explored previously. In [9], the distillation is performed by matching the gradient of the teacher and student. This method requires fine-tuning on target tasks, back-propagation to get gradients, and discriminator-based learning. Compared to [9], our proposed method does not require any fine-tuning, and it adapts robust features from pre-trained models without requiring any extra back-propagation. Moreover, both of these distillation methods require labels and are designed for single-domain training.

3. Preliminaries

Unsupervised Domain Adaptation aims to improve generalization on target domain by reducing domain discrepancy between source and target. Formally, we are given labelled data in the source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s} \sim P$ and unlabeled data in the target domain $D_t = \{x_j^t\}_{j=1}^{n_t} \sim Q$, where $P \neq Q$. Most unsupervised domain adaptation methods fine-tune a pre-trained backbone model $f(x;\theta)$ and train a classifier $C(f(x;\theta);\psi)$ on top of it. The training is done in such a way that it reduces error on the labeled source domain as well as learning features that are invariant in both source and target domains.

Adversarial examples [39, 14] are bounded and imperceptible perturbations in the input images that change the normal behavior of neural networks. Thus, the adversarial robustness of a model is its invariance to such small ℓ_p bounded perturbation in the input. To achieve this robustness, adversarial examples are created by maximizing the loss, and then it is minimized to train the model [30]:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}\left[\max_{||\delta||_p \leq \epsilon} \mathcal{L}(x+\delta, y; \theta)\right],$$

where ϵ is the perturbation budget that governs the adversarial robustness of the model. The model is trained to be robust in ℓ_p-norm ball of radius ϵ. Increasing ϵ means the model is stable for a larger radius. However, this framework is not appropriate for UDA as this requires labels and assumes data from a single domain.

Following [30], we define the **adversarial robustness** as the accuracy of target dataset (D_t) perturbed with a perturbation budget of ϵ in ℓ_∞-norm ball. To find the adversarial example x_{adv}, we use Projected Gradient Descent (PGD) with 20 iterations [30]. *We have used term robustness and adversarial robustness interchangeably.*

4. Pre-Training and Robustness

We start with a simple question: can we instill robustness in unsupervised domain adaptation by replacing the normally pre-trained feature extractor with a robust counterpart.

To answer this question, we replaced the normal backbone model with an adversarially trained. We call this setup **Robust Pre-Training** or Robust PT To demonstrate the effect of robust pre-training, we conducted a set of experiments with six UDA methods and three common datasets, i.e., Office-31 [34], Office-Home [45]and VisDA-2017 [31]. We employed a ResNet-50 [17] adversarially trained with different perturbation budgets as defined in Section 3. Unless stated otherwise, robustness is reported with PGD-20 and perturbation budget of $\epsilon = 3$. For a fair comparison, we use the default settings of all the hyper-parameters and report the average results over three independent runs. We only reported the best results averaged over all possible tasks of each dataset here. For detailed results, please refer to the supplementary material.

It is reasonable to expect that adversarial pre-training will not increase robustness for unsupervised domain adaptation. Previous work has shown that the transferability of robustness is due to the robust feature representations learned by the pre-trained models. Robustness is only preserved if we do not update the backbone [18, 38]. Specifically, to maintain the robustness, only an affine layer is trained on top of the fixed feature extractor with the help of the labeled data. However, we fine-tuned the backbone model to be accurate in the source domain and invariant for the source and target domains.

The best robustness results averaged over all tasks in each dataset are shown in Table 1. We find that an adversarially pre-trained backbone can improve the robustness under UDA settings.For example, robustness for CDAN [28] improves from 0% to 41.67%, with around 5.5% decrease in clean accuracy on VisDA-2017 dataset. For the DAN algorithm, improvement in robustness is 0% to 22.11% at the cost of an 18% drop in clean accuracy. Similar improvement in robustness is also visible in experiments involving Office-31 and Office-Home datasets as shown in Table 1.

However, adversarially pre-trained backbone decreases the generalization ability of models for the UDA setting. The decrease in accuracy can go as high as 20%. We hypothesize that robust pre-training is not the most efficient way of leveraging robust features of the backbone. In the next section, we design an algorithm to utilize these features more efficiently.

