ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Unsupervised Robust Domain Adaptation without Source Data

Master's Thesis

Peshal Agarwal
Master of Science in Statistics
Department of Mathematics

Advisors: Dr. Danda Pani Paudel, Jan-Nico Zäch
Supervisor: Prof. Dr. Luc Van Gool, Prof. Dr. Peter Lukas Bühlmann

March 12, 2021

Abstract

Unsupervised domain adaptation refers to the setting where labeled data on the source domain is available for training, and the goal is to perform well on the unlabeled target data. The presence of a domain shift between source and target makes it a non-trivial problem. 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 work aims to answer the question of finding the right strategy to make the target model robust and accurate in unsupervised domain adaptation without source data. The major findings of this work 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.

Acknowledgements

I would like to express my sincere thanks to my advisor Dr. Danda Pani Paudel, Mr. Jan-Nico Zäch and my supervisors Prof. Dr. Luc Van Gool and Prof. Dr. Peter Lukas Bühlmann. I feel fortunate to have access to excellent guidance for my master thesis. I am indebted for their supervision, feedback, and suggestions for the project work. Everyone was very supportive and ensured smooth progress despite the challenging times due to the novel Coronavirus. I would also like to thank my friends for helping me out, motivating me, and providing moral support during tough times. Last but foremost, I would like to thank my family for their constant support and encouragement earnestly, without which it would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning techniques have grown leaps and bounds over the past decade or so, improving the capacity of machine intelligence and moving closer to the long-term goal of artificial intelligence. In particular, deep learning has made it possible for machines to perform tasks that were unfathomable to humans 20 years ago. The driving force behind such advancements is the wide variety of applications and the level of performance of AI models, even surpassing human accuracy in some cases (image classification, object recognition, playing Go, etc.). Its ability to generalize well on unseen examples has been credited for most of the success so far. This has been possible partly due to the advancement in the field and partly due to the availability of a large amount of data. The deep neural networks are trained using data hungry algorithms and are able to generalize well on the unseen data.

Almost all learning paradigms rely on one of the basic assumptions: the training and test samples both belong to the same distribution. While this may hold in some cases, it is far from being the case for most real-world problems. Change is the fundamental law of nature and is something we frequently observe in our life. Whenever we move to a different city, read a new text or listen to foreign accents, we recognize the *shift* from the one we are used to. In the context of machine learning, this is called a distribution shift. While humans can adapt to a new setting without much difficulty, machines, in general, find it hard. This is not surprising in the light of the distributional assumption on which the models are trained. The long-term vision of developing AI and integrating them into devices to assist humans, such as autonomous driving, medical imaging, and numerous other applications, makes the distribution shift issue inevitable.

Formally, the distributional shift occurs when the model is trained on one (source) distribution but is tested on another (target). This often leads to poor model performance, which might be a simple task for humans. We focus on a closed set unsupervised domain adaptation setting under image classification. We have labeled source data but unlabeled target data with the same set of classes as the source. A general approach is to align the source and target distribution of learned representation in the feature space. Hoping that such an alignment also produces discriminable features, we can re-use the source classifier on the target to obtain predictions. This is has been extensively studied and generally lead to a min-max training objective [12, 37]. Such adversarial training mechanisms are, in general, hard to stabilize and often lead to divergence [63]. Most methods require access to both source and target data during target training, which may not be feasible in all scenarios. Unlike most of the previous work, our work is inspired by a method [35] that neither relies on any min-max objective nor requires source data during adaptation.

Apart from this, the gap between human and machine perception also makes models susceptible to an adversary. Even models with very high "performance" can easily be made to fall into a well-crafted trap. It was first shown in the context of computer vision [66] and has since been studied along various dimensions [16, 41]. They can be generated by a malicious adversary as well as are found in the real world [28]. This brittleness of machine learning models and the existence of adversarial examples raises concerns about security (attack on speech to text [6]) and safety (in autonomous driving), its deployment on systems being used in finance, search, healthcare, etc.

A "valid" adversarial example is the one that can be easily classified correctly by a human observer, but the machine learning model fails to do so. Such an idea of a legitimate input is hard to formalize. Hence, researchers use proxy metrics such as the l_p ($p = 0, 1, 2, \infty$) norm of the added "noise" in the process of adversarial generation and defense. An attack algorithm can be viewed as solving (1.1) an optimization problem. This can be intractable owing to the highly non-linear and non-convex nature of the function f for most models (e.g., DNNs). However, the same gradient-based methods, which are used to train these models, come to the rescue. In practice, approaches implementing gradient-based optimization reach reasonably good approximations of the minimal perturbation (1.1), which are often indistinguishable to human eyes. In this current work, we use Projected Gradient Descent [41] for generating adversarial examples under the l_2 attack model.

$$x^{adv} = x + \arg\min_{\epsilon}\{\|\epsilon\|_p \mid f(x+\epsilon) \neq f(x)\} \tag{1.1}$$

In this work, we address 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 very challenging problem. Up to our knowledge, this problem is addressed for the first time in this work. We study 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 use information from the source domain efficiently?
- How do we perform robust domain adaptation if only one model (robust or standard) is available?
- Is the best adaptation approach dataset dependent?

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. The problem of unsupervised domain adaptation without source data has recently been studied in [35, 24, 52, 33, 29, 27, 74]. However, the prevailing work does not consider robustness. In this work, we first show that robust domain adaptation performs reasonably well within the setup mentioned above when the method of [35] 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 [35] under the adversarial perturbations for robustness and then improve the performance by over 10% in accuracy consistently across four benchmark datasets. Such improvement is achieved by exploiting non-robust pseudo-labels and the target's pair-wise contrastive learning scheme. Our work's central finding is that *robust domain adaptation can greatly 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 complementary to each other. The exploitation of non-robust pseudo-labels in this fashion also offers significantly better performance compared to its robust counterpart. This observation leads us to suggest a new source data training and model sharing protocol. In the source domain, we suggest training two models; one being robust and the other not. 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 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.

The thesis is organized as follows. In Chapter 2, we refer to all the work closely related to this topic and has connections to our proposed framework. Chapter 3 briefly discusses the theoretical concepts necessary to understand the concept of domain adaptation and adversarial machine learning. We present our model in Chapter 4, providing details of our method and motivation for our choices. Chapter 5 consists of detailed experimental results on multiple benchmark datasets along with ablation studies and experimental setup. We discuss the advantages of approach work over the past work and point to some of the drawbacks in Chapter 6. We finally conclude in Chapter 7 by summarizing major findings and potential future research directions.

Chapter 2

Related Work

Over the last decade or so, there has been growing research along unsupervised domain adaptation focusing on neural-network-based methods. Multiple approaches have been proposed, including source and target domain distribution alignment, modifying normalization statistics, and image translation, to name a few. A recent survey [73] describes most of the methods in detail. However, we discuss the ones closely related to our work below.

2.1 Domain Adaptation

Domain invariant feature learning is one of the most popular approaches to attempt solving the domain adaptation challenge. It is based on the assumption that one can learn a feature extractor that generates domain invariant features. This way, we can reuse the classifier on the target domain trained with labels on the source domain. Early work is based on divergence based methods (described in detail in Chapter 3) such as Maximum Mean Discrepancy (MMD) [17], and Correlation Alignment (CORAL) [62] which is similar to MMD with a polynomial kernel and used second-order statistics to compute distances. Contrastive Domain Discrepancy (CCD) [22] is another method that uses contrastive loss to minimize intra-class discrepancy while maximizing inter-class distance. Unlike MMD and CORAL, CCD focuses on class conditional distribution for divergence.

Another line of work is inspired by the idea of a two-player game introduced in GAN [15], often referred to as adversarial domain adaptation. [12] first proposed a Gradient Reversal Layer along with a discriminator to learn domain invariant features. It forms a single network that is trained end-to-end with a standard cross-entropy loss that ensures the source's discriminative features and a (binary) discriminator loss to align the features. This has further motivated us to split the source and target training and move closer to GANs. Generative Adversarial Network is a deep generative model consisting of a generator G and a discriminator D, which push against each other. Adversarial Discriminative Domain Adaptation (ADDA) [67] proposes to have separate feature extractors for the source and the target domains. While the source encoder is trained in a supervised manner along with the labels, the target encoder is trained in an adversarial manner with a discriminator so to align its features with that of the source keeping the source encoder fixed. The source classifier is then later reused along with the target encoder during inference. It has further been extended with different variants of GAN such as CycleGAN [20], conditional GAN [38] and others [56, 72, 9].

Some researchers have shown that the input distribution shift can be taken care of by adjusting the batch normalization parameters in each layer. Adaptive Batch Normalization (AdaBN) [34] introduced this simple technique of re-computing the mean and variance for the target domain starting with a pre-trained source DNN. The simplicity is in sharp contrast to adversarial and divergence-based methods. [71] replaces the moving estimates of mean and variance with full estimates while [7] introduce pseudo-labels in addition to batch normalization. [64] implements model augmentation taking the help of an auxiliary task that shares its representation with the original task.

One of the requirements of all of the methods discussed above requires access to source data during the adaptation process. However, some of the more recent methods are not bound to that constraint and perform surprisingly well even in the absence of source images. Approaches like these lines can be broadly classified into three categories. (i) Generative approach: The idea here is to generate target-like images using a class conditional GAN. [33] proposes a collaborative class conditional generative adversarial network that does not require source data. Gradient Reversal Layer is deployed along with a discriminator and GAN that generates labeled data on the target domain in [29]. [27] rely on artificially generated source data to allow for source-free adaptation. Besides, GAN being tricky to train and stabilize, data generation would be computationally expensive in the case of large-scale datasets. (ii) Pseudo-labels based: Instead of generating synthetic images, one can obtain pseudo-labels to guide the adaptation process. [24, 35] follow this approach and generate pseudo-labels with the help of unsupervised clustering techniques. These have shown to be relatively inexpensive, easy to implement, and effective. (iii) Entropy-based: [52, 71] propose methods that aim to fine-tune the batch normalization layers by minimizing entropy and its variants. This works well under simplistic settings (images corruption [19]) but is difficult to extend to more complex domain adaptation tasks.

Source Hypothesis Transfer (SHOT) introduced in [35] is most closely related to our work. It seems to combine some of the previous approaches, including entropy minimization [71] and generating pseudo-labels [7] along with label smoothing in source training and a diversity term to avoid degenerate label prediction. On top of these, it also proposes to add a fully connected layer and a batch normalization layer on top of standard architectures like ResNet [18]. It differs significantly from the earlier approaches. It integrates entropy minimization, diversity maximization (together termed as information maximization), and cross-entropy loss calculated using pseudo-labels via k-means clustering the feature space. By carefully weighing each term's contribution in the final loss, it happens to surpass methods that require simultaneous access to source and target data in some domain adaptation tasks.

