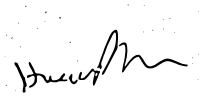
APPROVAL SHEET

Title of Thesis: Curation Bias in Domain Adaptation

Name of Candidate: Ajinkya Baban Tejankar
Master of Science, 2020

Thesis and Abstract Approved: 

Hamed Pirsiavash
Assistant Professor
Department of Computer Science and Electrical Engineering

Date Approved: 08/12/2020

NOTE: *The Approval Sheet with the original signature must accompany the thesis or dissertation. No terminal punctuation is to be used.

ABSTRACT

Title of thesis: Curation Bias in Domain Adaptation
Master's Thesis

Ajinkya Baban Tejankar, Master of Science, 2020

Thesis directed by: Professor Hamed Pirsiavash
Department of Computer Science and
Electrical Engineering

Domain adaptation is an important problem with many practical applications. The goal is to adapt a model trained on one domain (source) to another domain (target) with scarce or no annotation. We observe that the unlabeled target datasets of popular domain adaptation benchmarks do not contain any categories apart from testing categories. We believe this introduces a bias that does not exist in many practical applications. We note that this bias can be reduced easily by appending the datasets with images from non-testing categories. On these modified benchmarks, state-of-the-art domain adaptation methods show a large drop in performance. Thus, raising concerns about their practical applicability. Further, we show that a simple, two-stage method involving self-supervised task of rotation prediction and knowledge distillation is a competitive baseline.

Curation Bias in Domain Adaptation

by

Ajinkya Baban Tejankar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Science
2020

Advisory Committee:
Professor Hamed Pirsiavash, Chair/Advisor
Professor Francis Ferraro
Professor Tim Oates

Acknowledgments

I express my gratitude to my advisor, Dr. Pirsiavash, for having faith in me and providing me with an opportunity to do good science.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Consider the following problem: we have an image classification model trained on synthetic images (source domain) and we want to use this model to classify real images (target domain) to the same categories. Since the model has not seen any real images and is highly tuned to synthetic images, it does not perform well on real images. This difference in performance between source and target domain is called domain gap, and the technique of adapting the classifier to overcome this domain gap is called domain adaptation (DA).

In domain adaptation, we assume that the training has access to a large scale unlabeled data and just a small or even no annotated data from the target domain. The goal is to adapt a model to close the domain gap by performing well on the target domain. In general, it is assumed that the set of categories in the target domain are the same as those in the source domain. Refer to Figure 1.1 for a typical domain adaptation task setting.

Motivation. Recently, unsupervised domain adaptation (UDA) has become a hot research topic due to its various applications in real world where it is difficult and costly to annotate images in specific application domains. For instance, assume

a customer buys a household robot for which the visual perception system is trained at the factory. However, the appearance of objects in the customer's house may be different from the training data due to lighting and other instance specific variations. This will result in degraded robot vision. Hence, the robot's visual perception can be improved by collecting some unlabeled data from the customer's house and then adapting the model to this new domain using domain adaptation algorithms.

As another example, reinforcement learning (RL) has recently shown a lot of promise in various applications. However, most RL methods require lots of trials which are not possible in the real world due to physical limitations. Hence, most RL methods are being trained on graphics simulators and then tested in the real world. Clearly, the difference between synthetic and real data will lead to a domain gap in this setting. Consequently, domain adaptation can be used to reduce the domain gap.

Uncurated Unlabeled Data. As illustrated in above examples, we believe that using unlabeled data for domain adaptation is a practical setting. It can be used in many applications where collection of unlabeled data in the target domain is easy and almost for free. In such settings, there is no effort required for data annotation, so the data can be from any category. For instance, we train a classifier on the synthetic data for n categories and want to adapt it to the real data to perform classification for those n categories. However, in practice, the unlabeled real data may come from any category, even outside the n categories of interest.

