Unsupervised Robust Domain Adaptation without Source Data

Peshal Agarwal[1] Danda Pani Paudel[1] Jan-Nico Zaech[1] Luc Van Gool[1,2]
[1]Computer Vision Laboratory, ETH Zurich, Switzerland [2]KU Leuven, Belgium
agarwalp@student.ethz.ch {paudel, zaechj, vangool}@vision.ee.ethz.ch

Abstract

We study the problem of robust domain adaptation in the context of unavailable target labels and source data. The considered robustness is against adversarial perturbations. This paper aims at answering the question of finding the right strategy to make the target model robust and accurate in the setting of unsupervised domain adaptation without source data. The major findings of this paper are: (i) robust source models can be transferred robustly to the target; (ii) robust domain adaptation can greatly benefit from non-robust pseudo-labels and the pair-wise contrastive loss. The proposed method of using non-robust pseudo-labels performs surprisingly well on both clean and adversarial samples, for the task of image classification. We show a consistent performance improvement of over 10% *in accuracy against the tested baselines on four benchmark datasets. Our source code will be made publicly available.*

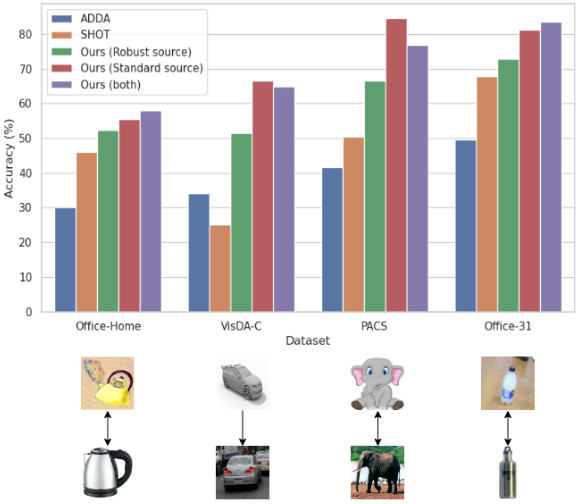


Figure 1: Test accuracy averaged over all domain adaptation tasks for multiple datasets. All our proposed methods show significant improvement over the baselines.

1. Introduction

Transferring the knowledge learned in one domain to another, in an unsupervised manner, is highly desired for a wide range of applications for learning-based methods [31, 46, 7, 47, 52]. Many of these applications also require models to be robust towards data perturbations [32, 33, 18]. In practice the source data may no longer be accessible during the knowledge transfer, due to privacy, storage or communication overhead; An example where such limitations are clearly manifested is image understanding with datasets[1] containing people's faces. These practical limitations call for methods that can adapt using only the target data. Furthermore, adaptation without the need of source data can also offer benefits in terms of computational cost and may simplify data handling.

This paper addresses a real world compound problem of (i) unsupervised domain adaptation; (ii) model robustness; and (iii) the lack of source data during transfer. All three mentioned issues are jointly considered, leading to a realistic yet a very challenging problem. Up to our knowledge, this problem is addressed for the first time in this work.

We consider that the source data is available only during the source model training, which is performed in a supervised manner. The model is then adapted to the target domain in an unsupervised manner, when the source data is no longer available. In this process, the robustness of the target model towards the adversarial perturbations is pursued. This paper studies several aspects of designing a robust domain adaptation method and proposes a simple yet novel technique to answer the key questions of:

- How to perform robust and unsupervised domain adaptation without source data?
- Can robust and standard models be combined to efficiently use information from the source domain?
- How do we perform robust domain adaptation, if only one model (robust or standard) is available?
- Is the best adaptation approach dataset dependent?

The problem of unsupervised domain adaptation with-

[1] Our work focuses on the image classification problem.

out source data has recently been studied in [23, 15, 54, 38, 22, 19, 17, 53]. However, the existing work does not take robustness into consideration. In this work, we first show that robust domain adaptation performs reasonably well within the aforementioned setup, when the method of [23] is directly applied. The method exploits the target's pseudo-labels, generated by the source model, for adaptation, which inevitably leads us to use robust pseudo-labels. We first study the performance of [23] under the adversarial perturbations for robustness, and then improve the performance by over 20% in accuracy consistently across four benchmark datasets, as shown in Fig. 1. Such improvement is achieved by exploiting non-robust pseudo-labels and the target's pair-wise contrastive learning scheme. The major finding of our work is that *robust domain adaptation can largely benefit from non-robust pseudo-labels and the pair-wise contrastive loss.* Our finding allows us to improve not only the robust accuracy, but also the clean accuracy in the target domains of a single robust model.