5. Robust Feature Adaptation

In this section, we introduce our method and its motivation. The goal of Robust Feature Adaptation (RFA) is to improve the adversarial robustness of unsupervised domain adaptation (UDA) algorithms without causing a significant drop in accuracy. Based on our experiments in the previous section, we hypothesized that the direct use of pre-trained models as backbone model is not an efficient way to instill robustness in UDA training. These pre-trained models have significantly less accuracy to begin-with [10]. This low pre-training accuracy makes it hard for UDA training to get better generalizations for the task. Our hypothesis is based on previous observations [23] that have shown a direct relationship between the accuracy of a pre-trained model and its final performance on a given task.

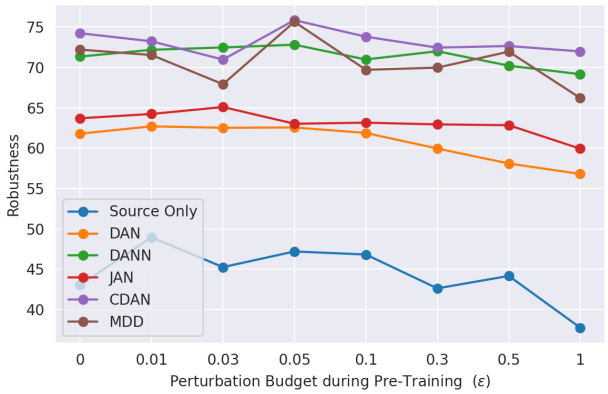

Figure 2. The clean accuracy of weak adversarially pre-trained (adversarial pre-training with small ϵ) models on VisDA-2017 dataset.

In our method, we propose to adopt robust features instead of directly using robust models as a backbone. The main idea of the proposed method is to align the features of the UDA backbone model with the robust features provided by multiple adversarially pre-trained models. This aligning is done as we do domain adaptation training for learning domain invariant features.

Each part of our framework is based on a hypothesis based on insights from previous works and detailed experimental investigation. In this section, we describe each component of our proposed algorithm along with their motivation. The empirical comparisons to support our method are given in Section 7.1. An overview of the proposed method is illustrated in Figure 1.

5.1. Feature Extractor for Domain Adaptation

As described earlier, existing UDA algorithms fine-tune normally pre-trained ImageNet models. However, adversarially pre-trained models learn 'fundamentally different features' compared to their normally pre-trained counterparts [43, 11, 20]. This difference can cause inconsistency between the features of student and teacher models which may cause difficulty in optimization. Hence, we propose to use a weak adversarially pre-trained model (model pre-trained with a small perturbation budget) as the backbone model.

As shown in Figure 2, these robust models do not hurt clean

Dataset	Method	Accuracy	Robustness
Office-31	Baseline	88.31	1.70
	Robust PT	80.72	67.54
	RFA	84.21	74.31
VisDA-2017	Baseline	72.20	4.03
	Robust PT	67.72	39.50
	RFA	72.90	47.66
Office-Home	Baseline	67.91	5.81
	Robust PT	63.30	43.42
	RFA	65.37	51.13

Table 2. Comparison of robustness and clean accuracy for RFA with Robust Pre-Training and baseline. RFA improves robustness compare to Robust Pre-Training while keeping good generalization.

UDA Method	Baseline	Robust PT	RFA
Source Only	43.05 / 0	25.67 / 6.64	44.65 / 11.10
DANN	71.34 / 0	65.79 / 38.21	65.32 / 34.11
DAN	61.79 / 0	42.24 / 22.11	55.70 / 21.59
CDAN	74.23 / 0	68.00 / 41.67	72.03 / 43.49
JAN	63.70 / 0	55.08 / 32.15	62.95 / 32.81

Table 3. Comparison of Robust Pre-Training and RFA for five UDA algorithms with the VisDA-2017 dataset. RFA significantly improves robustness while keeping good clean accuracy.

accuracy significantly but can solve the feature inconsistency problem. A experimental comparison is shown in Section 7.1.

5.2. Joint Training for Adaption of Robust and Domain Invariant Features

Our robust feature adaptation method aims to fine-tune the UDA feature extractor in such a way that it adapts robust features from adversarially trained models along with domain-invaraint features from UDA training.

In knowledge distillation, we initialize the student with random weights and force the student to mimic the feature space of the teacher by minimizing the pair-wise distance between features and/or softened class scores. Our UDA feature extractor, on the other hand, is also pre-trained and has already learned a set of features. This means that the student and the teacher may have learned features in different ways or the order of the learned feature maps may differ. Furthermore, *since the teacher is not trained directly on the target dataset, it can not provide the softened class scores.* This is also another reason not to directly minimize pair-wise distance as the teacher is trained on a different dataset. In conclusion, we only want to use the feature supervision of the teacher to align student's features with it to adapt robustness.