2.2 Robustness

A flurry of attack mechanisms [5, 8, 4, 45, 47, 3, 40] has been proposed since the vulnerability was shown first by *Goodfellow et al.* [16]. This also has lead to strategies that can defend against such attacks, called defense mechanisms [21, 36, 47, 55, 1, 58, 44]. Among them, adversarial training [16, 28, 41] has stood out as the most reliable way to train robust models. We follow the adversarial training method proposed by *Madry et al.* [41] because of being effective, fast, and easy to implement.

Transfer learning is a popular technique to adapt pre-trained for downstream tasks efficiently. Our work also draws inspiration from some of the recent work on robust transfer learning [59, 54, 68]. Working at the intersection of transfer learning and adversarially robust models [59] shows that robust source feature extractor can be effective in preserving robustness while maintaining *sufficiently high* accuracy on the clean samples. [54, 68] on the other hand, empirically shows that robust pre-trained models not only result in robust target models without adversarial training but also improves the accuracy of the downstream tasks on clean samples. All these results strengthen the hypothesis that more relevant features are learned by robust models resulting in a better transfer. However, the existing methods are neither developed nor tested in the settings of unsupervised domain adaptation.

Chapter 3

Theory

3.1 Multi-Task Learning

Multi-Task Learning (MTL) is a sub-field of machine learning that tries to learn multiple functions for the same input by exploiting the common properties of different functions. The motivation for MTL arises from learning networks designated to perform different tasks that are correlated. MTL avoids training task-specific models from scratch by allowing networks to share intermediate representations of the data. It can be viewed as sharing/transferring information between models to improve the learning mechanism.

3.2 Transfer Learning

Transfer learning is an area of machine learning that utilizes knowledge gained from one problem to solving a different yet related problem. For example, patterns learned from a car recognition model can be modified to identify a truck. Specifically, it refers to the scenario where one wishes to transfer a learned model for a particular task on a particular data distribution to another setting with either a different but related task or data distribution or both. Formally, given a function $f_S(.)$ learned using source data distribution $\mathcal{D}_\mathcal{S}$ for task $\mathcal{T}_\mathcal{S}$, one aims to learn a function $f_T(.)$ on the target distribution $\mathcal{D}_\mathcal{T}$ for task $\mathcal{T}_\mathcal{T}$ where $\mathcal{D}_\mathcal{S} \neq \mathcal{D}_\mathcal{T}$ and/or $\mathcal{T}_\mathcal{S} \neq \mathcal{T}_\mathcal{T}$. Since neural networks are expected to learn a meaningful representation of the input and generalize well to unseen data, transfer learning has found its way into applications ranging from text processing [46] and image recognition [43] to audio processing [13]. This technique is widely popular in data-constrained settings such as limited training samples or few labeled examples for a supervised learning algorithm.

3.3 Domain Adaptation

Distributional shift refers to the scenario where the distribution of the data used to train the model shifts during test time, giving rise to the problem of domain adaptation. Domain Adaptation can be viewed as a special case of transfer learning which will be the main focus of this work. Here, the task remains unchanged, but the domains differ. The objective is to adapt a model trained on one or more *source* domain to a *target* domain. Specifically, domain adaptation aims to correct for the distributional shift in the input domain, which is described as the change in the distribution of the data on which the model was initially trained to the data on which is tested. It is observed in numerous practical applications of AI and has been tackled using various approaches.

Machine learning techniques for domain adaptation can be broadly classified into three categories: supervised learning, semi-supervised learning, and unsupervised learning based on labels' availability.

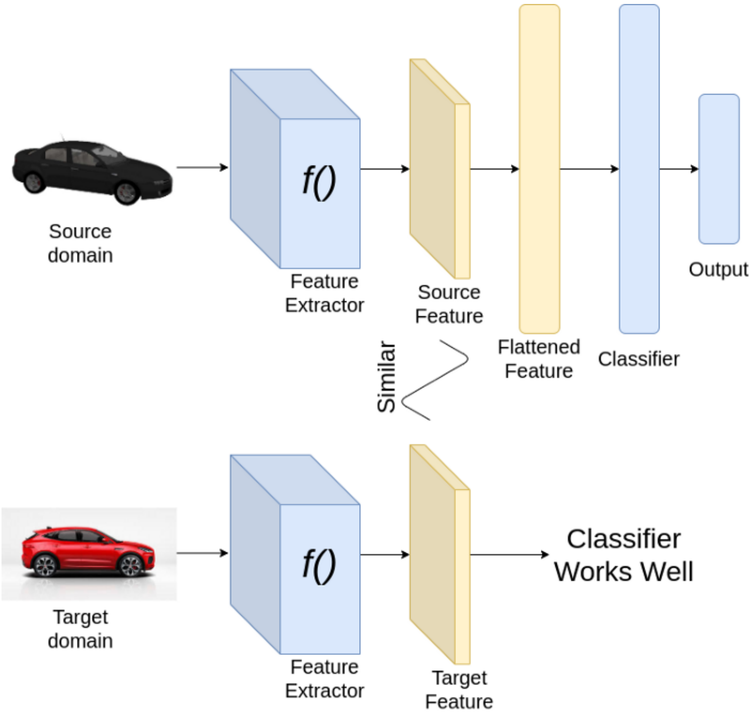


Figure 3.1: A semantic representation of domain adaptation. The objective is generate similar features for the source and the target domain so that the classifier trained on the source can be reused for the target. Image courtesy [42].

We restrain ourselves to the supervised learning on the source domain and unsupervised learning on the target domain, which is commonly referred to as unsupervised domain adaptation [39, 2, 53, 12] and has been extensively studied [73].

Almost all domain adaptation techniques fall into one of the following three categories:

3.3.1 Divergence Based

Divergence-based domain adaptation is built upon the idea of minimizing the divergence between source and target distribution. Maximum Mean Discrepancy [17] or MMD, in short, is a test statistic and is one of the first approaches in divergence-based methods. It was designed to test if two given samples belong to the same distribution or not. The distance between distributions is defined in a reproducing Hilbert kernel space (RKHS). Formally, given two distributions P and Q, MMD is defined as

$$\text{MMD}(P, Q) = \|\mathbb{E}_{X \sim P}[\phi(X)] - \mathbb{E}_{Y \sim Q}[\phi(Y)]\|_{\mathcal{H}} \tag{3.1}$$

where $\phi(.)$ is the feature map from the input domain to the Hilbert space $\mathcal{H}$. Thus, by minimizing MMD, one tries to bring the two feature distributions closer, leading to domain invariant features. While divergence-based methods are generally non-parametric, they might not work well for all kinds of problems (classification, object detection, segmentation, etc.) and datasets.

3.3.2 Adversarial Based

The adversarial-based domain adaptation method draws its inspiration from the idea behind Generative Adversarial Networks (GANs). It models the problem as a two-player game where one player tries to classify the input correctly while the other player pushes the features to be indiscriminable. Similar to

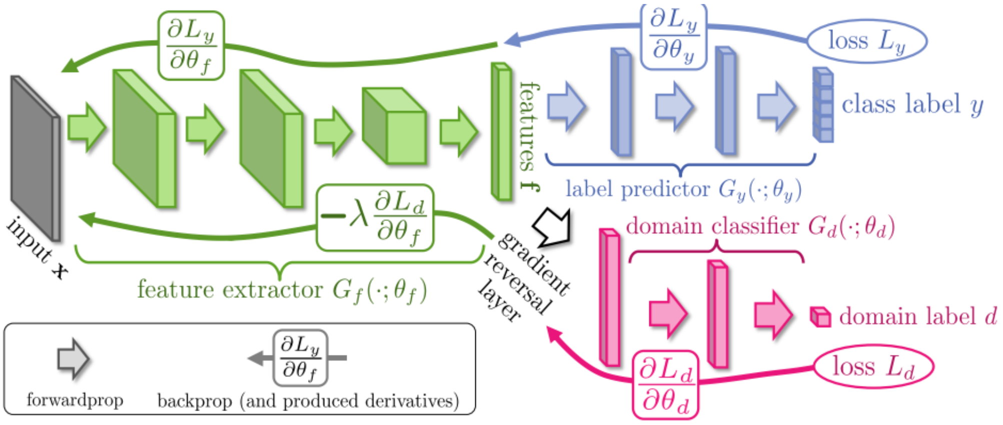

$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features f
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
input x
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
feature extractor $G_f(\cdot;\theta_f)$
domain label d
loss L_d
$\frac{\partial L_d}{\partial \theta_d}$
forwardprop
backprop (and produced derivatives)

Figure 3.2: Unsupervised domain adaptation with the help of a discriminator (in red) and GRL. Image courtesy [12]

GANs, it has two losses, namely, the classification loss and the discrimination loss, but unlike GANs, it does not have a "generator"; instead, it makes use of Gradient Reversal Layer (GRL) [12]. GRL reverses the sign of the gradients from the discriminator loss during the backward pass. Th This idea was first implemented with the help of GRL to solve the problem of domain adaptation in [12] and has since been extended in future works. Figure 3.2 shows the proposed model architecture where both the domain (source and target) inputs pass through the feature extractor followed by label predictor and domain classifier. While standard gradient flows back from the domain label loss, it is reversed inside the feature extractor leading to domain invariant features.

3.3.3 Reconstruction Based

Reconstruction-based domain adaptation is a relatively new approach that draws inspiration from the idea of Image-to-Image translation. The objective is to learn a transformation from images in the target domain to images in the source domain, thereby eliminating the domain gap. One can use a simple encoder-decoder-based architecture to model Image-to-Image translation. Besides, one uses a discriminator to distinguish between the source image and the translation of the target image generated by the encoder-decoder network, as shown in Figure 3.3. This enforces the generator to produces "source-like" images and can be directly used to a model trained on the source to classify during the test time. This idea has been further extended with the help of CycleGAN. Cycle GANs make use of two instead of one encoder-decoder network [11]. Here, we not only transform images from the target domain to the source domain but also vice-versa. The cycle consistency loss helps to ensure that the translated image looks similar to the original when translated back.