Interestingly, popular DA benchmarks in the computer vision community cre-

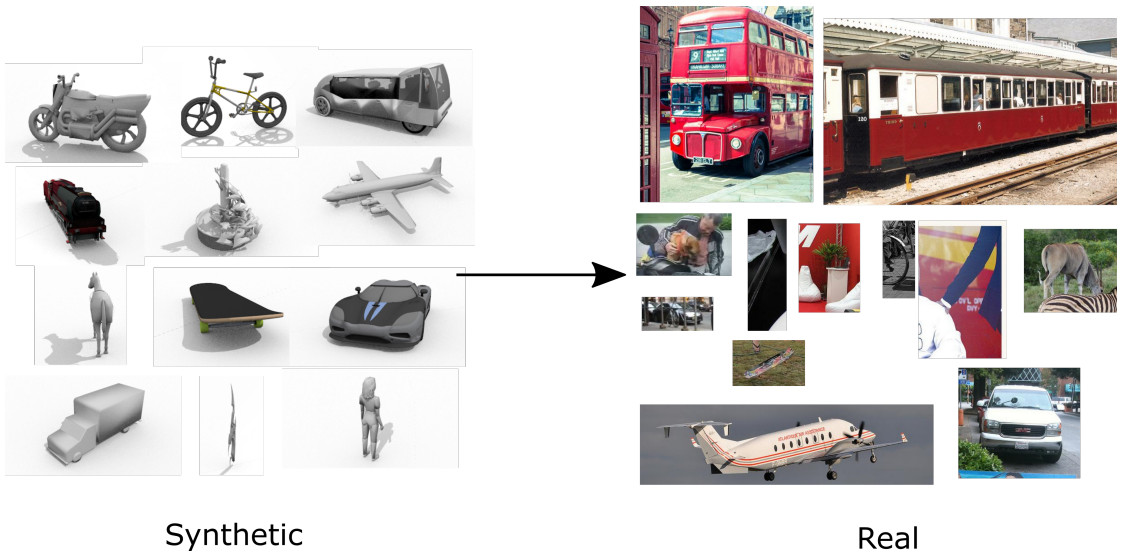


Figure 1.1: Illustration of source and target dataset in a typical domain adaptation setting. The images are from the VisDa-17 dataset. Left: images in the source domain which are collected by rendering the 3D CAD models under different conditions. Right: images in the target domain which collected by cropping objects from the COCO dataset [1].

ate their unlabeled data by choosing images of the n categories and then removing the labels. This strategy is used in DomainNet [2] which is a recent benchmark and also in VisDa-17 [3] which is a well-known benchmark. Even more interestingly, in VisDa-17 dataset, not only are the unlabeled images (adopted from MS-COCO [1] dataset) from the same n categories of interest, but are also cropped to contain only the bounding box of those objects of interest. Effectively, training and testing data are exactly the same. In this work, the terms "standard" or "curated" refer to an unlabeled dataset only containing n categories of interest, while the term "uncurated" refers to an unlabeled dataset containing categories other than n categories of interest. Refer to Figure 1.2 for a comparison of standard vs. uncurated images for VisDa-17. Also, refer to Figure 1.3 for understanding how the change from standard to uncurated dataset changes the count of images per category.

We believe, using curated unlabeled data is not a good idea and is not aligned with the final practical applications. Hence, it can be misleading as a benchmark. The problem is that the process of curation can be seen as a form of weak supervision leaked into the unlabeled training data that may not exist in the real applications. Since some algorithms may exploit this weak supervision, the resulting benchmark can be misleading.

For instance, the entropy minimization method [4] minimizes the entropy of the output to encourage the model to produce a prediction with low uncertainty. This is great in the case of curated data as we know that each unlabeled image corresponds to one of the known categories. However, in the case of uncurated data, the model may be uncertain for data from other unknown categories. Thus, mini-

mizing the entropy loss may not be a good idea. We support this hypothesis in our experiments by adding unknown categories from the target domain to the standard unlabeled data. Fig. 1.2 shows some samples from standard and our uncurated VisDa-17 dataset for comparison.

DA on Uncurated Datasets. Based on the experiments conducted on our uncurated dataset, we realize that all state-of-the-art DA methods drastically degrade in accuracy compared to using only the curated data. However, some methods exploit this bias more. Motivated by this finding, we design a two stage method to reduce the exploitation of this artificial bias. Moreover, our proposed method is simple and yet is either better or on par with other state-of-the-art methods on uncurated datasets. Refer to Figure 1.4 for an illustration of our proposed method.