To align features of student to that of robust teacher, we used similarity preserving loss to match the similarity of activations between robust and non-robust features [44]. The main idea of this loss is to align the student's feature in such a way that two inputs producing similar activations in the feature space of teacher model should also produce similar activations in the feature space of student model. Specifically, given a mini-batch of training data, let $Q_T^l \in \mathbb{R}^{b\times d}$ and $Q_S^l \in \mathbb{R}^{b\times d}$ denote the activations of l-th layer from teacher and student models, respectively, where b is the batch size and d is the dimension of the activations after reshaping. The similarity matrices of l-th layer from teacher and student models are defined as $G_T^l = Q_T^l \cdot Q_T^{l\intercal}/||Q_T^l \cdot Q_T^{l\intercal}||_2$ and $G_S^l = Q_S^l \cdot Q_S^{l\intercal}/||Q_S^l \cdot Q_S^{l\intercal}||_2$, respectively, where $||\cdot||_2$ is a row-wise L2 normalization. We then define the robust feature adaptation loss of l-th layer as

$$\mathcal{L}_{RFA}^l = \frac{1}{b^2}||G_T^l - G_S^l||_F^2,$$

where $||\cdot||_F$ is the Frobenius norm.

We use the sum of robust feature adaptation losses of intermediate layers:

$$\mathcal{L}_{RFA} = \sum_{l=1}^{L} \mathcal{L}_{RFA}^l,$$

where L is the number of intermediate layers. The joint training loss is then defined as

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_{DA} + \alpha\mathcal{L}_{RFA},$$

where $\mathcal{L}_C$ is the classification loss on source domain, $\mathcal{L}_{DA}$ is the loss term for domain adaptation and α is a hyper-parameter that balances domain adaptation and robust feature adaptation. Note that our proposed method can be applied to different UDA algorithms by using the corresponding domain adaptation method with loss term $\mathcal{L}_{DA}$.

5.3. Adapting Diverse Robust Features

The Figure 4 shows the diversity of discriminative features learned by the same model trained with different perturbation budgets. More details are in Section 7.1. To leverage these diverse robust features, we propose to supervise the student with multiple teachers. To reduce the computing cost during training, we randomly choose one teacher at each iteration during training. This means that we can guide the student model with the diversity of multiple teachers with the same computing cost as using one.

6. Experiments

6.1. Setup

We conduct experiments on 19 different tasks derived from 3 main-stream unsupervised domain adaption (UDA) datasets. **Office-31** [34] is a standard domain adaptation dataset with 6 tasks based on three domains: Amazon (**A**), Webcam (**W**) and DSLR (**D**). The dataset is imbalanced across domains

with 2,817 images in **A**, 795 images in **W** and 498 images in **D** domain. **Office-Home** [45] is a more complex dataset compared to Office-31 and contains more images (15,500) for 12 adaptation tasks based on 4 more diverse domains: Artistic (**Ar**), Clip Art (**Cl**), Product (**Pr**), and Real World (**Rw**). **VisDA-2017** [31] is a simulation-to-real dataset with two extremely different domains: synthetic domain in which images are collected from 3D rendering models and real-world images. It is also a large-scale dataset as it contains 280k images in the synthetic domain and 50k images in the real-world domain. Due to the extremely different domains and scale, it is one of the most challenging datasets in UDA.

Unless stated otherwise, we use ResNet-50 [17] as our backbone model and MDD [49] as the domain adaptation algorithm. We used this setup to show that our method can improve robustness without a significant drop in accuracy. To show that Robust Feature Adaptation (RFA) can work as a plug-in method, we conduct experiments with six UDA algorithms: Source Only (fine-tuning model on source data only), DAN [27], DANN [12], JAN [29], CDAN [28], and MDD [49]. We follow the experimental protocol of [12, 28] commonly used in UDA and adopt the hyper-parameters used in [22]. We compare RFA with **UDA algorithm Baseline** (adopting normally pre-trained ImageNet model) and **Robust PT** (UDA algorithm adopting adversarially pre-trained ImageNet model). For a fair comparison, we use the same values for all hyper-parameters for the UDA algorithm Baseline, Robust PT, and RFA. The new hyper-parameter of our proposed method is α. We choose it based on the magnitude of domain adaptation loss. Specifically, we multiply robust feature adaptation loss $\mathcal{L}_{RFA}$ by 1000 to make it have the equivalent magnitude to that of domain adaptation loss. We report average results over three runs for all the experiments.