3.4 Adversarial Machine Learning

Machine learning models have also been studied under an adversary's presence under a sub-domain called Adversarial machine learning. Even even well studied neural networks architecture trained on well known data sets (MNIST [30], CIFAR-10 [26], CIFAR-100 [26], ImageNet [10], etc.) can *attacked* by well-crafted adversary both with (white-box attacks) and without (black-box attacks) access to the trained weights. Moreover, the adversarial images are visually indistinguishable from the base image. This vulnerability reflects some imperfection in the standard neural network training process and is thus an active research area. This brittleness of machine learning models and the existence of adversarial examples raises concerns about security (attack on speech to text [6]) and safety (in autonomous driving),

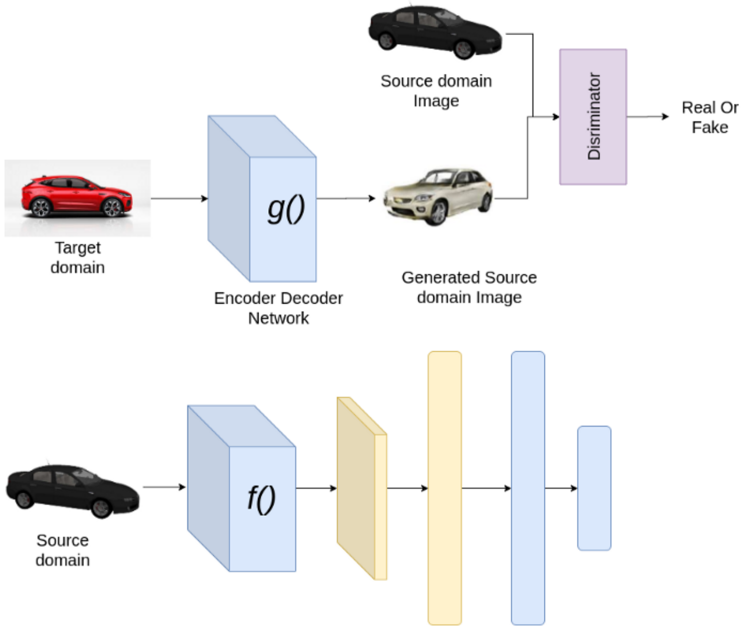


Figure 3.3: Schematic representation of general reconstruction based domain adaptation. The top half shows the GAN-like training to train the encoder-decoder network to be used at test time. The bottom half shows the regular training on the source domain. Courtesy of [42]

its deployment on systems being used in finance, search, healthcare, etc. Representation learned from a single robust classifier can be used for non-classification tasks such as image-to-image translation, in-painting, super-resolution, etc., without task-specific optimization [57]. This hints at other potential applications of robustness beyond security and reliability.

3.4.1 Projected Gradient Descent

An attacker's goal is to find input in the image space that is “close" to the base image but is classified differently than the base image by the network. One of the earliest works argued that it was the linear nature of the model that made adversarial examples easy to generate [16]. They supported their claim by introducing one of the earliest attacking methods called the Fast Gradient Sign Method (FGSM). It is a straightforward yet effective technique at the time. The perturbation is obtained by taking a small step in the direction of the sign of the gradient of the loss function. Fig. 3.4 shows an example of generating an adversarial image via FGSM. While FSGM is fast, it may not be able to find the "best" adversarial input.

Projected Gradient Descent (PGD) attack [41] is one of the most popular methods used to create adversarial images and test the robustness of the models. It is based on the simple idea of applying FGSM over multiple steps with a reduced step size. This further leads to the idea of adversarial training in which one trains with adversarial examples generated on-the-fly instead of the clean samples. It is one of the most effective methods of robust training, and we stick with PGD for training robustness models and evaluate the robust accuracy in this work. We use the Foolbox [50] library for generating adversarial images via PGD.

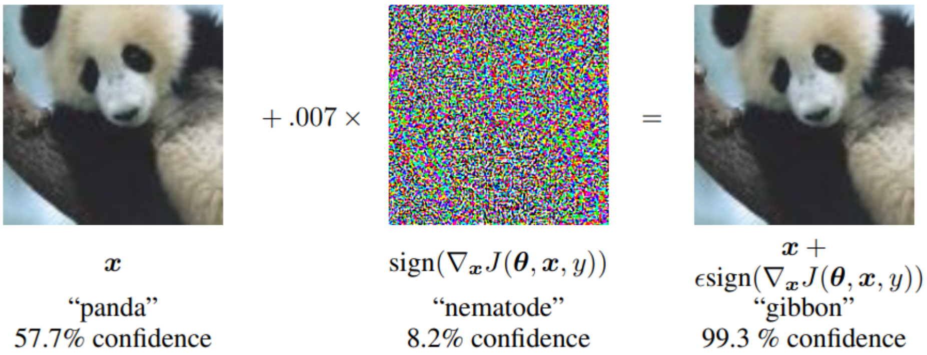

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 3.4: A example of FGSM from [16] applied to GoogLeNet [65] on an image from ImageNet [10]. Note that the noise is visually imperceptible to humans while the model prediction and confidence changes drastically.

Chapter 4

Method

In the following, we elaborate our methodology for unsupervised robust domain adaptation for multi-class classification problems without access to the source data. Given a dataset $\{(x_s^1, y_s^1), \ldots, (x_s^n, y_s^n)\}$ 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 $f_s : X_s \to Y_s$ on the source domain and use that information along with the target data to learn another function $f_t : X_t \to Y_t$. 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: source model, robust source model, target model, and robust target model, as shown in Figure 4.1. 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.

4.1 Source Training

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

$$\mathcal{L}_s(f_s; X_s, Y_s) = \mathbb{E}_{(x_s, y_s) \sim \mathcal{D}_s} \mathcal{L}_{CE}(f_s(x_s), y_s). \tag{4.1}$$

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,

$$\mathcal{L}_s^r(f_s; X_s, Y_s) = \mathbb{E}_{(x_s, y_s) \sim \mathcal{D}_s} \max_{x' \in S(x_s)} \mathcal{L}_{CE}(f_s(x'), y_s), \tag{4.2}$$

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 [41].

Both standard and robust source models described in Figure 4.2 have two components, namely, an encoder and a classifier. We will use $\Phi_s : X_s \to \mathbb{R}^d$ and $\Phi_s^r : X_s \to \mathbb{R}^d$ to denote the standard and robust source encoders, respectively. Similarly, the corresponding classifiers are denoted as, $\delta : \mathbb{R}^d \to \mathbb{R}^C$ and $\delta^r : \mathbb{R}^d \to \mathbb{R}^C$, 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 [35].

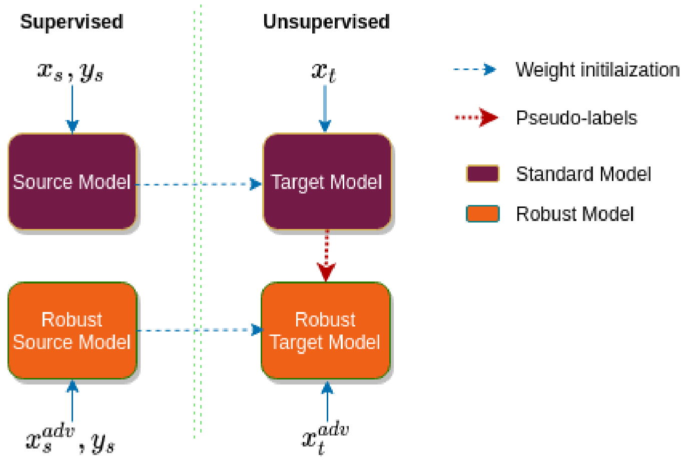


Figure 4.1: The training in the source domain uses the source data x_s, labels y_s, and adversarial examples x_s^{adv}. The training in the target domain uses the target data x_t and the adversarial examples x_t^{adv} generated using the pseudo labels.

4.2 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 by the source hypothesis transfer [35], which has shown impressive performance on the standard unsupervised domain adaptation. In this work, we extend [35] 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. We will first present our approach for training the standard model followed by the same for the robust model. The key aspects of our method are summarized in Algorithm 1.

4.2.1 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 that 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 from performing direct alignment between the two features as in [67]. 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 [35] and other prior works [71, 60]. 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 [71]. The Shannon entropy [60] for a prediction probability $\hat{p}_i$ of class i is defined as,

$$\mathcal{L}_{ent} = -\sum_i \hat{p}_i \log \hat{p}_i. \tag{4.3}$$

Unfortunately, entropy minimization can produce degenerate labels with loss converging to zero. Therefore, we take the information maximization (IM) [14] approach as adopted by [35]. 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

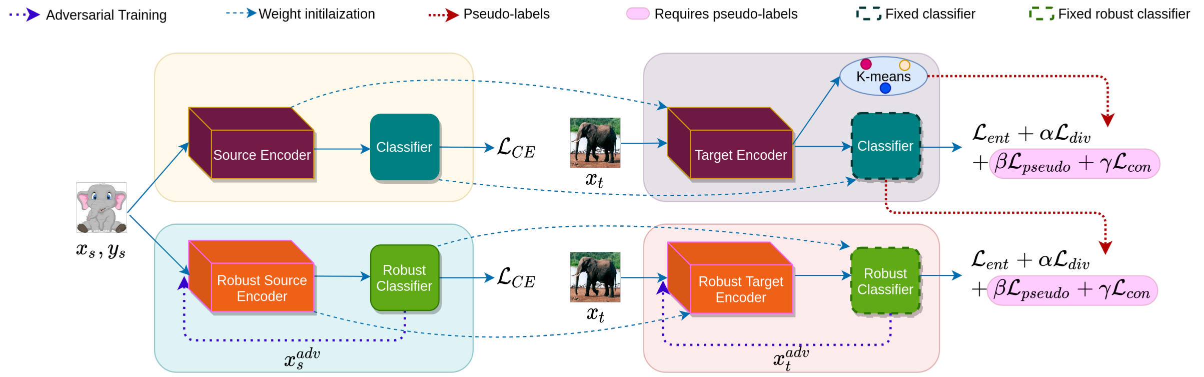


Figure 4.2: First, a standard (top-left) and a robust model (bottom-left) are trained on the source. A target encoder (top-right) is then trained by combining four losses with pseudo-labels obtained via k-means. Finally, a robust target encoder (bottom-right) is trained similarly to the standard target 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.

the class i, then the diversity loss is defined as,

$$\mathcal{L}_{div} = \sum_i q_i \log q_i. \tag{4.4}$$

Non-robust Pseudo-labels: While IM can make the model confident while ensuring diverse predictions, it may still push the output towards incorrect prediction in certain cases. In order to overcome such undesired behavior, [35] proposed to use pseudo-labels [31] 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 [35]. Let $\hat{y}$ be the pseudo-label obtained for the image x. Then, the pseudo loss is defined using the cross-entropy as,

$$\mathcal{L}_{pseudo} = \mathcal{L}_{CE}(\delta(\Phi_t(x)), \hat{y}). \tag{4.5}$$

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 [61, 23], 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,

$$\mathcal{L}_{con} = \frac{1}{2}[y \cdot D^2 + (1-y) \cdot \max(0, m-D)^2], \tag{4.6}$$

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,

$$\mathcal{L}_t(f_t; X_t, Y_t) = \mathcal{L}_{ent} + \alpha\mathcal{L}_{div} + \beta\mathcal{L}_{pseudo} + \gamma\mathcal{L}_{con}, \tag{4.7}$$

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

4.2.2 Robust Model

The idea of robustness transfer is inspired by some of the recent works [59, 68, 54] 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 [75], this process will result in 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 minimizing the cross-entropy and pair-wise contrastive losses of (4.5) and (4.6), respectively.