In the first stage of our method, a self-supervised task of rotation prediction is used to learn good representations in the target domain with unlabeled data. In the second stage, knowledge distillation is used to train a student from scratch using the model trained in the first stage as teacher. Note that the choice of combining these two methods is not arbitrary. Our experiments show that when going from standard to uncurated dataset, these two methods degrade the least. Also, they are conceptually simple and easy to train.

We would like to note that application of these methods to domain adaptation is not our contribution. Domain adaptation using self-supervision has been explored in [5–7]. Training with pseudo labels, closely related to knowledge distillation, has been explored in [8, 9].

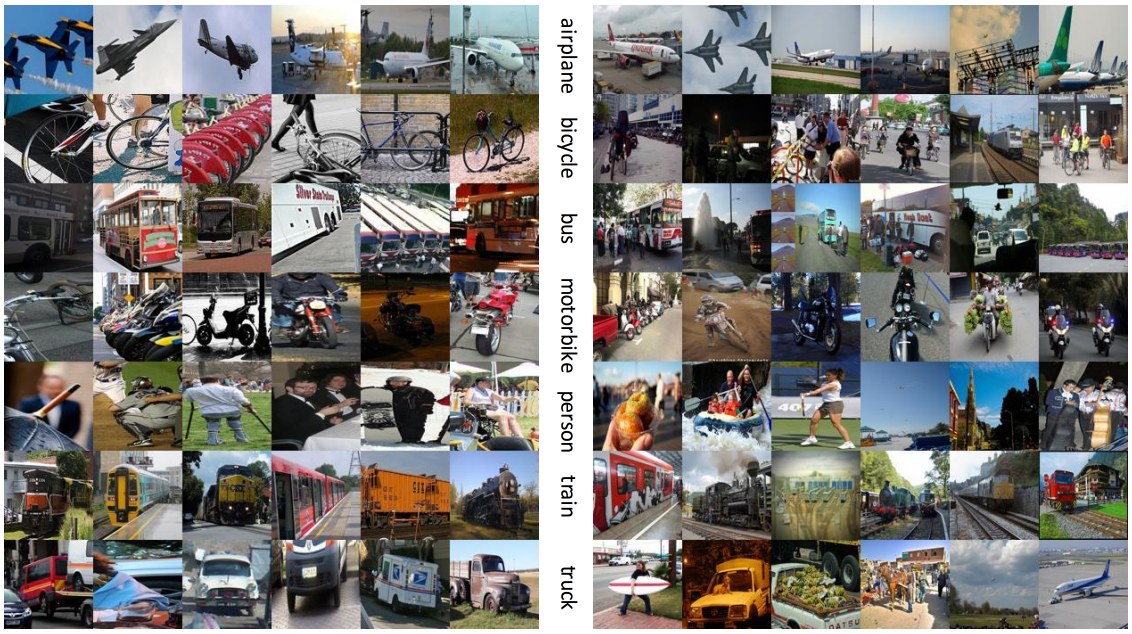


Figure 1.2: Comparison of curated vs. uncurated unlabeled data for VisDa-17 dataset. Left (curated): random samples from the unlabeled target domain dataset used in standard VisDa-17 benchmark. They are originally sampled from MS-COCO dataset and then the objects of interest are cropped and centered. Right (uncurated): random samples for the same categories from MS-COCO dataset without cropping. We believe this cropping process injects bias into the unlabeled dataset that does not exist in practical applications. Such a bias in the benchmark can produce misleading conclusions as some methods may exploit this bias. In our uncurated experiments, we not only use un-cropped images but also use images from other unknown categories.