6.2. Main Results

On Improving Robustness. To achieve better robustness, we choose four strong teachers, i.e., ImageNet ResNet-50 models, trained with different perturbation budgets. More specifically, we use perturbation budget of $\epsilon \in \{3, 5\}$ with ℓ_2-norm and $\epsilon \in \{2, 4\}$ with ℓ_∞-norm. To show the effectiveness of our method, we choose a backbone adversarially trained with $\epsilon = 1$. For the bulk of our experiments, We use MDD as a domain adaptation algorithm.

The average results for Office-31, Office-Home, and VisDa-2017 are shown in Table 2. These results clearly show that our method can improve the robustness of the backbone model by adapting the robust features without a significant drop in clean accuracy. The improvement in robustness is due to the robust teachers while the improvement in clean accuracy is because of the backbone model used in RFA. This model has higher accuracy compared to backbone use in Robust Pre-Training. This way, our method has a significant advantage over Robust PT as it can use backbone models with higher clean accuracy while adapting robustness from any teacher.

α	100	500	1000	5000
Acc.	71.61	73.62	72.90	70.31
Rob.	40.07	46.36	47.66	47.27

(a)

Teachers	Acc.	Rob.
Single	70.31	40.15
Multiple	**73.45**	**40.87**

(b)

Table 4. **Ablation Studies.** (a) The effect of varying α on accuracy and robustness (%) for RFA on VisDA-2017 dataset. (b) The effect of multiple teachers on accuracy and robustness (%) on VisDA-2017 dataset.

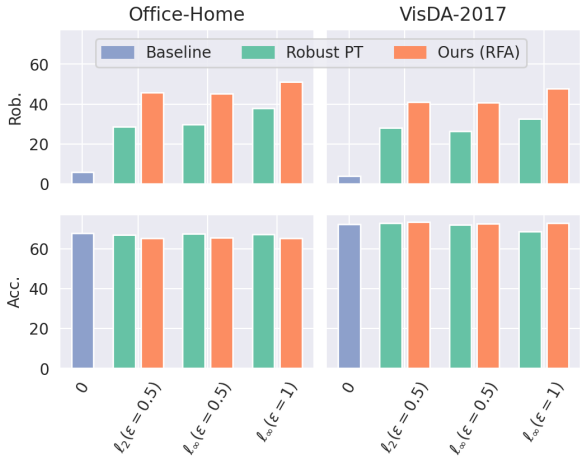

Figure 3. Comparison of MDD Baseline, Robust PT (Pre-Training), and RFA for average robustness and accuracy (%) on Office-Home and VisDA-2017. The x-axis shows the perturbation budget of the pre-trained model.

On RFA as a Plug-in Method. A salient characteristic of our method is that it can complement existing or new domain adaption algorithms that use ImageNet pre-trained models. To show this, we conduct experiments with six different UDA algorithms (Source only, DAN, DANN, JAN, CDAN, and MDD) on the challenging and large-scale VisDA-2017 dataset. As shown in Table 3, RFA improves robustness for all the six UDA algorithms.

7. Discussion and Analysis

7.1. Empirical Investigation of Design Principles for Our Framework

Choosing Student Model. One major insight of our framework is the use of weak adversarially pre-trained models (adversarially pre-trained models with small perturbation budget ϵ) as feature extractors. To see the effect of the weak adversarially pre-trained model, we compare it with a normally pre-trained student in Table 5(a). Normally pre-trained student can improve robustness albeit not significantly. Weak adversarially pre-trained students, on the other hand, can improve robustness significantly.