Adversial Target Examples: We generate adversarial examples using PGD method [41]. These generated images are used to compute the IM loss of (4.3) and (4.4). 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,

$$\mathcal{L}_t^r(f_t^r; X_t, Y_t) = \mathcal{L}_{ent}^r + \alpha\mathcal{L}_{div}^r + \beta\mathcal{L}_{pseudo}^r + \gamma\mathcal{L}_{con}^r. \tag{4.8}$$

Algorithm 1 Target adaption 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 the $\Phi_t(.)$ using Eq (4.7)
    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 the $\Phi_t^r(.)$ using Eq (4.8)
    end for
end for
```

4.2.3 Adaptation with a Single Source Model

The method previously presented suggests a model handover protocol, where the user with access to the source data provides two models. In some practical scenarios, however, both models may not be available. Under such circumstances, we still suggest to *use pseudo-labels with the proposed method for the best outcome of robust adaptation*, irrespective of the model being robust or standard. Our extensive experiments support this suggestion. 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.

Chapter 5

Experiments and Results

5.1 Data

We conduct experiments on four benchmark datasets, including a tiny, 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. 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).

- **Office-31** [51] is a standard benchmark consisting of consists of total 4,652 images from three domains - Amazon (**A**), DLSR (**D**) and Webcam (**W**) - each having 31 classes as shown in Figure A.5. We evaluate on all six combinations of source and target domain.

- **Office-home** [70] is another popular benchmark containing images 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 as shown in Figure A.6. We test all methods on all the 12 possible domain adaptation tasks.

- **PACS** [32] is a dataset to set benchmarks for domain generalization techniques. It consists 9991 images from four domains - Art (**A**), Clipart (**C**), Photo (**P**) and Sketch (**S**). Each of the domain contains images belonging to 7 classes as shown in Figure A.7.

- **VisDA-C** [49] is a challenging domain adaptation data with roughly 152k synthetic images in the source domain and about 55k real images in the target domain. Therefore, each of the 12 different classes has a significantly larger number of samples than in the other datasets. Some sample images are shown in Figure A.8.

5.2 Implementation

We use ResNet50 [18] as the backbone feature encoder for all our experiments. Moreover, we initialize it on the source with weights pre-trained on ImageNet [10]. For robust source training, we use weights obtained after adversarial training[1] 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 [49] by training our source models on synthetic images and adapting the models to the real images. All the experiments were conducted using the PyTorch framework [48].

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

5.2.1 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 4.2.3. When only the standard source 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 the 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.

5.3 Hyperparameters

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 [25] 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 [41] 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 two epochs, and the adaptation process is performed for five epochs. We train the source model for 20 epochs and run adaptation for ten epochs for all other datasets. Our methods are trained using a combination of four weighted loss terms. The diversity loss is has a weight $\alpha = 1$, the pseudo-label cross-entropy is weighted by $\beta = 0.3$ and the contrastive loss is multiplied with a factor of $\gamma = 0.2$. The values for α, β are borrowed from [35], and the sensitivity analysis for γ is shown in the following section.

5.4 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 [67, 35] that we adapt to use adversarial training in the source domain.

The first adapted approach is Adversarial Discriminative Domain Adaptation (ADDA) by [67] which, in addition to the data our approach requires, also uses source data in the adaptation phase. We perform adversarial training [41] in the source domain and follow the target adaptation protocol as described in [67].

The second method we use for comparison is Source Hypothesis Transfer (SHOT) [35]. 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 [41] in the source domain and subsequently follow the adaptation strategy as described in [35].

Most of the source-free UDA methods require an image/feature generator [33, 29, 27] which are difficult to scale while ensuring robustness on large datasets like VisDA-C [49]. Other recently introduced approaches [52, 64, 71] that are designed for pixel-level corruptions [19] do not extend well to the more complex domain adaptation tasks we present in this paper.

5.5 Architecture

Figure 5.1 shows the network architecture we use in this work. The structure remains identical in all the stages of the training in both the source and the target domain. The first block in Figure 5.1 refers to the ResNet50 backbone, which is trained on ImageNet in both standard and adversarial manner even before the source training. Following [12, 35] a feature bottleneck layer is introduced that consists of a linear (256 dimension) layer along with batch normalization. On top, we add a classification block containing

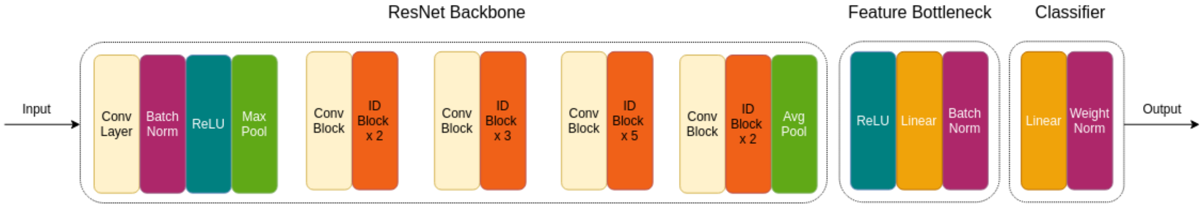


Figure 5.1: Detailed architecture of the network used in all the experiments. The network is divided into three blocks, namely, the backbone, the feature bottleneck and the classifier. The backbone together with the bottleneck forms the feature encoder.

a fully-connected layer and weight normalization as in [35] that outputs the class logits. Recall that the classification block is only trained during the source training and remains fixed during target training. However, the backbone and the bottleneck are tuned per the task. Note that even though we choose ResNet50 as the backbone, our method is not restrained by the backbone architecture.

5.6 Results

We evaluate all our methods and 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. 5.2 which shows that all our methods outperform the baselines. More detailed results on all the datasets are presented in Table 5.1.

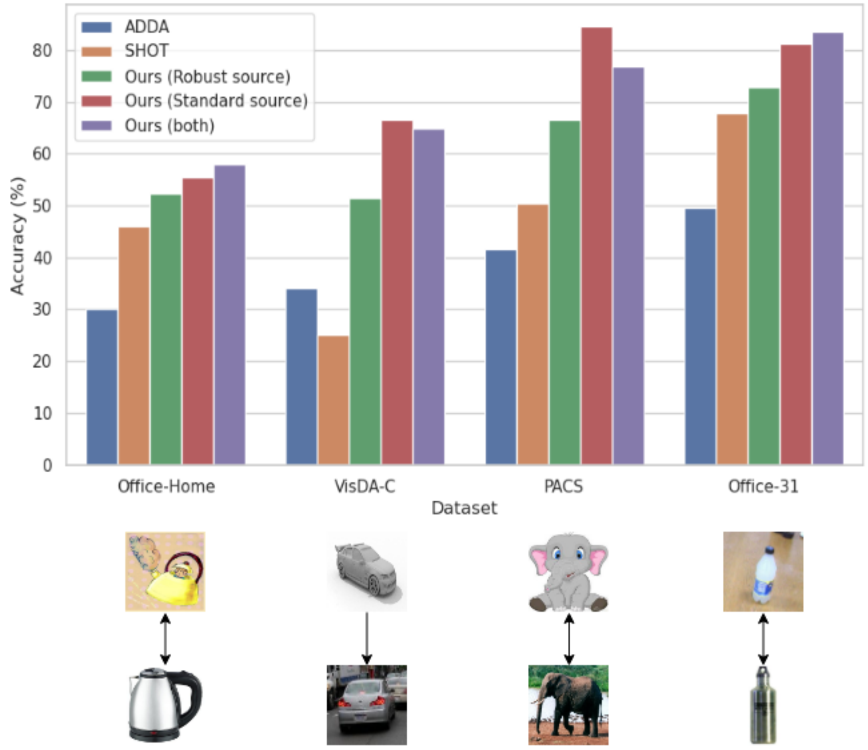


Figure 5.2: Test accuracy averaged over all domain adaptation tasks for multiple datasets. All our proposed methods show significant improvement over the baselines.

Method	**Office-31** [51]		**Office-home** [70]		**PACS** [32]		**VisDA-C** [49]	
	Adv acc	Clean acc	Adv acc	Clean acc	Adv acc	Clean acc	Adv acc	Clean acc
ADDA [67]	0.4	75.0	0.9	50.2	0.8	64.6	0.9	70.1
SHOT [35]	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 5.1: 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	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.**
ADDA robust	22.3	18.2	34.2	21.4	34.9	35.0	16.0	26.5	40.4	28.4	36.9	47.2	30.1
SHOT robust	45.6	55.3	56.2	31.9	53.3	49.7	25.5	37.7	54.2	37.0	45.5	61.6	46.1
Ours (Robust source)	51.4	60.5	60.6	38.3	55.5	57.0	33.1	47.4	59.5	**47.5**	51.1	67.0	52.4
Ours (Standard source)	50.9	69.0	60.4	42.2	63.1	60.1	40.9	46.8	63.1	41.4	53.3	73.2	55.4
Ours (Both)	**52.5**	**71.6**	**63.3**	**45.5**	**67.0**	**62.6**	**42.6**	**50.1**	**65.4**	46.9	**54.0**	**75.2**	**58.0**

Table 5.2: Adversarial accuracy of robust models on Office-home. Our methods not only beat the baselines on average but also on each of the tasks. Also, our methods perform very similar to each other compared to the baselines.