We study three different cases of model availability: (i) given only the standard source model; (ii) given only the robust source model; (iii) given both models. In the following, we will first present the case when both models are available. In this case, we wish to adapt the robust source model to the target while guarding the robustness. During the adaptation of the robust model, the labels generated by the standard one are used in three different ways: (i) cross-entropy loss; (ii) adversarial examples generation; (iii) contrastive feature learning. These three aspects of utility have shown to be complimentary to each other. Exploitation of non-robust pseudo-labels in this fashion also offer a significantly better performance compared to its robust counterpart. In fact, this observation leads us to suggest a new source data training and model sharing protocol. In the source domain, we suggest to train two models; one being robust and the other not. As shown in Fig. 2, the transfer process utilizes both models, while the source data is not required during transfer. However, once the model has been adapted, only the robust model is required for inference.

Our main contributions are threefold:

- We study a new problem of unsupervised robust domain adaptation in the setting of missing source data.
- A simple yet a very effective method is proposed, which exploits the non-robust pseudo-labels for robustness, to address the problem at hand.
- The proposed method is extensively tested on four benchmark datasets, consistently demonstrating the excellent improvements of over 10% in accuracy.

2. Related Works

Unsupervised Domain Adaptation (UDA). Unsupervised domain adaptation is a topic of broad interest [37, 31, 46, 9, 50, 35]. UDA aims at transferring supervised source domain models to an unlabeled target domain. The traditional UDA works [25, 7, 26, 47, 54] typically focus on solving the adaptation problem using source data, while being oblivious to the adversarial attacks. Despite of being very insightful, the traditional UDA methods are restricted in many practical settings due to the considered assumptions.

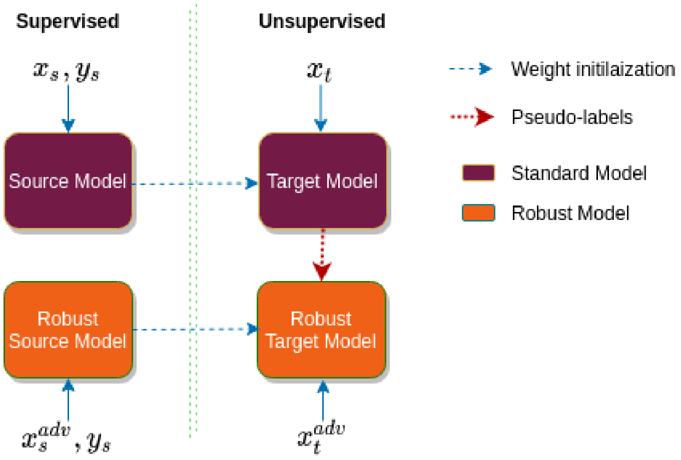


Figure 2: The training in the source domain uses the source data x_s, labels y_s, and adversarial examples x_s^{adv}. The training in the target domain uses the target data x_t and the adversarial examples x_t^{adv} generated using the pseudo labels.

UDA without Source Data. The UDA without source data can be broadly divided into three categories: (i) generative approach [22, 19, 17]; (ii) pseudo-label approach [15, 23]; and (iii) others [38, 53]. The generative approach is often difficult to scale up, as learning to generate the images/features is known to be difficult. On the other hand, pseudo-label based method are easy to handle and have recently provided very promising results [23]. The third category of the methods are either designed under very simplistic settings ([38] is dedicated to the pixel level corruptions) or demand sophisticated mechanism without offering significant gain over the pseudo-label based methods. Therefore, we also leverage the pseudo-labels as in [15, 23].

Robust Training. A flurry of attack mechanisms [4, 5, 3, 30, 33, 2, 27] has been proposed since the vulnerability was shown first by *Goodfellow et al.* [10]. This has also lead strategies that can defend against such attacks, called defense mechanisms [13, 24, 33, 40, 1, 41, 29]. Among them, adversarial training [10, 18] has stood out as the most reliable way to train robust models. We follow the adversarial training method proposed by *Madry et al.* [28] because of being effective, fast, and easy to implement.

Robust Transfer. Our work is also inspired from the recent works on robust transfer learning [42, 39, 48] in supervised settings. A notable work of *Shafahi et al.* [42] shows that a