To further see how the UDA feature extractor model should

Student	Acc.	Rob.
Baseline	72.20	4.03
Normal	71.22	7.63
Adv.	72.71	40.61

(a)

Loss	DANN	CDAN	MDD
L1	45.02 / 9.58	55.16 / 13.53	54.52 / 18.89
L2	54.28 / 1.45	58.16 / 1.76	64.20 / 8.29
SP	65.32 / 34.11	72.03 / 43.49	72.90 / 47.66

(b)

Method	RN-18	WRN-50-2
Baseline	69.61 / 0.15	73.36 / 5.47
Robust PT	64.44 / 24.40	71.20 / 37.63
Ours (RFA)	65.05 / 36.46	74.98 / 50.47

(c)

Table 5. **Ablation Studies.** (a) Effect of Normal student robustness of six UDA algorithms. (b) Effect of minimizing pairwise loss compare to similarity preserving for robustness on VisDA-2017. (c) Comparison of accuracy / robustness (%) for MDD Baseline, Robust PT and RFA with different neural network architectures on VisDA-2017. RFA consistently improves robustness for different architectures. Here RN represents ResNet and WRN WideResNet.

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Avg
Baseline	54.59	72.38	77.19	61.52	71.19	71.54	63.04	50.31	79.0	72.5	57.66	83.92	67.91
Robust PT	55.07	73.87	78.26	60.82	71.84	71.88	60.65	51.89	79.02	**72.64**	**60.50**	82.81	68.27
Ours (RFA)	**55.65**	**77.13**	**80.69**	**64.43**	**74.81**	**75.54**	**63.99**	**53.07**	**80.59**	71.80	58.41	**84.31**	**70.03**

Table 6. Classification accuracy (%) for all the twelve tasks from Office-Home dataset based on ResNet-50. Our method improves clean accuracy of 10 out of 12 tasks as well as the average.

be pre-trained, we compare the robustness and accuracy of different feature extractor models with different pre-training perturbation levels in Figure 3.

Comparison of Pairwise and with Non-Pairwise Loss. An important aspect of our algorithm is the loss function. We hypothesized that similarity preserving loss that preserves similarity between the activations is better to compare to pair-wise loss. This is because our student model is already trained and we only want to fine-tune it and require weak supervision. To illustrate it, we compare the robustness and clean accuracy for two pairwise losses with similarity preserving loss in Table 5(b).

Effect of Multiple Teachers. We hypothesized that the same model trained with different perturbation budgets can supervise student models with the diverse features. In Figure 4, we show the maximally activated neurons (maximum value across channels) of four different residual blocks of the robust ResNet-50 model. The first row shows activations of residual blocks for a normally pre-trained model and other rows represent activations for robust ResNet-50 models trained with different values of ϵ. The figure shows the diversity of discriminative features learned.

To illustrate the effect of multiple teachers, we compare it with single teacher in Table 4(b). Single model supervision is enough to distill the robustness. However, the diversity of supervision from multiple teachers improves both accuracy and robustness.

7.2. Ablation Studies

Sensitivity of Weight of Robust Feature Adaptation (α). We study the sensitivity of our method to the weight of robust feature adaptation term α on VisDA-2017. Table 4(a) demonstrates the clean accuracy and adversarial robustness by varying $\alpha \in \{0, 100, 500, 1000, 5000\}$. Increasing α decreases the clean accuracy while increasing the robustness. This shows that α can control the trade-off between clean accuracy and adversarial robustness.

Effect of number of PGD iterations on robustness. To further show the transferability of robustness, we test our method with an increasing number of iterations for PGD attack (PGD-k). The robustness of our method is consistent as shown in Table 7.

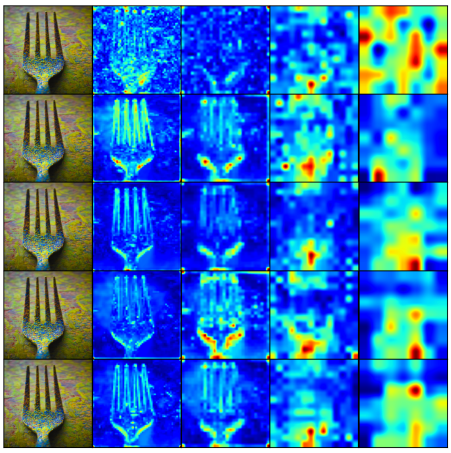

Figure 4. Maximally activated neurons for an image from Office-Home dataset. *The first row shows activations for normally pre-trained model* and other rows show activations for robust pre-trained models trained with different perturbation budget (ϵ). Highlighted regions can be interpreted as the discriminative parts of the input that activates the neurons the most. Note that different models have learned different discriminative features.