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 losing 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. A.4 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 [75].

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[2] 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 remains the same in all three cases.

Results for these adaptation tasks are illustrated in Fig. 5.3 where we compare the method using only the standard model against the method using both source models. Fig. 5.3 indicates that in the case of few classes having only a standard source model suffices for robust adaptation. However, if a larger set of classes needs to be handled, it is better to use both the standard and robust source model and follow the procedure as described in Section 5.2.1.

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

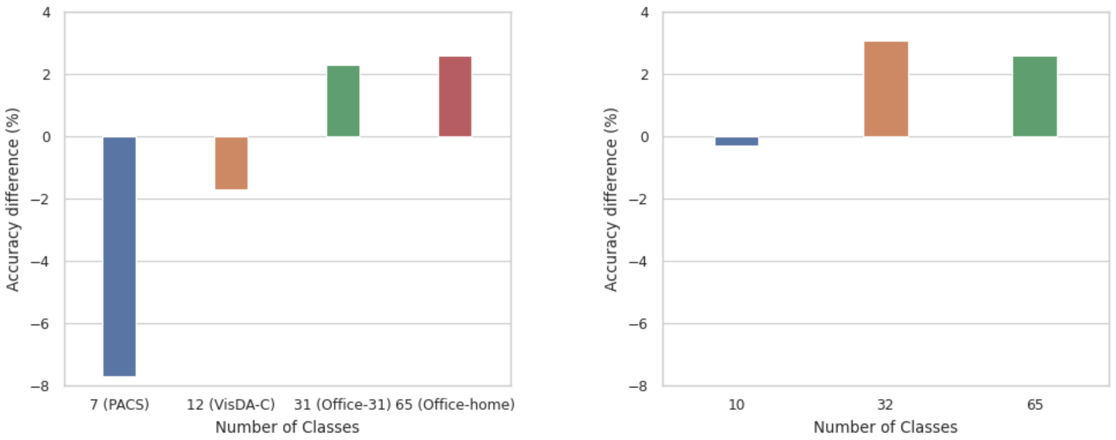


Figure 5.3: Performance of our method that use both source models relative to our method using only standard source model. The plot on the left shows the comparison on all the four dataset while on the right compares performance on Office-home by varying the number of classes.

Method	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
ADDA robust	48.0	44.7	24.8	69.2	36.7	74.0	49.6
SHOT robust	61.0	64.2	52.5	90.6	52.0	87.0	67.9
Ours (Robust source)	64.0	66.7	62.2	**95.0**	59.2	90.0	72.8
Ours (Standard source)	**80.0**	86.2	**74.5**	90.6	71.1	85.0	81.2
Ours (Both)	79.0	**88.7**	73.8	93.7	**73.6**	**92.0**	**83.5**

Table 5.3: Adversarial accuracy of robust models on Office-31. Our methods not only beat the baselines on average but also on each of the tasks. Our method performs significantly better than both ADDA and SHOT on all of the tasks.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Per-class Avg.**
ADDA robust	2.3	13.3	60.2	9.7	44.8	**32.4**	65.9	56.1	61.1	29.1	27.5	5.2	34.0
SHOT robust	48.0	26.4	23.9	26.3	33.9	5.7	33.4	10.7	16.5	15.5	47.8	11.4	25.0
Ours (Robust source)	78.3	42.7	61.1	60.0	70.2	4.4	62.1	55.4	54.9	31.1	78.0	19.0	51.4
Ours (Standard source)	**88.0**	56.9	**71.0**	**68.9**	**83.3**	0.5	**80.6**	**68.7**	**79.8**	**82.1**	**82.7**	**38.2**	**66.7**
Ours (Both)	86.1	**61.8**	69.4	67.5	82.8	1.1	76.3	67.4	75.3	79.7	80.8	31.5	65.0

Table 5.4: Adversarial accuracy of robust models on VisDA-C (Synthetic to Real). There is large variation in performance among different methods due to increased difficulty. Ours (standard source) performs the best on most of the classes.

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.**
ADDA robust	63.8	65.0	5.3	55.1	65.0	8.9	50.2	58.2	10.7	32.7	52.9	31.4	41.6
SHOT robust	73.1	91.0	43.1	53.9	74.0	66.8	60.7	49.5	35.2	18.0	27.5	12.9	50.5
Ours (Robust source)	90.6	**94.3**	68.6	72.7	84.1	92.6	77.6	87.2	68.8	20.7	33.7	6.9	66.5
Ours (Standard source)	**92.3**	94.0	93.1	79.3	**94.0**	91.2	**81.0**	91.9	70.6	**81.0**	**91.7**	**55.4**	**84.6**
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 5.5: Adversarial accuracy of robust models on PACS. The performance among three of our methods is close to each on relatively simpler (based on highest and lowest accuracy) tasks (e.g. A → C, A → P) but result into significantly different numbers on harder tasks (e.g. S → A, S → P).

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	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 5.6: **Robust** Accuracy of the Robust Model on PACS. Note that the contrastive loss term, entropy term and diversity loss term were removed from both the target models while CE was only removed from the target robust model since removing from both will make it very hard to adapt.

γ	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.**
0.2	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
0.3	92.5	93.7	94.0	**77.6**	**91.6**	94.0	**82.2**	93.6	72.0	23.4	82.9	20.4	76.5
0.5	**93.4**	**94.6**	94.4	**77.6**	91.0	93.9	81.5	93.0	**81.0**	16.1	79.1	**21.0**	76.4

Table 5.7: Sensitivity with respect to hyper-parameter γ on PACS dataset. We find that by modifying the weight of the contrastive loss the performance of our method does not change drastically.

5.7 Ablation Study

We study the impact of each of the components in our model on the PACS dataset in Table 5.6. 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 [69] for dimensionality reduction. Fig. 5.4a 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. A.2a. In the following setting, we use pseudo-labels generated from a robust target model instead. Fig. A.2b 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. A.2c. Fig. 5.4 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.

5.8 Hyper-parameter sensitivity

We perform the sensitivity analysis of our method with respect to γ by evaluating the test accuracy on adversarial images. We conduct the experiment for all six domain adaptation tasks on the Office-31 dataset. Table 5.8 shows the variation in accuracy for different values of γ. We conduct another experiment on a relatively large dataset (PACS) with even more variation on the values of γ. Table 5.7 shows that, on average, the adversarial accuracy is not impacted much by the variation in γ. Both the experiments empirically verify that changing the weight of the contrastive loss does not have a significant impact on the robust average accuracy. Therefore, our method is robust against changes of hyper-parameters, making the transfer between datasets easy.

γ	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
0.1	78.0	83.6	72.5	**95.0**	73.0	90.0	82.0
0.2	79.0	88.7	73.8	93.7	73.6	**92.0**	83.5
0.3	**81.0**	**89.3**	**74.5**	**95.0**	**73.9**	**92.0**	84.3

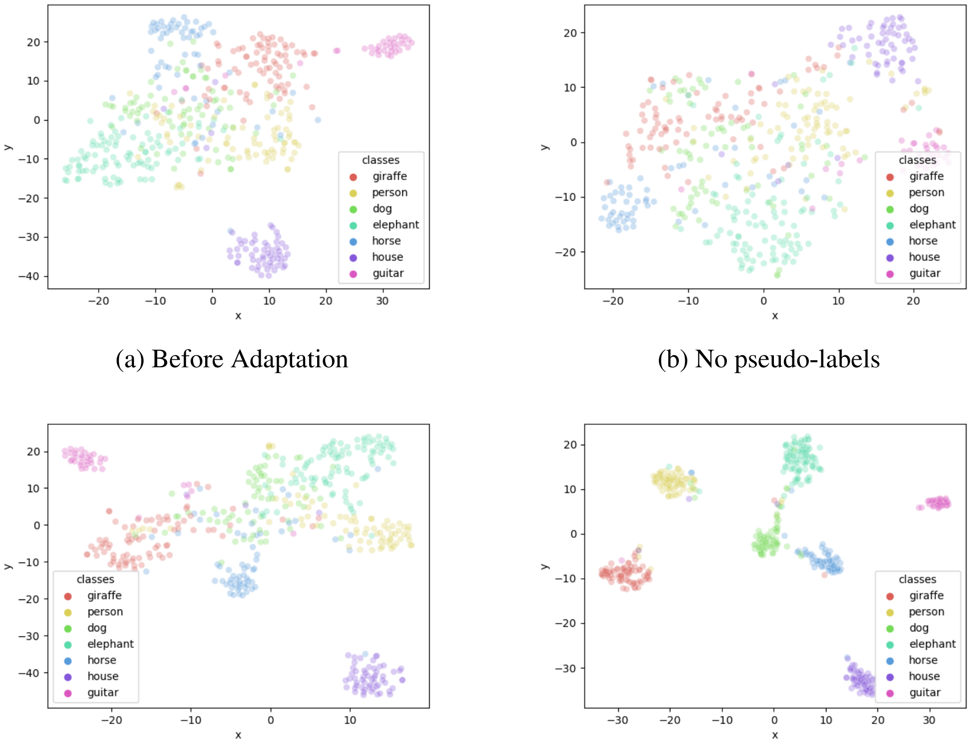
Table 5.8: Sensitivity with respect to hyper-parameter γ on Office-31 dataset. The table indicates that in different weights to the contrastive loss can result in very similar performance.

Pseudo-label accuracy	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
50%	69.0	76.1	64.7	88.1	68.6	89.0	75.9
70%	76.0	87.4	76.4	93.1	72.5	89.0	82.4
90%	**79.0**	**91.8**	**79.3**	**97.5**	**80.0**	**93.0**	86.8

Table 5.9: Impact of pseudo-label quality on Office-31. We can clearly observe that the “better" the pseudo-labels, the higher the accuracy on each of the tasks.

5.9 Influence of Pseudo-labels Quality

We emphasize that pseudo-labels play a key role among other aspects in the proposed method. To further analyze the impact of pseudo-labels on the performance of our method, we conduct additional experiments. Recall that during the robust training on the target domain, we obtain pseudo-labels from the standard target model, which was trained in the previous step. To study their influence, we use the ground truth labels and create different sets of pseudo-labels by varying their accuracy with respect to the ground truth. We conduct the experiments on Office-31 datasets on all six domain adaptation tasks. Table 5.9 shows the results for 50%, 70% and 90% accurate pseudo-labels. Recall that pseudo-labels are required for three purposes, including contrastive loss, pseudo-label loss, and the generation of adversarial examples, each of with plays an important role. Thus, as we would expect, the accuracy improves with the quality of the pseudo-labels. Note that there is not only significant improvement in the average accuracy but also on each of the six tasks consistently, as shown in Table 5.9.



(a) Before Adaptation

(b) No pseudo-labels

(c) Ours (robust source)

(d) Ours (standard source)

Figure 5.4: 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.

Chapter 6

Discussion