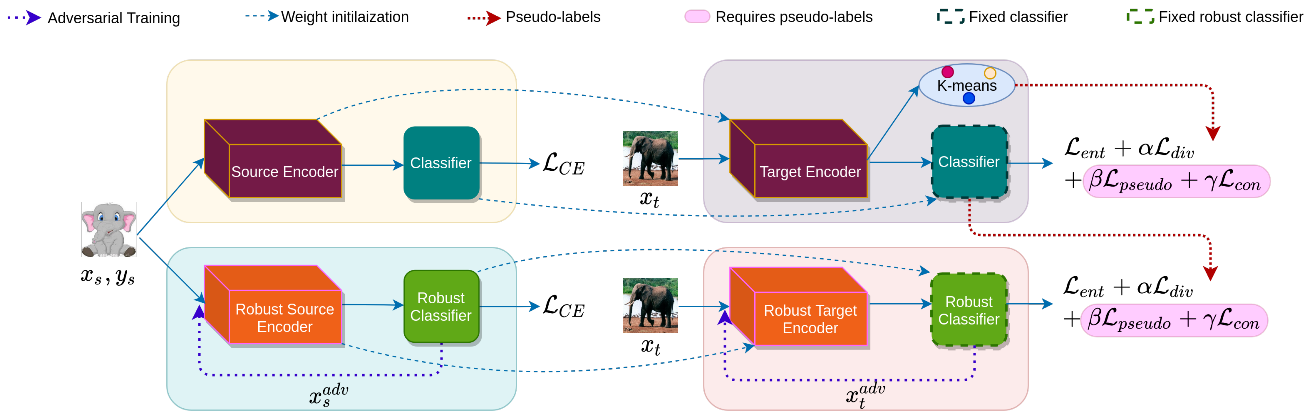


Figure 3: First, a standard (top-left) and a robust model (bottom-left) are trained on source. Then, a target encoder (top-right) is trained by combining four losses with pseudo-labels that are obtained via k-means. Finally, a robust target encoder (bottom-right) is trained similarly to standard target with two modifications. One, the pseudo-labels are obtained from the pre-trained standard target model. Two, adversarial images are generated to facilitate adversarial training.

robust source feature extractor can be effective in preserving robustness, while maintaining *sufficiently high* accuracy on the clean samples. On the other hand, [39, 48] show that the robust pre-trained models not only perform well on targets without adversarial training, but also improve the accuracy on clean samples. These results strengthen the hypothesis that robust models also transfer better. However, the existing methods are neither developed or tested in the settings of unsupervised domain adaptation.

3. Robust Adaptation

In the following we elaborate our methodology for unsupervised robust domain adaptation for multi-class classification problem without access to the source data. Given a dataset $\{(x_s^1, y_s^1), (x_s^2, y_s^2), \dots, (x_s^n, y_s^n)\}$ where $(x_s^i, y_s^i) \sim \mathcal{D}_s$ comes from the source domain, our goal is to train a model that can predict target labels y_t for the corresponding target images x_t where $(x_t, y_t) \sim \mathcal{D}_t$ and is robust to adversarial examples at the same time. We can broadly separate the process into two phases. In the initial phase, we train a model on the source domain in a supervised fashion, and in the final phase, we adapt the model to the target domain. Formally, we need to learn a function $f_s : X_s \rightarrow Y_s$ on the source domain and use that information along with the target data to learn another function $f_t : X_t \rightarrow Y_t$. To this end, we train two models in each domain (source and target), one of them following the standard protocol and one being robust to adversarial examples. We propose to train four models in total: source model, robust source model, target model, and robust target model, as shown in Figure 2. For the sake of brevity, we will refer to pseudo-labels obtained from standard and robust models as *non-robust pseudo-labels* and *robust pseudo-labels*, respectively.

3.1. Source Training

A deep neural network is trained on the source domain by minimizing the standard cross-entropy loss given by,

$$\mathcal{L}_s(f_s; X_s, Y_s) = \mathbb{E}_{(x_s, y_s) \sim \mathcal{D}_s} \mathcal{L}_{CE}(f_s(x_s), y_s). \quad (1)$$

Besides a standard source model, we also train a robust source model. Here, the objective is to learn a function on the source domain that is robust to adversarial images. We generate adversarial perturbations η under the l_∞ threat model. This leads to minimizing the worst case cross-entropy loss, within the l_∞ ball of fixed radius, as follows,

$$\mathcal{L}_s^r(f_s; X_s, Y_s) = \mathbb{E}_{(x_s, y_s) \sim \mathcal{D}_s} \max_{x' \in S(x_s)} \mathcal{L}_{CE}(f_s(x'), y_s), \quad (2)$$

where $S(x) = \{x' \mid ||x - x'||_\infty < \epsilon\}$ and ϵ is the perturbation threshold. Note that finding a sample within $S(x)$ that maximizes the cross-entropy is computationally challenging due to infinitely many samples in $S(x)$ and no closed-form solution. Thus, we empirically generate adversarial examples using Projected Gradient Descent (PGD) and perform adversarial training [28].

Both standard and robust source models described in Figure 3 have two components, namely, an encoder and a classifier. We will use $\Phi_s : X_s \rightarrow \mathbb{R}^d$ and $\Phi_s^r : X_s \rightarrow \mathbb{R}^d$ to denote the standard and robust source encoders, respectively. Similarly, the corresponding classifiers are denoted as, $\delta : \mathbb{R}^d \rightarrow \mathbb{R}^C$ and $\delta^r : \mathbb{R}^d \rightarrow \mathbb{R}^C$, for feature dimension d and C classes. We will make use of only two classifiers for both source and target domains. Both classifiers are trained on the source data, and will remain unchanged for the target, similar to [23].

3.2. Target Training

Our target training stage assumes that only the source trained models are available. Furthermore, the target data are provided without class labels. Our method for target only training is inspired from the source hypothesis transfer [23], which has shown impressive performance on the standard unsupervised domain adaptation. In this work, we extend [23] to the case of robust model adaptation. Similar to the source domain, our approach relies on two separate models in the target domain, namely, the standard and robust target models. We initialize the weights of each model with the corresponding source models. During the adaptation process, the encoders are optimized while keeping the classifier fixed. In the target domain, standard and robust models are trained differently. In the following, we will first present our approach for training standard model followed by the same for the robust model. The key aspects of our method is summarized in Algorithm 1.