Method	Clean	FGSM	PGD-k			
			10	20	50	100
Baseline	72.20	41.15	11.82	4.03	3.24	3.06
Robust PT	71.95	63.23	39.54	28.21	25.55	24.69
Ours	**73.45**	**67.87**	**42.25**	**40.87**	**40.28**	**40.11**

Table 7. The effect of an increasing number of iterations for PGD attack. Results of the proposed method are consistent, showing a successful convergence of PGD attacks.

Improvement by RFA is consistent across architectures. In Table 5(c), we demonstrate that our proposed method can improve robustness using different architectures. RFA improves the robustness of Wide-ResNet-50-2 from 5.47%

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg.
Baseline	91.40	98.74	100.00	92.17	73.06	74.47	88.31
Robust PT	91.78	99.12	100.00	92.77	73.85	74.11	88.60
Ours (RFA)	**92.80**	**99.21**	**100.00**	**93.04**	**78.00**	**77.74**	**90.15**

(a)

Method	Source	DANN	DAN	CDAN	JAN	MDD
Baseline	43.05	71.34	61.79	74.23	63.70	72.20
Robust PT	47.20	72.81	62.56	75.85	63.02	75.64
Ours (RFA)	**59.00**	**75.05**	**65.58**	**77.54**	**66.68**	**79.42**

(b)

Table 8. **Improved Clean Accuracy.** (a) Classification accuracy (%) for all the six tasks from Office-31 dataset based on ResNet-50. (b) Comparison of classification accuracy (%) for Baseline, Robust PT and RFA with six UDA algorithms on VisDA-2017 dataset. RFA consistently improves accuracy for all UDA algorithms.

Method	Art-Painting	Cartoon	Sketch	Photo	Average
Baseline	77.93	80.29	78.90	94.55	82.92
	0	0.13	2.24	0.18	0.64
Ours (RFA)	76.56	76.83	75.97	94.61	81.00
	23.15	51.58	62.82	40.00	44.38

Table 9. Comparison of accuracy and robustness (%) for DecAug Baseline, Robust PT and RFA for all the four tasks from PACS based on ResNet-18.

Dataset	Rob. PT	Source Only	DANN [12]	DAN [27]	CDAN [28]	JAN [29]	MDD [49]
VisDA 2017	×	43.05	71.34	61.79	74.23	63.70	72.20
	✓	**48.95**	**72.81**	**62.70**	**75.85**	**65.51**	**75.64**
Office 31	×	**77.80**	85.79	81.72	86.90	85.68	88.31
	✓	77.66	**86.06**	**82.08**	**88.05**	**86.05**	**88.60**
Office Home	×	58.29	63.39	59.64	67.03	64.61	67.91
	✓	**58.87**	**64.08**	**60.38**	**67.67**	**65.60**	**68.27**

×: Normally Pre-Trained Model, ✓: Adversarially Pre-Trained Model, Rob. PT: Robust Pre-Training.

Table 10. Comparison between normally and adversarially pre-trained models on classification accuracy (%) with different UDA algorithms. Adversarial pre-training improves classification accuracy for UDA.

to 50.47% and accuracy of ResNet18 from 0.15% to 36.46%.

7.3. Can RFA Improve Robustness for Domain Generalization?

An important aspect of our method is that it is domain-agnostic and can be applied to tasks involving more than one domain. To illustrate this, we also conduct experiments for Domain Generalization (DG) with our method on PACS [24] dataset. DG methods [24, 8, 50, 2] learn models from multiple domains such that they can generalize well to unseen domains. PACS dataset contains four domains with different image styles: art painting, cartoon, sketch, and photo. We follow the same leave-one-domain-out validation experimental protocol as in [24]. For each time, we select three domains for training and the remaining domain for testing. We apply RFA to the SOTA DG method DecAug [2] and report results in Table 9. It illustrates that our method can also significantly improve the robustness while maintaining good clean accuracy in domain generalization.

7.4. Can Adversarially Pre-Trained Models Improve Clean Accuracy?

A recent work [35] has shown that weak adversarially pre-trained models (AT with small $\epsilon \in [0.01, 0.5]$) can also improve clean accuracy for target tasks in transfer learning, e.g., ImageNet to Pets dataset. In this section, we explore this hypothesis for unsupervised domain adaptation (UDA). Specifically, we did experiments for two settings: using weak adversarially pre-trained models as feature extractors and using them as teachers in our proposed algorithm.