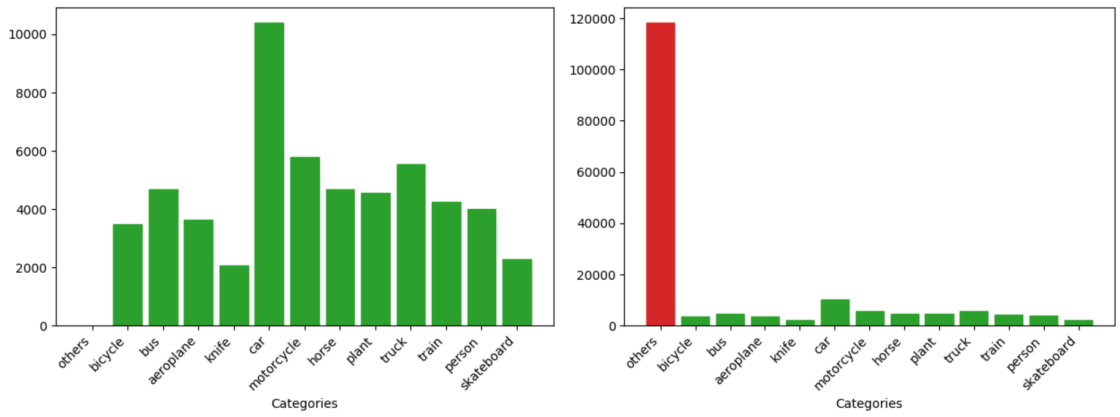


Figure 1.3: Comparison of images per category for standard vs. uncurated dataset. Domain adaptation needs unlabeled data from the target domain to close the domain gap. Interestingly, most benchmarks sample images from the categories of interest and simply remove their labels. We believe this introduces a form of curation bias that some methods can exploit in learning. We add similar images of other categories to reduce the curation bias. We show the number of images per category in unlabaled target data of standard VisDA-17 (left) and our uncurated VisDa-17 (right). Note that the first bin corresponds to images from other categories and the left histogram is almost uniform.

Our Contributions. Inspired by above practical applications, this work focuses on analysing a representative subset of DA methods on a more practical benchmark. We list our key findings below:

- The process of creating an unlabeled target dataset by simply removing the labels is artificial, but the dataset can be made less artificial by appending it with images from categories other than testing categories. This is easy since the datasets are sampled from a bigger dataset.

- We show that the performance of state-of-the-art DA methods degrades drastically when using a less artificial unlabeled target dataset. This suggests that the recent success of DA methods may not translate to practical settings.

- An important source for degradation is entropy minimization, a commonly used component in many state-of-the-art DA methods [10–13]. It degrades more than all the methods we studied.

- Rotation prediction, a self-supervised task, and knowledge distillation degrades the least. Inspired by these results we combine the two methods and show that it is either better than or comparable to the state-of-the-art methods in both semi-supervised and unsupervised domain adaptation settings for uncurated dataset. This approach is illustrated in Figure 1.4.