3.2.1 Standard Model

The idea of standard training is to learn a standard target encoder $\Phi_t : X_t \rightarrow \mathbb{R}^d$ that generates features which align closely with the corresponding source feature distribution, making it possible to re-use the source classifier $\delta(.)$. No access to source data restricts us to perform direct alignment between the two features as in [47]. To address this problem, our approach involves (i) entropy and divergence of the predicted labels, (ii) pseudo-label based supervision, and (iii) contrastive target features. The first two aspects are borrowed from [23] and other prior works [51, 43]. The aspect of using contrastive feature learning is proposed in this work, for the first time to address the problem at hand.

Entropy and Divergence: Entropy minimization is a widely used technique for unsupervised domain adaptation [51]. The Shannon entropy [43] for a prediction probability $\hat{p}_i$ of class i is defined as,

$$\mathcal{L}_{ent} = -\sum_i \hat{p}_i \log \hat{p}_i. \tag{3}$$

Unfortunately, entropy minimization can produce degenerate labels with loss converging to zero. Therefore, we take the information maximization (IM) [8] approach as adopted by [23]. IM adds an additional diversity term that pushes the predicted labels to be uniformly distributed avoiding the trivial outcome of the same one-hot vector for all inputs. Let q_i be the average probability of a prediction for the class i, then the diversity loss is defined as,

$$\mathcal{L}_{div} = \sum_i q_i \log q_i. \tag{4}$$

Non-robust Pseudo-labels: While IM can make the model confident while ensuring diverse prediction, it may still push the output towards incorrect prediction in certain cases. In order to overcome such undesired behaviour, [23] proposed to use pseudo-labels [20] in addition to IM for better supervision. We use two-step weighted k-means clustering on the feature space to obtain pseudo-labels as described in [23]. Let $\hat{y}$ be the pseudo-label obtained for the image x. Then, the pseudo loss is defined using the cross-entropy as,

$$\mathcal{L}_{pseudo} = \mathcal{L}_{CE}(\delta(\Phi_t(x)), \hat{y}). \tag{5}$$

Constrastive Feature Learning: We use the obtained pseudo-labels also to learn the discriminative features in the target. The proposed use of the contrastive loss is inspired by [44, 14], which were originally used in different contexts. The contrastive loss minimizes the intra-class distance, while maximizing inter-class distance between the encoder features. For two input images x_1, x_2 with pseudo-labels y_1, y_2, the pair-wise contrastive loss is given by,

$$\mathcal{L}_{con} = \frac{1}{2}[y \cdot D^2 + (1-y) \cdot \max(0, m-D)^2], \tag{6}$$

where $y = \mathbb{I}_{\{y_1=y_2\}}$, $D = ||\Phi_t(x_1) - \Phi_t(x_2)||_2$ and $m > 0$ is the margin between features of different classes.

To optimize the target standard model, we minimize the weighted combination of the loss terms described above. In this context, the minimized loss is given by,

$$\mathcal{L}_t(f_t; X_t, Y_t) = \mathcal{L}_{ent} + \alpha\mathcal{L}_{div} + \beta\mathcal{L}_{pseudo} + \gamma\mathcal{L}_{con}, \tag{7}$$

where α, β, and γ are the weights corresponding to the respective loss functions.

3.2.2 Robust Model

The idea of robustness transfer is inspired by some of the recent works [42, 48, 39] in this direction. Some existing works perform the knowledge transfer using a robust source model. Such transfer is shown to preserve the robustness also for the new tasks. In our work, we show that the robust source model also transfers robustly to the target, up to some extent. To improve the robustness further, we propose adversarial training also in the target. Unfortunately, the adversarial robust training often requires labeled examples. One may consider using the pseudo-labels from the robust model. However, due to the trade-off between clean and robust accuracy [55], this process will result into less accurate pseudo-labels. Instead, we propose to obtain the required pseudo-labels using the standard model. Note that the clean accuracy of the standard models is higher than that of the robust ones. More importantly, the pseudo-labels obtained using a standard model, for clean samples, are sufficient to generate the required adversarial examples.

At this point, we wish to transfer the source robustness using a robust source model. On the other hand, we require better pseudo-labels to generate adversarial examples.

Therefore, we use both robust and standard source models and transfer them to the target domain. In this process, the robustness of the robust model is reinforced by using the pseudo-labels from the standard model. Additionally, we believe that the used pseudo-labels offer better domain alignment by means of minimizing the cross-entropy and pair-wise contrastive losses of (5) and (6), respectively.