In this project, we propose a new approach for transferring robustness in domain adaptation task that does not require access to source data while adapting to any target domain. It is one of the first works on this problem to the best of our knowledge. The motivation is to learn a robust model even in the absence of labeled data and achieve reasonable accuracy on clean samples. To achieve this goal, we train four models, two in the source domain and two in the target.

Our experiments demonstrate that a simple method efficiently achieves sufficient accuracy on both the clean and the adversarial samples in the target domain. Our method also performs significantly better than previous approaches (adapted to be robust) for domain adaptation, including adversarial methods that require both the source and target images simultaneously in the adaptation phase. Generating the pseudo-labels is one of the core components of our method for which we implement a popular unsupervised learning technique, k-means clustering. These pseudo-labels are further utilized to guide the training of the robust target model. It is important to note that clean accuracy drop as one tries to increase the robust accuracy. Thus, to extract “better" pseudo-labels, we train an additional model on the target domain on the clean samples, which requires a corresponding model in the source domain.

Another important aspect of our technique is the presence of different loss components. While the source models are trained by back-propagating the gradients of the standard cross-entropy loss, the target training involves more than one type of loss. Minimizing the entropy loss pushes the model to be more confident about the predicted label and has been widely used in the literature. The diversity term ensures that the model does not get trapped into the trivial case of degenerate prediction. Also, we have a contrastive loss which is popular among unsupervised learning techniques. It operates directly in the feature space by pulling similar features together and pushing away dissimilar ones with pseudo-labels. We also use the Cross-entropy loss for target training but instead use pseudo-labels since the ground truth is unknown. Each of the loss terms has its significance and, when combined, results in efficient model adaptation to the target domain in the absence of source images.

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

Do robust models transfer robustly?–Yes. Besides the proposed method, our baselines also allow us to 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 is noteworthy. This observation is in accordance with the existing works [59, 68, 54], although in different settings. In this works' setting, 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. 4.2.2 and 4.2.3 for more details. It goes without saying, non-robust pseudo-labels are preferred when non-robust source models are available.

Which model to transfer?–Robust. Provided a good transfer of non-robust models to the target, it has been observed that robustness can be achieved by generating the adversarial examples in the target. Such robustness, however, entirely relies on the pseudo-labels alone. We observed that for datasets with few classes, such transfer is often is not a problem. However, as the number of classes increases, the transfer of non-robust models followed by robust training is not a good idea. Please, refer to Fig. 5.3 for robust and non-robust models transfer for an increasing number of classes. Such behavior can be attributed to the following: as the number of classes increases, the chance 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 transferring the non-robust source model followed by robust target training (using the method proposed in this work).

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. 4.2.3. This can be observed in Tab. 5.6 and 5.1. Note that the baseline SHOT robust differs from our method with robust source in terms of the contrastive feature learning. Please, refer Sec. 4.2.1 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 using two models as presented in Fig. 4.1 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. 5.1. The non-robust model is only used to generate more reliable pseudo-labels for adversarial examples during robust training in the target domain.

A crucial component of the method presented in this work is the initialization of model weights. All the target model weights are initialized with their pre-trained source counterpart, and the source model backbone is not initialized randomly as in most machine learning training algorithms. In the light of relatively small data samples, the ResNet backbone is initialized with a model pre-trained on ImageNet. The large size and diversity of ImageNet proves to be a better initialization than random weights and has been followed by most of the work on non-robust training in the past. However, for a robust source model, one needs to initialize with ResNet trained robustly on ImageNet. For our work, we use[1] a pre-trained robust ResNet50. For non-standard architecture, performing adversarial training [41] on ImageNet can prove to be expensive both in terms of computing power and time required for training. Our method requires two source and two target models almost doubling the number of trainable parameters compared to the previous approaches. Moreover, the target model training is sequential, that is, the robust target model can be only be trained after the standard target model has been trained.

Our experiments on the multiple datasets each have different domains, varying classes and overall dataset size signifies the versatility of our method. Moreover, unlike most of the previous work, which utilizes the entire target for training and predicts on the same samples, we instead divide the data into multiple splits randomly and report the accuracy on the images which were neither used for training nor for model selection.

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

Chapter 7

Conclusion

We proposed a method to perform unsupervised robust domain adaptation without access to source data in this work. Given the growing privacy concerns around data, our method is more suitable for real-life domain adaptation problems. We introduce a framework consisting of two models in the source domain and two in the target domain. We first adapt the non-robust source model to the target domain. Then, we adapt the robust source model to the target domain using pseudo-labels from the non-robust target model. The high accuracy of the pseudo-labels leads to improved robustness of the robust target model. 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. We perform extensive experiments on four benchmark datasets, and in all three cases, we obtained very promising results, thanks to the proposed method. Our extensive study shows that transferring both robust and standard models is often the best choice for 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.

7.1 Future Work

Our work provides some promising directions for future research. One can study the benefit of source data on robust accuracy by performing domain adaptation with source data. The proposed model relies on both robust and non-robust training separately. Besides, one may explore single-source models that perform both robust and non-robust predictions in a multitasking fashion. This will avoid sharing two models trained on the source data. Even though we evaluate the performance on image classification, the idea is generic. It could be extended to image segmentation and even to other domains such as natural language text and audio.

Another line of work can be in the direction of adversarial robustness. While our currently focuses on attacks under l_∞ norm, one could extent the threat model to l_1 and l_2. We can also verify if the target trained to be robust under one threat model can automatically robust to other threat models. Similarly, we evaluated robustness via Projected Gradient Descent (PGD) [41] but could be extended to other white-box and box-attacks and defense strategies.