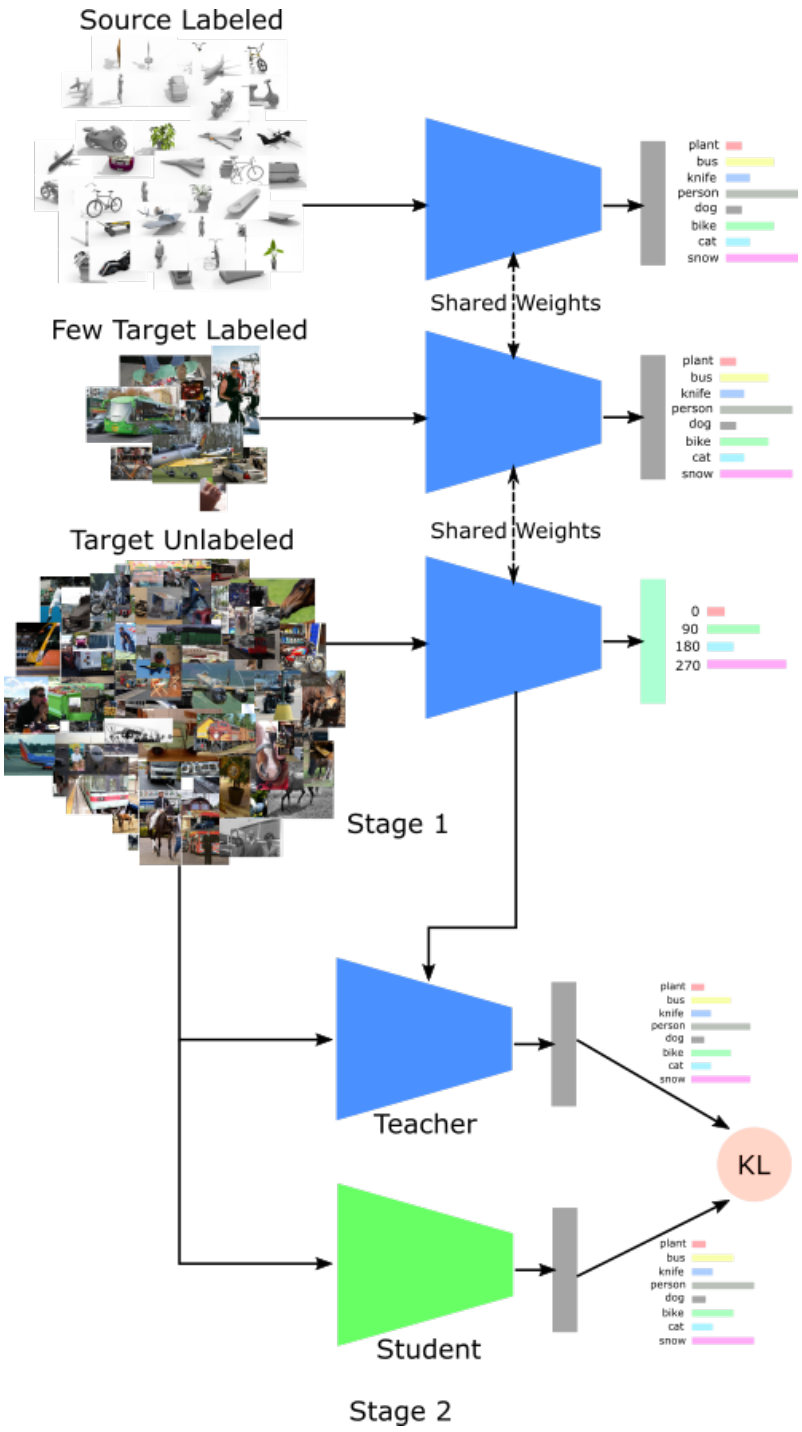


Figure 1.4: Illustration of our proposed method for semi-supervised domain adaptation setting. We do supervised learning and rotation prediction in the first stage and then do knowledge distillation in the second stage.

Chapter 2: Related Work

2.1 Domain Adaptation

A popular strategy for dealing with the domain gap is to learn features that are consistent across domains. One of the most popular methods of aligning features for both domains is adversarial training of a discriminator and a feature extractor such that the discriminator cannot distinguish between the features of source and target domains [11–18]. This method is illustrated in Figure 2.1. Since most of the domain adaptation work focuses on the unsupervised setting, semi-supervised domain adaptation (SSDA) is not well studied. In [19], the standard unsupervised domain adaption (UDA) methods were shown to be less effective in the SSDA setting. [19] introduced an iterative algorithm that alternates between minimizing and maximizing the entropy of the output. Some other works have employed semi-supervised learning. In [8], a network is trained to match the ensembled predictions of its own output obtained at different time intervals during training. Further, combination of adversarial training with semi-supervised techniques like entropy minimization [4] and VAT [20] have been explored in [12, 21]. An improved form of self-training with pseudo-labels is proposed in [9].