Adversial Target Examples: We generate adversarial examples using PGD method [28]. These generated images are used to compute the IM loss of (3) and (4). We train two models independently on the target domain. The standard model is trained first, followed by the robust one. The final loss use for the robust training is given by,

$$\mathcal{L}_t^r(f_t^r; X_t, Y_t) = \mathcal{L}_{ent}^r + \alpha\mathcal{L}_{div}^r + \beta\mathcal{L}_{pseudo}^r + \gamma\mathcal{L}_{con}^r. \quad (8)$$

Algorithm 1 Target adaptation using two models.

```
Initialize weights of $\Phi_t(.)$ with $\Phi_s(.)$
for $epoch < MaxEpochs$ do
    Obtain pseudo-labels $\hat{y}$ via k-means
    for each mini-batch do
        Update weights of $\Phi_t(.)$ using Eq (7)
    end for
    if $epoch \% update = 0$ then
        Update the pseudo-labels
    end if
end for
Initialize the weights $\Phi_t^r(.)$ with $\Phi_s^r(.)$
Obtain pseudo-labels $\hat{y}$ via $\delta(\Phi_t(x))$
for $epoch < MaxEpochs$ do
    for each mini-batch do
        Obtain $x_t^{adv}$ for $x_t$ using $\hat{y}$ and $\delta(\Phi_t^r(x))$
        Update weights of $\Phi_t^r(.)$ using Eq (8)
    end for
end for
```

3.3. Adaptation with a Single Source Model

The method previously presented suggests a model hand over protocol, where the user with the access to the source data provides two models. In some practical scenarios however, both models may not be available. Under such circumstances, we suggest to still *use pseudo-labels with the proposed method for the best outcome of robust adaptation*, irrespective of the model being robust or standard. This suggestion is supported by our extensive experiments. We will present these results as, (i) **Robust source**: uses only the robust source model, (ii) **Standard source**: uses only the standard source model, (iii) **Both**: uses both models. In the source robust case, the robust pseudo-labels are used for adaptation. In the other two cases, non-robust pseudo-labels are used. Needless to say, the source standard case adapts the standard model robustly to the target domain.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments on four benchmark datasets, including one small, two medium and one large-scale dataset. The datasets vary in their number of classes from 7 to 65 and contain between two and four different domains. Office-31 [37] consists of total 4,652 images from three domains - Amazon (**A**), DLSR (**D**) and Webcam (**W**) - each having 31 classes. Office-home [50] is collected in four different domains - Art (**Ar**), Clipart (**Cl**), Product (**Pr**) and Real-world (**Rw**) - each with 65 classes a total of 15588 images in the dataset. PACS [21] contain 9991 images from four domains - Art (**A**), Clipart (**C**), Photo (**P**) and Sketch (**S**) - where each image belongs to one of 7 classes. The largest considered dataset, VisDA-C [36] has only two domains - Synthetic (**S**) and Real (**R**) - with 152k and 55k images respectively. Therefore, each of the 12 different classes has a significantly larger number of samples than in the other datasets. For all datasets and all the adaptation tasks, we randomly split both the source and the target domain samples into train/val/test (0.7/0.1/0.2).

Network Architecture. We use ResNet50 [11] as the backbone feature encoder for all our experiments. Moreover, we initialize it on the source with weights pre-trained on ImageNet [6]. For robust source training, we use weights obtained after adversarial training[2] on ImageNet. We maintain non-overlapping training, validation and test splits created randomly and evaluate the performance of all methods and tasks on the test split while using the validation split for model selection. For the VisDA-C dataset, we follow the established standard protocol [36] by training our source models on synthetic images and adapting the models on the real images. All the experiments were conducted using the PyTorch framework [34].

Implementation Details. We keep the batch size fixed to 64 for all the datasets, tasks and methods. The learning rate is set to 10^{-3} for the classifier and the feature bottleneck layers while the backbone is trained at a slower rate of 10^{-5} using the Adam [16] optimizer. We use early stopping in all training-runs with a stop patience of 5. For generating adversarial examples we set the number of PGD [28] steps to 20, attacking under the l_∞ norm ($\epsilon = 4/255$) with a relative step size equal to $0.1/0.03$. Given the large size of Vis-DA, the source model reaches high-accuracy in just 2 epochs and the adaptation process is performed for 5 epochs. For all other datasets, we train the source model for 20 epochs and run adaptation for 10 epochs. The loss components weights $\alpha = 1.0$, $\beta = 0.3$, are borrowed from [23] and $\gamma = 0.2$.