First, we use a weak adversarially pre-trained model as a feature extractor while keeping everything else the same as in UDA training. We found that this simple setup can improve clean accuracy. The results are shown in Table 10. To further see the effect of robust features, we used these weak adversarially trained models in our robust adaptation algorithm. The results on different tasks from Office-31, Office-Home and average accuracy for different UDA algorithms on VisDA-17 are shown in Tables 8(a),6, 8(b), respectively. RFA outperforms both Baseline and Robust Pre-Training with significant margins. Our method achieves 90.15% compared to 88.31% of Baseline and 88.60% of Robust Pre-Training on Office-31. Similarly, on a more complex Office-Home dataset, it achieved 70.03% compared to 67.91% of Baseline and 68.27% of Robust PT. On challenging the VisDA-2017 dataset, we achieved even higher improvements. For instance, MDD with normally pre-trained ResNet-50 achieves an accuracy of 72.20%, but our proposed algorithm achieves 79.42% – an absolute 7% improvement. It is noteworthy that our method significantly *improves accuracy on hard tasks*, e.g., for Office-31, **D** → **A** (73.06% to 78%) and **W** → **A** (74.47% to 77.74%); for Office-Home, **Cl** → **Ar** (61.52% to 64.43%), **Cl** → **Pr** (71.19% to 74.81%) and **Cl** → **Rw** (71.54% to 75.54%); for VisDA-2017, simulation to real (72.20% to 79.42%). This highlights the importance of adaptation of robust features for UDA.

8. Conclusion

Existing interventions for adversarial robustness require labels and assume learning from a single domain. This hinders their application in unsupervised domain adaptation. To make unsupervised domain adaptation robust, we introduced a simple, unsupervised and domain-agnostic method that does not require adversarial examples during training. Our method is motivated by the transferability of robustness. It utilizes adversarially pre-trained models and adapts robustness from their internal representations. Our results show that it significantly improves the robustness for UDA.

Acknowledgements. Authors are thankful to the anonymous reviewers, Faaiz, Teerath, Salman and Asim for their help and constructive feedback.

References

[1] Muhammad Awais, Fahad Shamshad, and Sung-Ho Bae. Towards an adversarially robust normalization approach. *arXiv preprint arXiv:2006.11007*, 2020. 3

[2] Haoyue Bai, Rui Sun, Lanqing Hong, Fengwei Zhou, Nanyang Ye, Han-Jia Ye, S-H Gary Chan, and Zhenguo Li. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. *arXiv preprint arXiv:2012.09382*, 2020. 8

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 2

[4] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137–144, 2006. 2

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. 1, 3

[6] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. 1, 3

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1, 3

[8] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 8

[9] Alvin Chan, Yi Tay, and Yew-Soon Ong. What it thinks is important is important: Robustness transfers through input gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 332–341, 2020. 1, 2, 3

[10] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 4

[11] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019. 3, 4

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 2, 3, 6, 8

[13] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:3996–4003, 04 2020. 1, 3

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 3

[15] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020. 1, 2

[16] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 1, 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 4, 6

[18] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *Proceedings of the International Conference on Machine Learning*, 2019. 1, 3, 4

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[20] Andrew Ilyas, Shibani Santurkar, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 1, 3, 4

[21] Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 4816–4827. PMLR, 2020. 1, 2

[22] Mingsheng Long Junguang Jiang, Bo Fu. Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library, 2020. 1, 6

[23] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 4

[24] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 8

[25] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. 1, 3

[26] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022, 2019. 1, 2

[27] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2, 3, 6, 8

[28] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018. 1, 2, 3, 4, 6, 8

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 1, 2, 3, 6, 8

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 3

[31] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 4, 6

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[33] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 3

[34] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 4, 5

[35] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020. 8

[36] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*, pages 1262–1273, 2019. 1

[37] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019. 1, 3

[38] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. In *International Conference on Learning Representations*, 2019. 1, 3, 4

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 3

[40] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020. 1, 2

[41] Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *arXiv preprint arXiv:1904.13000*, 2019. 1, 3

[42] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 1, 3

[43] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1, 3, 4

[44] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019. 3, 5

[45] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 6

[46] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 1, 3

[47] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017. 3

[48] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[49] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I Jordan. Bridging theory and algorithm for domain adaptation. *arXiv preprint arXiv:1904.05801*, 2019. 1, 2, 3, 6, 8

[50] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. *arXiv:2007.03304*, 2020. 8