To sum up, we address a challenging problem that is unique in its own way. We hope to draw the community's attention towards these new dimensions and encourage others to find more efficient and elegant solutions to the problem.

Appendix A

Additional Results

A.1 Feature Visualization

To understand the impact of pseudo-labels, we plot the encoder features with PCA's help, followed by t-SNE. Figure A.1 shows the adversarial images under three different scenarios. It clearly shows that cluster formation gets progressively better as we introduce pseudo-labels and switch from robust target model to standard target model for obtaining those pseudo-labels. We also plot the same scenarios but for clean samples instead in Figure A.2. Thus, pseudo-labels not only help in improving the accuracy but also provide better clustering.

A.2 Contrastive Loss

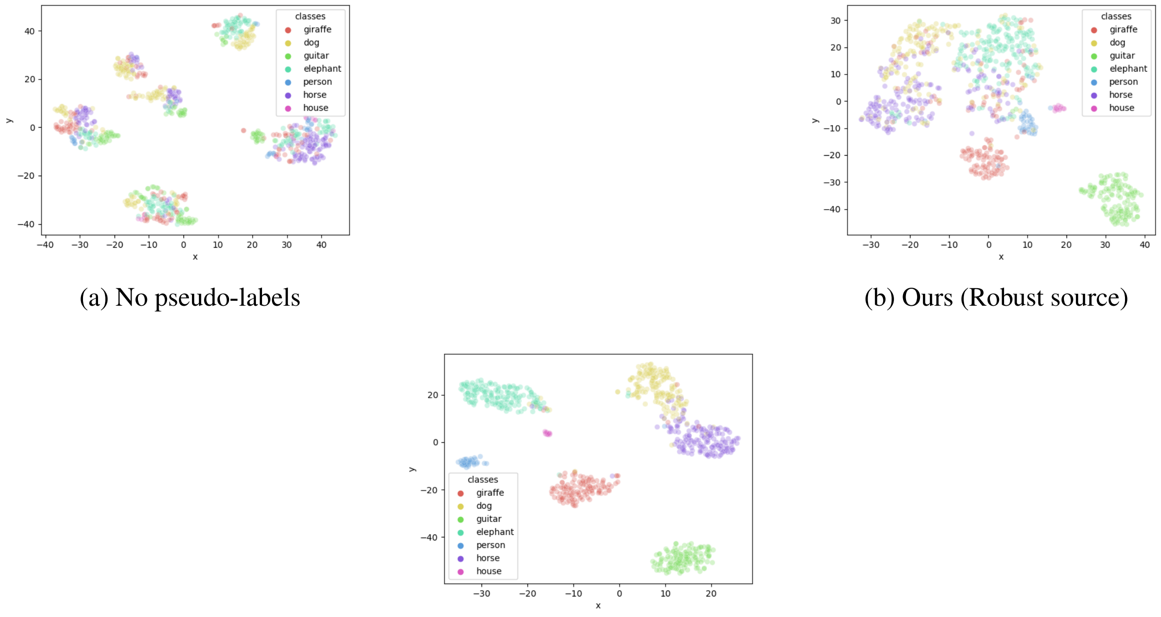
We make use of the contrastive loss for both standard and robust target training. Figure A.3 shows a toy-example of the impact of contrastive loss. The circle positions represent images in the feature space, and the color represents its pseudo-label. We illustrate the pair selection in the feature space with the help of dotted lines. The idea of the contrastive loss is to pull together pair of points that belong to the same class and push apart others. In Figure A.3 we show the change in relative distance of three pairs before (left) and after (right), minimizing the contrastive loss. Note that unlike other losses (e.g., Triplet loss), the contrastive loss does not require the additional cost of finding a suitable anchor. We apply contrastive loss on all possible pairs in a batch of 64 images simultaneously in our implementation.

A.3 Clean Accuracies

We evaluate each of the methods on clean samples and report the accuracies in Tables A.1 to A.4. We present the case of sub-setting only ten classes from the Office-home dataset in Table A.5. It affirms our hypothesis that only a standard source model is sufficient to transfer robustly on the target if we have few classes in total.

A.4 Non-robust Models

We also compare the performance of our method with both the non-robust baseline. Table A.6 to Table A.9 report the accuracy of the methods on clean images for all the four datasets. Similarly, we also evaluate the robust accuracy for the baselines and contrast them with our methods. The performance on adversarial images are reported from Table A.10 to Table A.13



(a) No pseudo-labels

(b) Ours (Robust source)

(c) Ours (standard source)

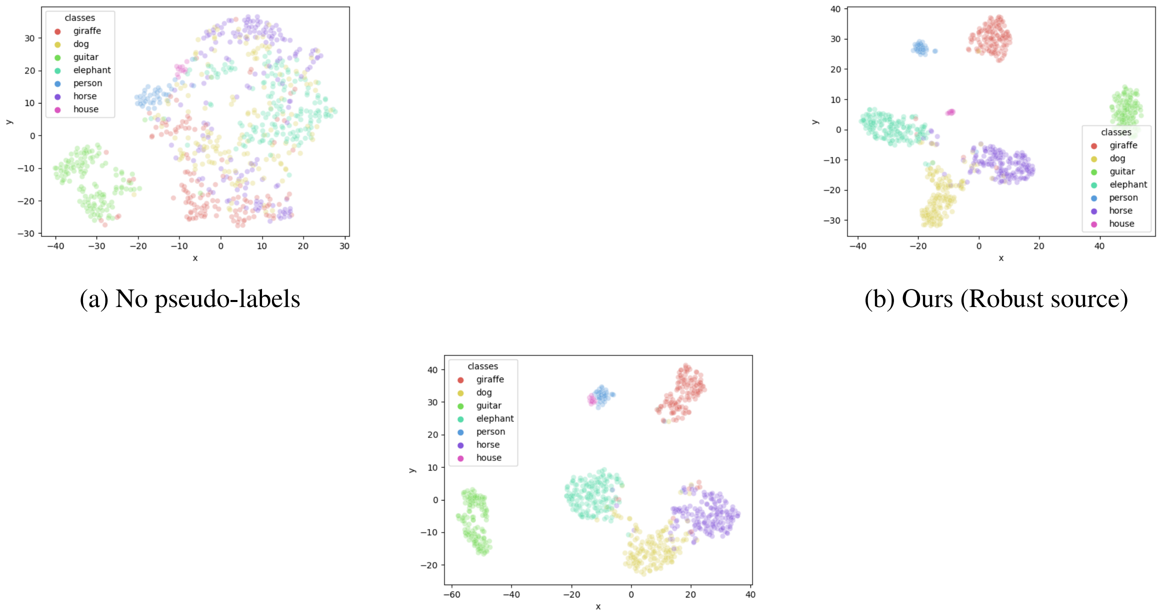
Figure A.1: Impact of pseudo-labels on PACS with Cartoon as the source and Sketch as the target domain for **adversarial images**. The plot clear shows the importance of pseudo-labels. (a) Without the use of pseudo-labels the clusters are all mixed. (b) With pseudo-labels from robust model, some clusters tend to overlap. (c) With pseudo-labels from standard source model, well separated clusters are formed.

Method	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
ADDA robust	60.0	59.7	28.9	74.2	40.8	82.0	57.6
SHOT robust	66.0	68.6	57.4	93.1	59.4	**94.0**	73.1
Ours (Robust source)	70.0	68.6	67.0	**95.6**	64.4	93.0	76.4
Ours (Standard source)	82.0	91.8	**78.4**	95.0	76.1	91.0	85.7
Ours (Both)	**84.0**	**92.5**	78.0	**95.6**	**77.7**	**94.0**	**87.0**

Table A.1: Clean accuracy of robust models on Office-31. Our methods beats the baselines not just on average but also on most of the tasks.

A.5 Samples Images

We present some sample images from each domain for each of the datasets used in this study. Images from Office, Office-home, PACS, and VisDA-C are shown in Figures A.5, A.6, A.7 and A.8, respectively.



(a) No pseudo-labels

(b) Ours (Robust source)

(c) Ours (Standard source)

Figure A.2: Impact of pseudo-labels on PACS with Cartoon as the source and Sketch as the target domain for **clean images**. (a) Without the use of pseudo-labels, features of different classes tend to overlap. (b) With pseudo-labels, points features tend to cluster with little difference with respect to the model being robust or not.

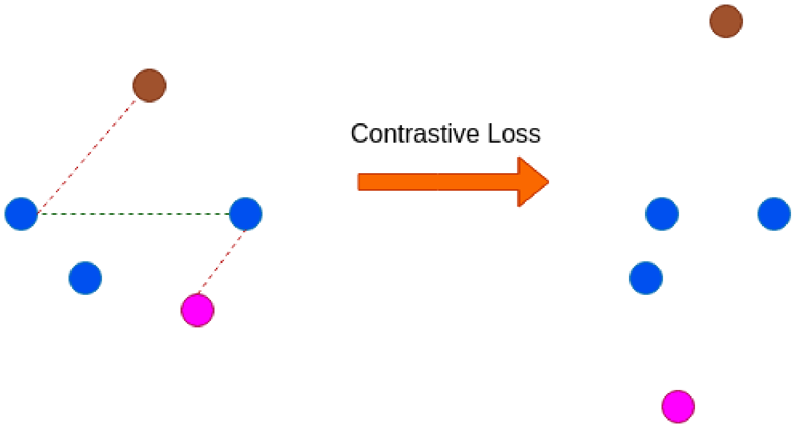


Figure A.3: Depiction of contrastive loss in the feature space. On the left, the positive (same class) and negative (different class) pair of images are connected with a green and a red dotted line, respectively. The impact of the loss is shown on the right.

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.**
ADDA robust	26.8	23.1	39.3	30.2	41.9	44.2	24.9	31.7	50.1	42.0	43.4	54.5	37.7
SHOT robust	48.2	61.5	64.0	38.7	57.8	59.3	34.6	42.2	62.7	51.9	49.7	66.8	53.1
Ours (Robust source)	54.6	65.3	69.5	50.4	59.8	65.3	42.8	51.1	69.3	57.0	54.3	71.4	59.2
Ours (Standard source)	53.5	73.6	71.6	52.1	69.3	69.5	52.3	50.6	73.7	51.6	56.9	78.2	62.7
Ours (Both)	**54.9**	**74.9**	**73.1**	**56.4**	**72.1**	**71.9**	**54.1**	**53.4**	**75.2**	**57.4**	**58.5**	**79.8**	**65.1**

Table A.2: Clean accuracy of robust models on Office-home. Ours (both) method beats all others not just on average but on all the tasks.

APPENDIX A. ADDITIONAL RESULTS

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.**
ADDA robust	72.7	79.6	37.8	74.1	77.2	51.4	65.6	65.9	50.8	44.4	58.8	39.5	59.8
SHOT robust	78.3	95.2	46.6	68.0	82.6	70.5	73.2	57.6	37.3	26.3	33.9	18.0	57.3
Ours (Robust source)	94.7	**99.1**	69.5	84.4	94.3	95.0	91.7	93.0	70.0	26.6	42.2	12.0	72.7
Ours (Standard source)	95.1	97.9	94.1	**92.0**	**97.9**	92.1	92.4	94.7	71.5	**92.4**	**95.1**	**58.1**	**89.4**
Ours (Both)	**95.9**	**99.1**	**96.2**	88.8	**97.9**	**96.2**	**93.2**	**96.2**	**72.1**	45.9	91.0	30.2	83.6

Table A.3: Clean accuracy of robust models on PACS. Similar to adversarial accuracy (see Table 5.5) we find that ours (standard source) significantly outperforms others on relatively difficult tasks (e.g. S → A, S → P).

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Per-class Avg.**
ADDA robust	2.9	26.4	77.2	24.3	59.1	**42.3**	78.6	66.8	72.3	35.8	48.6	21.0	46.3
SHOT robust	59.5	36.7	33.8	35.9	44.6	6.4	47.0	18.3	25.1	26.3	59.5	18.7	34.3
Ours (Robust source)	88.0	59.4	74.0	71.3	83.5	6.0	77.1	69.8	67.1	47.7	85.0	34.8	63.6
Ours (Standard source)	**93.9**	70.5	**81.8**	**78.6**	91.4	1.4	**90.3**	**78.0**	**89.9**	**91.4**	**90.0**	**52.8**	**75.8**
Ours (Both)	92.5	**75.3**	79.9	77.5	**91.5**	3.0	87.9	77.0	87.5	89.2	89.5	47.5	74.9

Table A.4: Clean accuracy of robust models on VisDA. Performances of ours (Standard source) and ours (Both) are very close though ours (Standard source) does better on average.

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.**
ADDA robust	36.8	32.8	54.2	48.6	52.7	71.1	35.2	47.7	72.3	63.8	65.2	70.2	54.2
SHOT robust	69.0	82.4	91.6	61.9	75.6	91.0	52.4	61.3	88.0	66.7	65.2	87.0	74.3
Ours (Robust source)	89.0	87.8	**95.2**	74.3	90.8	**95.8**	69.5	79.4	93.4	75.2	85.8	93.1	85.8
Ours (Standard source)	89.0	93.1	**95.2**	**78.1**	91.6	95.2	**78.1**	**91.6**	**95.2**	76.2	**91.6**	93.1	**89.0**
Ours (Both)	**91.0**	**95.4**	**95.2**	74.3	**94.7**	94.6	74.3	87.1	94.6	**77.1**	**91.6**	**94.7**	88.7

Table A.5: Adversarial accuracy of robust models on Office-home with only **10 classes**. Unlike the data with all 65 classes, here ours (standard source) better on average than ours (both) and is comparable on most tasks.

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.**
ADDA	78.7	86.2	41.0	85.1	82.6	44.3	75.1	72.5	44.7	**55.4**	66.5	**42.8**	64.6
SHOT	74.0	98.5	41.6	83.2	94.3	46.4	79.5	64.4	32.4	28.3	22.6	18.9	57.0
Ours (Both)	**95.9**	**99.1**	**96.2**	**88.8**	**97.9**	**96.2**	**93.2**	**96.2**	**72.1**	45.9	**91.0**	30.2	**83.6**