Three Cases of Our Method. Recall that we also account for the case where only a single source model is available, as described in Section 3.3. When only the standard source

[2] https://github.com/MadryLab/robustness

Method	**Office-31** [37]		**Office-home** [50]		**PACS** [21]		**VisDA-C** [36]	
	Adv acc	Clean acc	Adv acc	Clean acc	Adv acc	Clean acc	Adv acc	Clean acc
ADDA [47]	0.4	75.0	0.9	50.2	0.8	64.6	0.9	70.1
SHOT [23]	0.0	**87.6**	0.3	**65.8**	0.0	57.0	0.3	**79.0**
ADDA *robust*	49.6	57.6	30.1	37.7	41.6	59.8	34.0	46.3
SHOT *robust*	67.6	73.1	46.1	53.1	50.5	57.3	25.0	34.3
Ours (Robust source)	72.8	76.4	52.4	59.2	66.5	72.7	51.4	63.6
Ours (Standard source)	81.2	85.7	55.4	62.7	**84.6**	**89.4**	**66.7**	75.8
Ours (Both)	**83.5**	87.0	**58.0**	65.1	76.9	83.6	65.0	74.9

Table 1: Accuracy on adversarial and clean images on the test data averaged over all domain adaptation. All our methods have higher adversarial accuracy compared to the baselines. The performance of our methods on clean samples is comparable and mostly higher than the other methods. The best accuracy is presented in bold and the second best is underlined.

Method	A → C	A → P	A → S	C → A	C → P	C → S	P → A	P → C	P → S	S → A	S → C	S → P	**Avg.**
w/o Contrastive	73.8	93.4	41.1	68.0	83.8	42.9	71.0	60.1	29.4	6.6	22.0	17.1	50.8
w/o Cross-entropy	92.3	93.4	48.3	78.5	91.9	65.4	82.4	50.3	37.2	2.4	8.7	2.7	54.5
w/o Entropy	88.7	94.3	43.5	77.8	88.0	52.5	82.2	67.4	40.6	14.4	55.0	41.6	62.2
w/o Adv. Images	87.0	93.7	24.9	76.3	92.8	21.5	74.6	88.3	17.3	**76.1**	**91.5**	**54.2**	66.5
w/o Diversity loss	**96.2**	**99.7**	90.2	**88.8**	**99.1**	89.6	**93.9**	81.0	33.0	22.2	59.1	30.8	73.6
Ours (both)	92.1	92.8	**94.9**	77.3	91.6	**95.2**	78.8	**94.2**	**71.0**	30.7	84.2	20.4	**76.9**

Table 2: Ablation study of our (both) target model on PACS datatset. The contrastive loss term, entropy term and diversity loss term were removed from both (standard and robust) the target models while Cross-entropy term was only removed from the target robust model since removing from both will make it very hard to adapt.

model is available, we initialize the encoder with weights obtained after adversarial training on ImageNet, while the classifier is initialized randomly. This is done due to absence of a corresponding robust model in the source for initialization. To distinguish among the three scenarios, we refer to our method as Ours (robust source), Ours (standard source) or Ours (both) when only robust, only standard or both the models are available in the source domain.

4.2. Baselines

Since,to the best of our knowledge, there is no previous work on robust domain adaptation, we construct two baselines. The baselines use state-of-the-art domain adaptation approaches [47, 23] that we adapt to use adversarial training in the source domain.

The first adapted approach is Adversarial Discriminative Domain Adaptation (ADDA) by [47] which, in addition to the data our approach requires, also uses source data in the adaptation phase. We perform adversarial training [28] in the source domain and follow the target adaptation protocol as described in [47].

The second method we use for comparison is Source Hypothesis Transfer (SHOT) [23]. This approach is, similar to our approach, a source-free method, and thus, does not require access to source data during adaptation. We again modify this approach to use adversarial training [28] in the source domain and subsequently follow the adaptation strategy as described in [23].

Most of the source-free UDA methods require image/feature generation [22, 19, 17] which are difficult to scale while ensuring robustness on large datasets like VisDA-C [36]. Other recently introduced approaches [38, 45, 51] that are designed for pixel level corruptions [12] do not extend well to more complex domain adaptation tasks we present in this paper.

4.3. Results

We evaluate all our methods along with the introduced baselines on all four datasets, Office-31, Office-home, PACS and VisDA-C. We report the averages over all classes and adaptation tasks for all datasets. An exception is the VisDA-C dataset, where we follow the standard protocol and report the per-class average for Synthetic (S) to Real (R). The accuracies on adversarial attacks are visualized in Fig. 1 which shows that all our methods outperform the baselines. More detailed results on all the datasets are presented in Table 1.

All introduced methods perform consistently better than the baselines on adversarial images on all datasets. Besides having a good performance in the case of adversarial attacks, our models also perform competitively on clean samples. On the PACS dataset, our (standard source) method outperforms all others, both in clean and adversarial accuracy. On Office-31 and Office-home, our method (both) improves robust accuracy by 15.9% and 11.9% respectively while only loosing 0.6% and 0.7% clean accuracy compared to the best non-robust model. Overall, our two best approaches (standard source and both) significantly improve adversarial accuracy while only reducing clean accuracy slightly (max -4.1% on VisDA-C). Fig. 6 shows randomly selected adversarial images from the target domain (Art) which are classified correctly and incorrectly by the five different robust models adapted from the Real-world (Rw) domain in Office-home.

It is important to note that the clean accuracy for both ADDA and SHOT drops considerably if they are directly trained robustly. This is in line with the general observation that robust models tend to hurt the performance on clean samples [55].