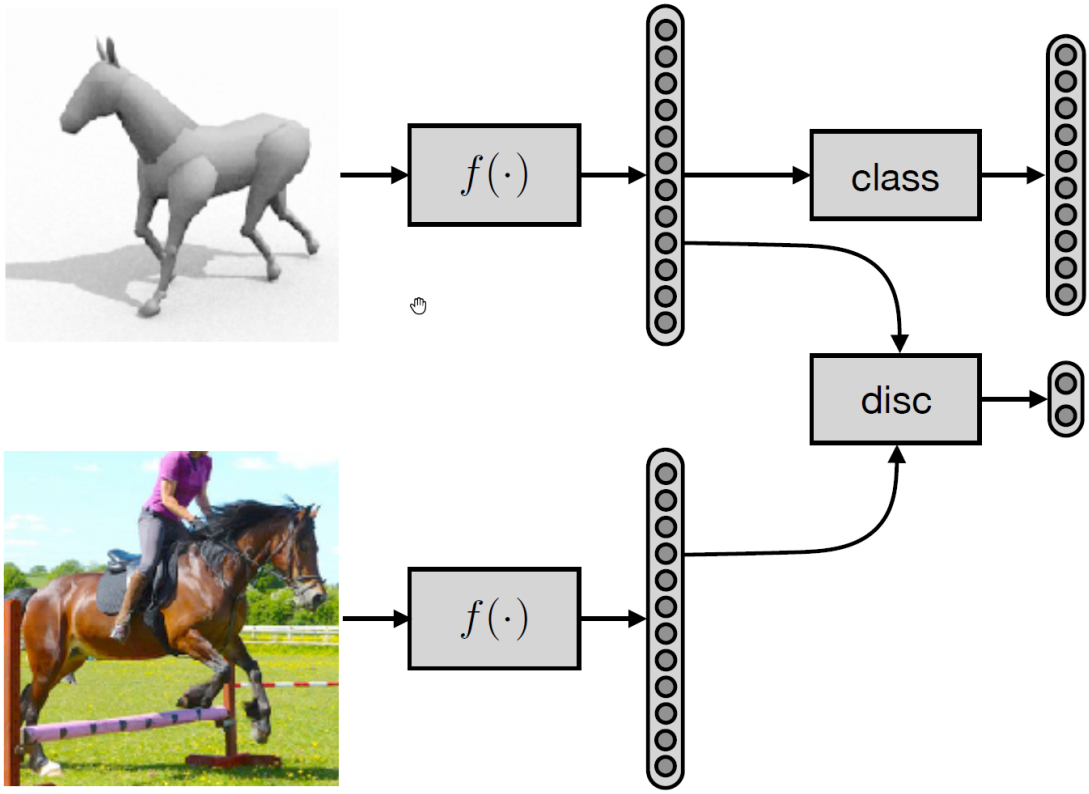


Figure 2.1: Illustration of standard adversarial domain adaptation method. The image on the top is from source domain while the bottom one is from target domain. The feature extractor $f(.)$ is trained to produce features such that: one, *disc* network cannot distinguish between source and target images, and, second, *class* network can classify the source images correctly. The training of $f(.)$ and *disc* is adversarial. The image is take form [18].

2.2 Self-supervised Learning

Numerous pretext tasks, also called pseudo tasks, have been developed for unsupervised representation learning. [22, 23] predict image transformations. In [24, 25], spatial structure of the image is exploited to create pretext tasks. In [26], a model is trained by enforcing count consietncy in image and its tiles. In [27], a model is iteratively trained to classify images based on the labels obtained using k-means clustering. In [28], a teacher network is trained on a hard pretext task and its knowledge is transferred to a student network via k-means clustering. Recently, another class of methods based on contrastive learning have been very successful [29–37]. In [29, 32, 37], a task called instance discrimination is used. In this task, the features from two augmented views of the same image are pulled closer to each other as compared to a bank of negative samples.

For simplicity, we briefly evaluate the Jigsaw [25] pretext task, but focus on the simpler, more effective pretext task of predicting rotations [23].

Aside from representation learning, auxiliary pretext-tasks can also help the model generalize better [38–40]. In [41], self-supervision is applied to semi-supervised learning by incorporating a supervised loss on a small amount of labeled data while solving the pretext task on the entire dataset. In [5], a modified implementation of the Jigsaw pretext task is used as an auxiliary task for domain generalization from multiple source domains to any target domain. In [42], rotation prediction and clustering tasks were used to learn good visual representations from uncurated dataset.