Table A.6: Clean accuracy of the standard model on PACS.

Method	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
ADDA	78.0	82.4	49.8	83.6	59.2	97.0	75.0
SHOT	85.0	**88.7**	79.4	**96.2**	**77.3**	**99.0**	**87.6**
Ours (Both)	**79.0**	**88.7**	73.8	93.7	73.6	92.0	83.5

Table A.7: Clean accuracy of the standard model on Office-31.

Method	A → C	A → P	A → R	C → A	C → P	C → R	P → A	P → C	P → R	R → A	R → C	R → P	**Avg.**
ADDA	35.9	41.6	59.9	42.6	49.9	58.5	39.9	39.6	69.0	53.5	43.3	68.9	50.2
SHOT	50.4	**77.0**	**78.1**	**57.6**	71.5	**73.2**	**58.2**	47.7	**78.7**	**64.2**	53.0	**80.4**	**65.8**
Ours (Both)	**54.9**	74.9	73.1	56.4	**72.1**	71.9	54.1	**53.4**	75.2	57.4	**58.5**	79.8	65.1

Table A.8: Clean accuracy of the standard model on Office-home.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Avg.**
ADDA	93.6	72.9	74.6	49.6	90.3	**23.7**	86.6	76.6	80.9	84.8	83.8	23.3	70.1
SHOT	**96.6**	**87.4**	**83.4**	73.7	**95.5**	3.7	**89.3**	**81.0**	**95.4**	88.3	**93.1**	**60.8**	**79.0**
Ours (Both)	92.5	75.3	79.9	**77.5**	91.5	3.0	87.9	77.0	87.5	**89.2**	89.5	47.5	74.9

Table A.9: Clean accuracy of the standard model on VisDA.

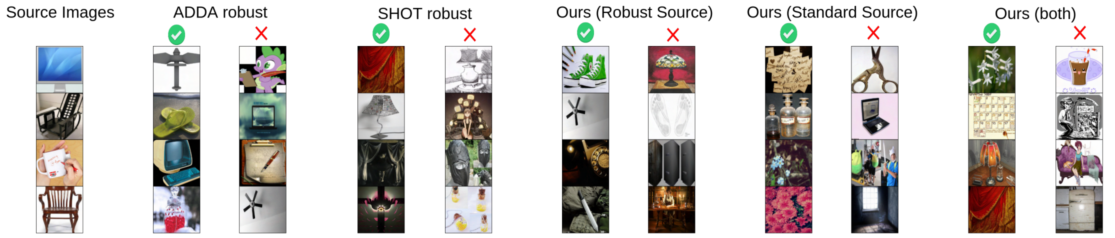


Figure A.4: Sample adversarial images in the target (Art) domain of Office-home. The figure shows the correctly classified and misclassified images by the target model for each of the method. The source model was trained on Real-world (Rw) images.

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.**
ADDA	0.8	1.5	1.9	0.0	1.4	0.7	0.2	0.9	0.9	0.2	1.0	1.6	0.9
SHOT	0.2	0.5	0.5	0.0	0.2	0.5	0.2	0.1	0.3	0.4	0.1	0.7	0.3
Ours (Both)	52.5	71.6	63.3	45.5	67.0	62.6	42.6	50.1	65.4	46.9	54.0	75.2	58.0

Table A.10: Robust accuracy of the standard model on Office-home.

Method	A → D	A → W	D → A	D → W	W → A	W → D	**Avg.**
ADDA	0.0	0.0	0.9	1.3	0.0	0.0	0.4
SHOT	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ours (Both)	79.0	88.7	73.8	93.7	73.6	92.0	83.5

Table A.11: Robust accuracy of the standard model on Office-31.

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.**
ADDA	3.6	0.0	0.0	0.0	0.3	0.0	0.5	4.9	0.0	0.0	0.9	0.0	0.8
SHOT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
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 A.12: Robust accuracy of the standard model on PACS.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	**Avg.**
ADDA	0.0	0.7	2.5	0.5	5.2	1.1	0.0	0.0	0.0	0.0	0.8	0.0	0.9
SHOT	0.0	1.7	0.4	0.0	0.2	0.0	0.0	0.1	0.7	0.0	0.1	0.0	0.3
Ours (Both)	86.1	61.8	69.4	67.5	82.8	1.1	76.3	67.4	75.3	79.7	80.8	31.5	65.0

Table A.13: Robust accuracy of the standard model on VisDA.

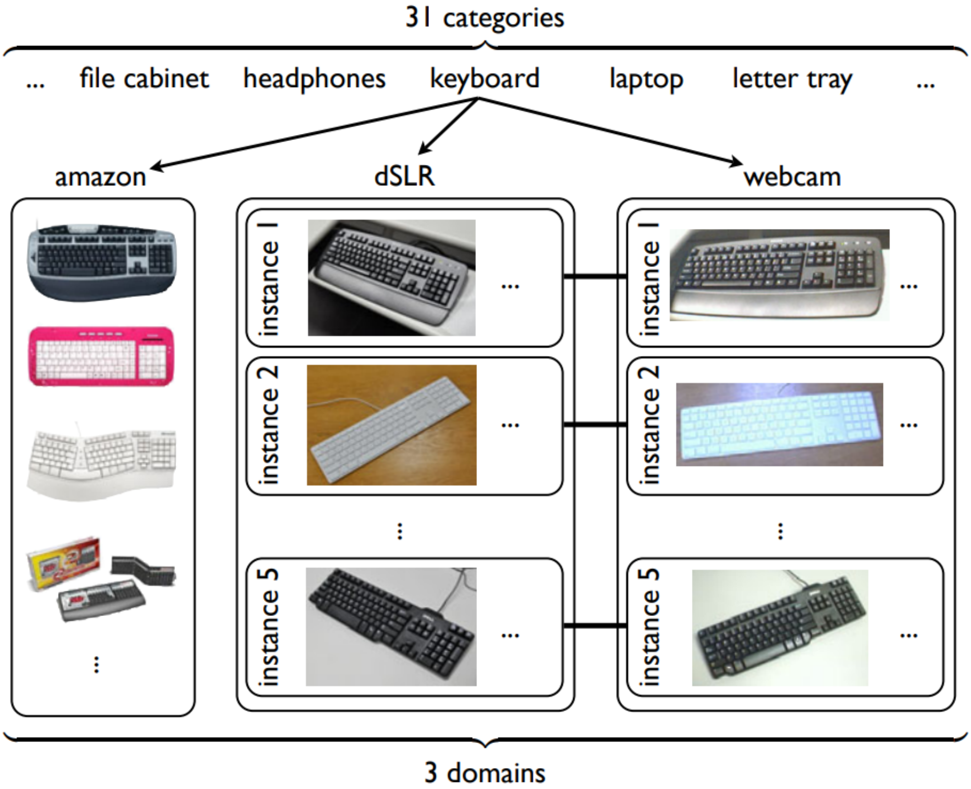


Figure A.5: Sample images from the Office [51] dataset. Image courtesy [51].

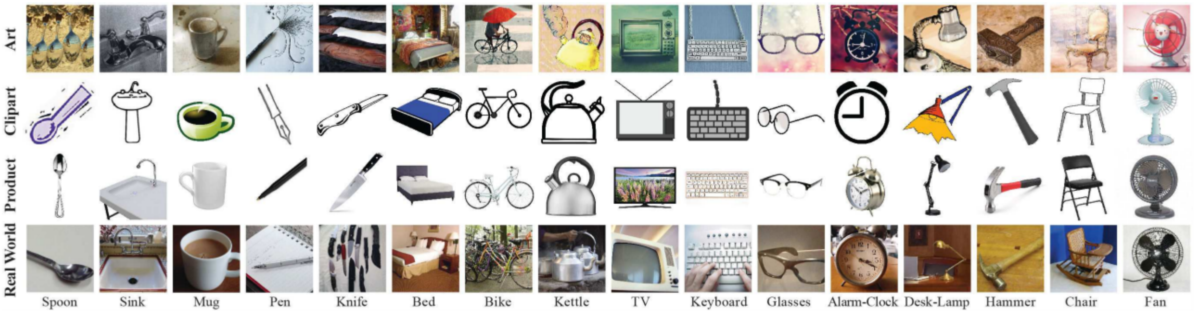


Figure A.6: Sample images from the Office-home [70] dataset. Image courtesy [70].

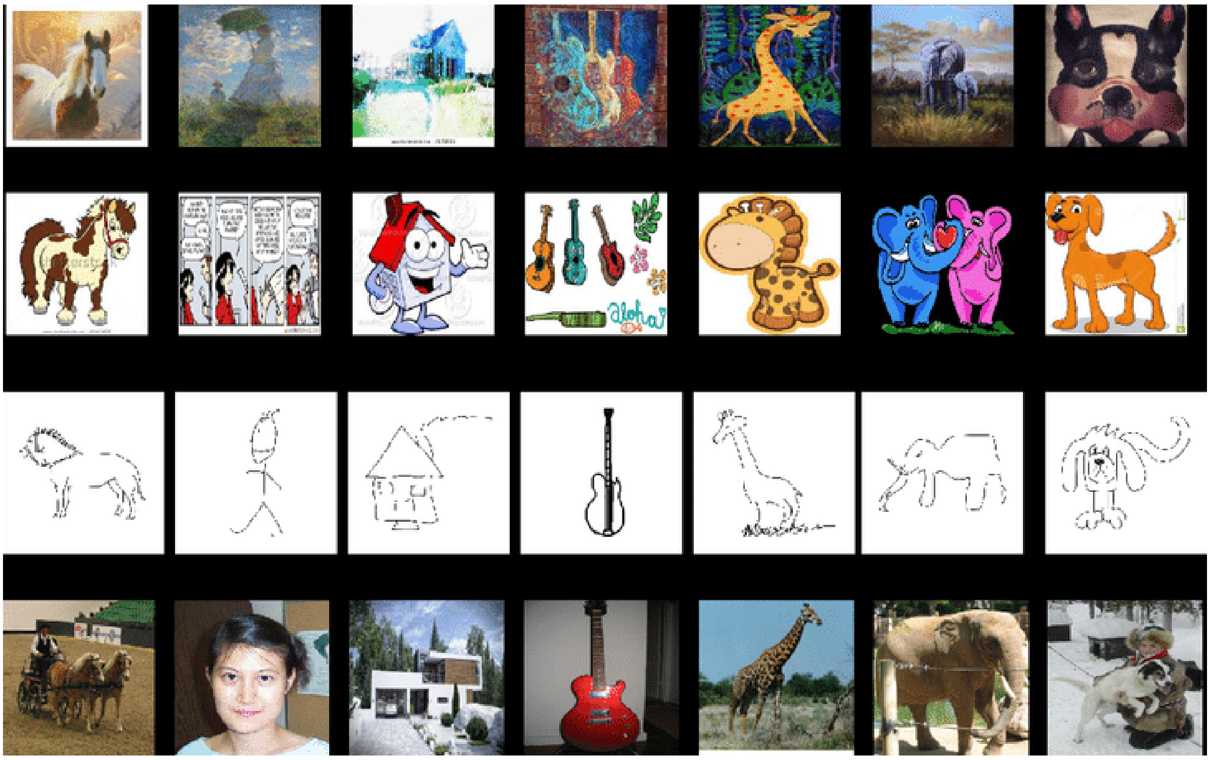


Figure A.7: Sample images from the PACS [32] dataset. Image courtesy [32].

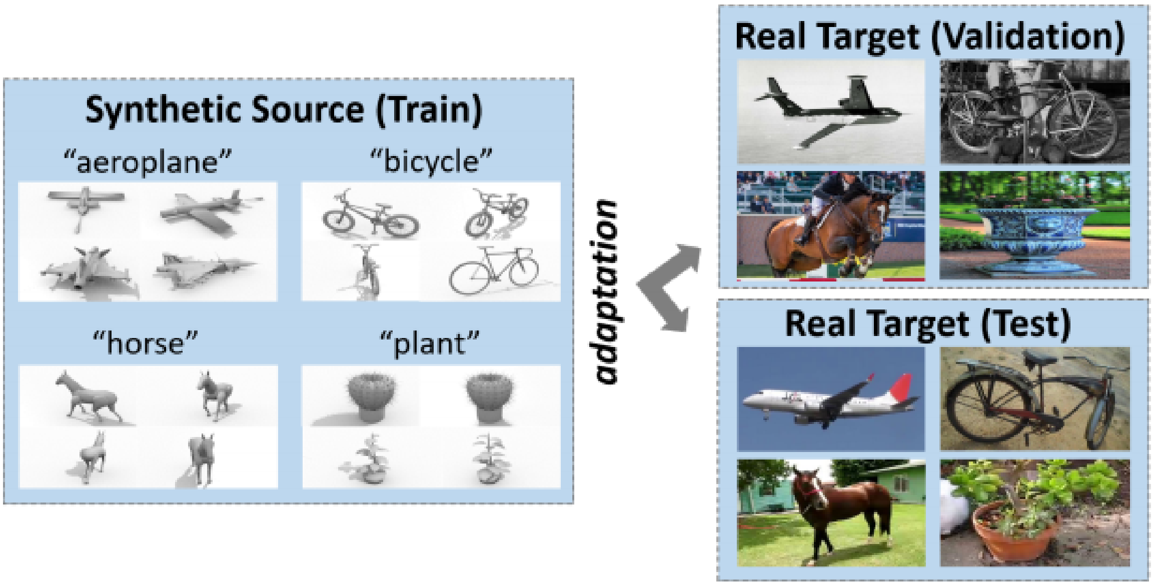


Figure A.8: Sample images from the VisDA [49] dataset. Image courtesy [49].

Bibliography

[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] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013.

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

[4] 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.

[5] 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.

[6] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[7] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.

[8] 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.

[9] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019.

[10] 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.

[11] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 994–1003, 2018.

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

[13] Shayan Gharib. Unsupervised domain adaptation for audio classification. *Machine learning*, 2020.

[14] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. 2010.

[15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

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

[17] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006.

[18] 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.

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

[20] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[21] 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.

[22] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

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

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

[26] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009.

[27] 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.

[28] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[29] 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.

[30] Yann Lecun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

[31] 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.

[32] 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.

[33] 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.

[34] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[35] 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.

[36] 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.

[37] 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.

[38] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017.

[39] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

[40] 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.

[41] 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.

[42] Harsh Maheshwari. Understanding domain adaptation. *URL: https://levelup.gitconnected.com/understanding-domain-adaptation-63b3bb89436f*.

[43] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[44] 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.

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

[51] 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.

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

[53] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

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

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

[56] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[57] 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.

[58] 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.

[59] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019.

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

[61] 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.

[62] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[63] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[64] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. 2019.

[65] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[66] 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.

[67] 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.

[68] Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020.

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

[70] 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.

[71] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[72] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable normalization: Towards improving transferability of deep neural networks. 2019.

[73] 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.

[74] 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.

[75] 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.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

UNSUPERVISED ROBUST DOMAIN ADAPTATION WITHOUT SOURCE DATA

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
AGARWAL	PESHAL

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 12.03.2021	Peshal

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.