In two of the datasets (Office-31 and Office-home) our method which utilizes both standard and robust source models performs best. This is switched in the other two datasets, where our method which only requires the standard source model is better. To further analyze this behavior, we create two subsets of the data by only keeping all the images that belong to the first[3] 10 and 32 classes respectively, both in the source and the target domain. This is done to ensure that the number of samples per class remain the same in all the three cases.

Results for these adaptation tasks are illustrated in Fig. 4 where we compare the method using only the standard model against the method using both source models. Fig. 4 indicates that in the case of few classes having only standard source model suffices for robust adaptation. However, if a larger set of classes needs to be handled, it is better to make use of both the standard and robust source model and follow the procedure as described in Section 3.

4.4. Ablation Study

We study the impact of each of the components in our model on the PACS dataset in Table 2. Removing the contrastive loss from both target model and the target robust model reduces the average performance. Similarly, the target accuracy decreases without the entropy minimization term or the diversity loss. The absence of cross-entropy loss calculated with help of pseudo-labels also makes it hard for the model to adapt well. Furthermore, we find that generating adversarial images using pseudo-labels also plays a significant role in improving the robust accuracy of the model.

Recall that we require pseudo-labels to calculate the cross-entropy and contrastive loss and generate adversarial images in the target domain. To analyze the impact of pseudo-labels, we visualize the features of the adversarial images for the adaptation from Art (A) to Cartoon (C) on PACS under four different scenarios. We make use of PCA followed t-SNE [49] for dimensionality reduction. Fig. 5a shows the target features of the robust source model. Next, we perform domain adaptation without using any pseudo-labels and plot the encoder features as shown in Fig. 5b. In the next setting, we use pseudo-labels generated from a robust target model instead. Fig. 5c shows that adversarial test images in this scenario form better clusters in the feature space. Finally, we compare it with our protocol, where we generate pseudo-labels from the standard target model to train the robust target encoder in Fig. 5d. Fig. 5 clearly demonstrates that the learned features become more and more discriminative, forming better clusters as we introduce pseudo-labels and obtain them from the standard target model instead of the robust target model.

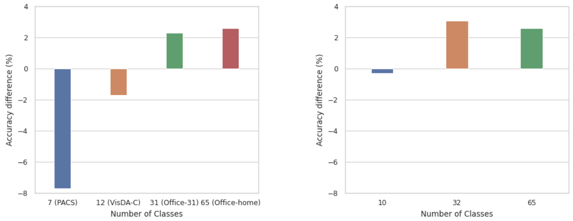


Figure 4: Performance of our method that use both source models relative to our method using only standard source model. The plot on the left shows the comparison on all the four dataset while on the right compares performance on Office-home by varying the number of classes.

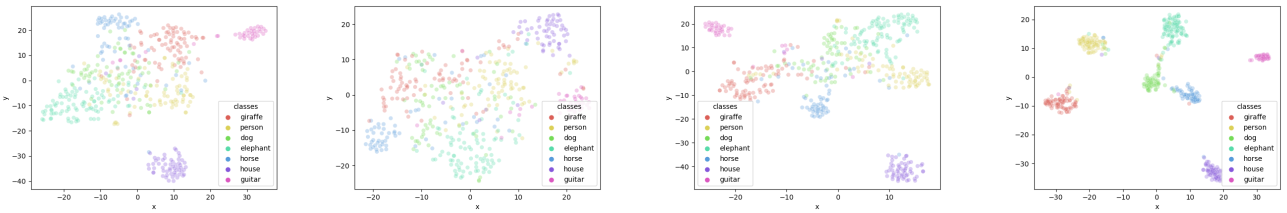
5. Discussion

Based on our experimental evaluations, we attempt to answer some key questions. We believe that our answers to these questions help to better understand the outcome of our study as well as the problem addressed in this paper.

Do robust models transfer robustly?–Yes. Beside the proposed method, our baselines also allow us conclude that the robust models indeed transfer robustly. In particular, two baselines, SHOT robust and ADDA robust do not even use the adversarial examples in the target domain. The performance of these methods on the adversarial examples are noteworthy. This observation is in accordance to the existing works [42, 48, 39], although in different settings. In the setting of this paper, the robust source models are found to be very useful for datasets with many classes.

Which pseudo-labels to use?–Non-robust. Our experiments demonstrate the clear benefit of using non-robust pseudo labels for robustness in the target domain. Please, refer to Sec. 3.2.2 and 3.3 for more details. It goes without saying, non-robust pseudo-labels are preferred when non-robust source models are available.

[3] In alphabetical order of the class labels, which is not related to the class complexity.



(a) Before Adaptation (b) No pseudo-labels (c) Ours (robust source) (d) Ours (standard source)

Figure 5: Impact of pseudo-labels on PACS with Art painting as the source and Cartoon as the target domain. The subplots clearly show that the features learned become more and more discriminative, forming better clusters as we introduce pseudo-labels and obtain them from the standard target model instead of the robust target model.