2.3 Knowledge Distillation

Knowledge distillation was originally proposed by [43], and used in [44] to transfer the knowledge from one or more teacher networks to a single student network. [45–47] show that self-distillation, where the teacher and student share the same architecture, improves supervised learning by reducing the generalization gap. It should be noted that the second stage of training with a teacher in [21] is fundamentally different from knowledge distillation. In their second stage, previous version of a model is used to constrain the gradient step of the current model. While, in knowledge distillation, teacher is frozen and student is initialized from scratch. Moreover, it is possible to train the student using an ensemble of teacher where each teacher is trained using a different method.

Chapter 3: Study Setup

3.1 Datasets

This section lays out various components of our study. Sections 3.1.1 and 3.1.2, provide details about the benchmarks, and section 3.2 discusses DA methods studied in this work.

3.1.1 Standard Benchmarks

We conduct experiments on DomainNet [2] and VisDa-17 [3] datasets. DomainNet is a large-scale domain adaptation dataset introduced recently. It has been used in multi-source and semi-supervised domain adaptation settings. VisDa-17 is a widely used dataset in UDA works.

Standard DomainNet. DomainNet [2] is a large scale domain adaptation dataset with 6 domains (Real, Clipart, Sketch, Painting, Quickdraw, and Infograph) and 345 categories. It contains about 0.6 million images. It surpasses all other previous domain adaptation datasets in terms of size and diversity. We refer to the subset of DomainNet used in [19] as standard DomainNet. This subset consists of 4 domains (Real, Clipart, Sketch, and Painting) and 126 categories. Of all possible

domain pairs (source-target), 7 are chosen for evaluation. Further, two different semi-supervised settings, 1-shot and 3-shot, are created by keeping the labels for 1 and 3 samples per class while discarding the labels for the rest. We use the same dataset splits as [19].

Standard VisDa-17. VisDa-17 [3] is a dataset for UDA. The source dataset consists of synthetic images obtained by rendering 3D models at different angles and lighting conditions. The target domain consists of images filtered and cropped from MS-COCO dataset [1] using ground truth bounding boxes to only contain objects of interest. Both source and target domains contain 12 categories. The target dataset has 55k images, while the source dataset has 150k images.

3.1.2 Uncurated Benchmarks

Here, we describe the uncurated versions of the standard datasets listed above. We compare the sizes of unlabeled images in standard and uncurated versions in Table 3.1.

Uncurated DomainNet. One of the reasons to choose DomainNet and particularly its subset used in semi-supervised setting is the ability to simulate true unlabeled data. We create a dataset by taking images from all 345 available categories for 4 domains in the standard DomainNet. We discard all labels and only use it as unlabeled images for target domain. We refer to this dataset as uncurated DomainNet.