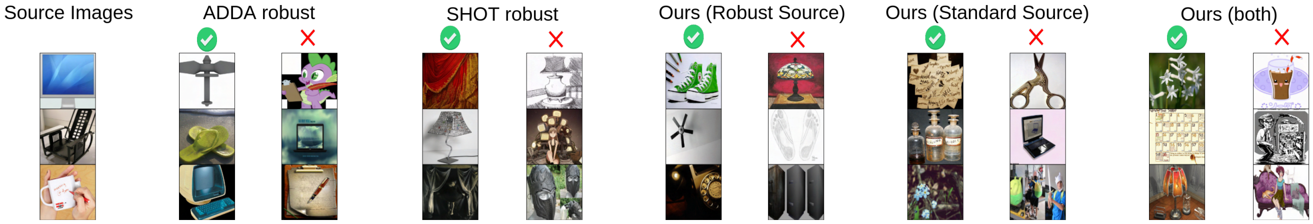


Figure 6: Sample adversarial images in the target (Art) domain of Office-home. The figure shows the correctly classified and misclassified images by the target model for each of the method. The source model was trained on Real-world (Rw) images.

Which model to transfer?–Robust. Provided a good transfer of non-robust models to the target, it has been observed that the robustness can be achieved by generating the adversarial examples in the target. Such robustness however fully relies on the pseudo-labels alone. We observed that for datasets with few class such transfer is often is not a problem. However, as the number of classes increases, the transfer of non-robust models followed by the robust training is not a good idea. Please, refer to Fig. 4 for robust and non-robust models transfer for increasing number of classes. Such behaviour can be attributed to the following: as the number of classes increases, the chanced of pseudo-labels being incorrect in the target becomes higher. As the transfer of robust source model does not fully rely only on the pseudo-labels, we suggest to adapt the roust source model. This suggestion is however, meant to be followed for the two models case. Otherwise, we recommend to transfer the non-robust source model followed by robust target training (using the method proposed in this paper).

What makes any given model better?–Contrastive loss. The use of contrastive loss for the addressed problem is found to be very helpful in all three cases of the model availability presented in Sec. 3.3. This can be observed in Tab. 2 and 1. Note that the baseline SHOT robust differs from our method with robust source in terms of the contrastive feature learning. Please, refer Sec. 3.2.1 for the details.

How do I design the transfer protocol?–Transfer two models. When the availability of source models is not a problem, we suggest to use two models as presented in Fig. 2 and Algo. 1. This may be particularly important, when designing the model transfer protocol is possible.

Do I need to keep two models after transfer?–No. Only using the transferred robust model will offer the adversarial and clean accuracy of Tab. 1. The non-robust model is only used to generate more reliable pseudo-labels, for adversarial examples during robust training in the target domain.

6. Conclusion

We study three different cases of model availability for the unsupervised robust domain adaptation without source data. These cases were chosen to model practical scenarios. In all the three cases, we obtained very promising results, thanks to the proposed method. Our extensive study shows that the transfer of both robust and standard model is often the best choice for the robustness in the target domain. Overall, the non-robust pseudo-labels and contrastive feature learning strategies are found to be very effective, when combined with the existing model transfer methods. In future, we will explore single source models that perform both robust and non-robust predictions, in a multi-tasking fashion. This will avoid sharing two models trained on the the source data.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018.

[2] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[4] 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.

[5] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 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*, pages 15–26, 2017.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[8] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Neural Information Processing Systems*, 2010.

[9] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

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

[11] 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.

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[13] Yunseok Jang, Tianchen Zhao, Seunghoon Hong, and Honglak Lee. Adversarial defense via learning to generate diverse attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2740–2749, 2019.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[15] Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. Domain adaptation without source data. *arXiv preprint arXiv:2007.01524*, 2020.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[18] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world.

[19] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021.

[20] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[21] 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*, pages 5542–5550, 2017.

[22] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[24] 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.

[25] 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.

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[27] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

[29] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3394, 2019.

[30] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

[31] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[32] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[33] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[35] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[36] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2102–21025, 2018.

[37] 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.

[38] Roshni Sahoo, Divya Shanmugam, and John Guttag. Unsupervised domain adaptation in the absence of source data. *arXiv preprint arXiv:2007.10233*, 2020.

[39] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? 2020.

[40] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

[41] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019.

[42] 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*, 2020.

[43] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[44] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[45] Yu Sun, Xiaolong Wang, Liu Zhuang, John Miller, Moritz Hardt, and Alexei A. Efros. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

[46] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[47] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[48] Francisco Utrera, Evan Kravitz, N. Benjamin Erichson, Rajiv Khanna, and Michael W. Mahoney. Adversarially-trained deep nets transfer better. In *International Conference on Learning Representations*, 2021.

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[50] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[51] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[52] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

[53] Shiqi Yang, Yaxing Wang, Joost van de Weijer, and Luis Herranz. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020.

[54] Jan-Nico Zaech, Dengxin Dai, Martin Hahner, and Luc Van Gool. Texture underfitting for domain adaptation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, page 547–552. IEEE Press, 2019.

[55] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.