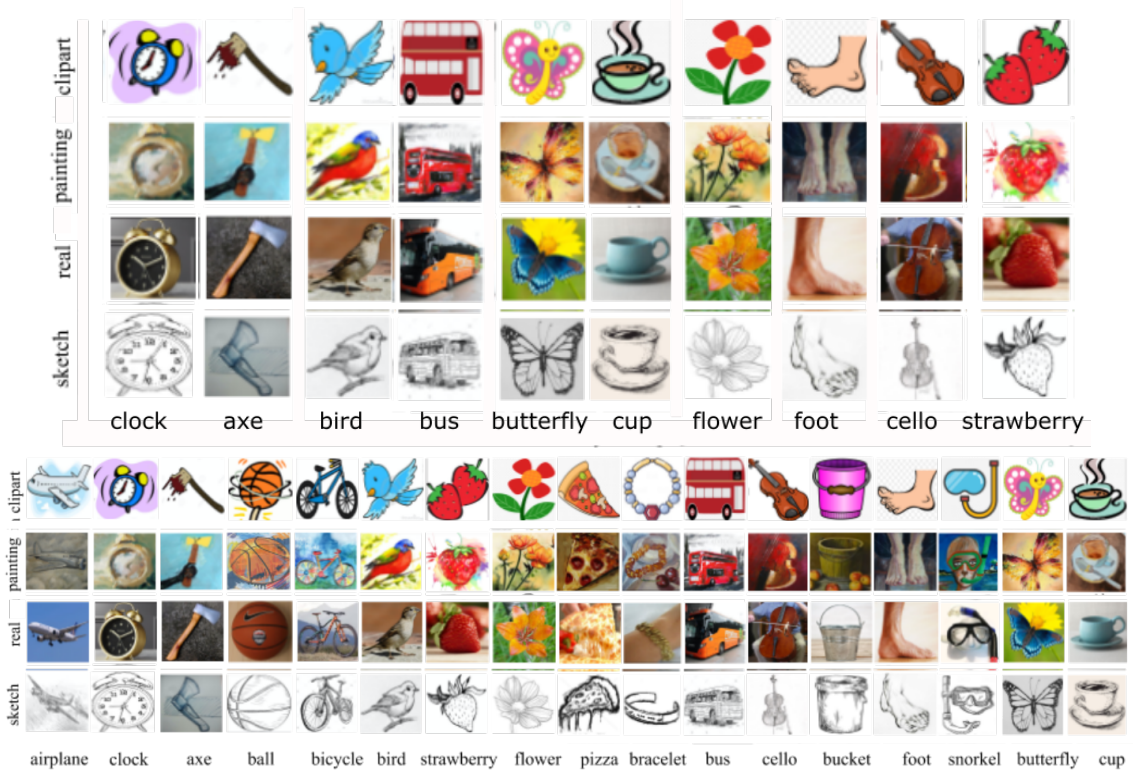


Figure 3.1: Illustration of images in standard and uncurated DomainNet. Top: sample images for each of the categories and 4 domains (real, clipart, sketch, and painting). Bottom: sample images for each of the categories and domains. Note that the categories in the uncurated DomainNet are the superset of categories in standard DomainNet. The images from these extra categories are used to construct an unlabeled target dataset with reduced curation bias for domain adaptation.

Domain	Standard	Uncurated
DomainNet Real	70k	175k
DomainNet Clipart	18k	48k
DomainNet Sketch	24k	70k
DomainNet Painting	31k	75k
VisDa-17 Real	55k	173k

Table 3.1: Comparison of the count of unlabeled target images in standard vs. uncurated datsaet. We add more unlabeled images from other categories to reduce the curation bias.

Uncurated VisDa-17. Similar to uncurated DomainNet, we construct uncurated VisDa-17 by adding all training images of MS-COCO to the target dataset. This ensures that the uncurated, unlabeled target dataset contains more than just training categories.

3.2 Methods

Here, we describe the general framework for domain adaptation used in this study. We consider different state-of-the-art domain adaptation methods. First, we describe the supervised component that is common to all the listed methods. Given an image x^s and its label y^s in the source domain, an image x^t and its label y^t in the target domain, and also an unlabeled image x^u in the target domain, we define the following loss terms:

$$\mathcal{L}^s_{sup}(f) = \sum_i \ell_{ce}(f(x^s_i), y^s_i)$$
$$\mathcal{L}^t_{sup}(f) = \sum_i \ell_{ce}(f(x^t_i), y^t_i)$$

where $\ell_{ce}(.)$ is the cross entropy loss, $f(.)$ is the classifier, $\mathcal{L}^s_{sup}$ is the supervised loss on the source domain, and $\mathcal{L}^t_{sup}$ is the supervised loss on the target domain. Both losses are optimized jointly in SSDA while $\mathcal{L}^s_{sup}$ is absent in UDA since there is no labeled data in the target domain. This is the supervised component of domain adaptation. In addition, we describe various unsupervised loss terms that can be optimized jointly with above terms in a multi-task setting.

Rotation Prediction (ROT) is a self-supervised task for learning representations [23]. Following the RotNet method [23], given an unlabeled image, we rotate it using a rotation angle randomly chosen from the list $\{0°, 90°, 180°, 270°\}$, and then define its corresponding label to be the rotated angle (1 out of 4 possibilities). Then, we input rotated images to the network and optimize the network to detect the rotation angle using the cross entropy loss function.

$$\mathcal{L}^t_{ssl}(r) = \sum_i \ell_{ce}(r(T_a(x^u_i)), a)$$

Where $r(.)$ is the rotation prediction classifier, and $T_a(.)$ is an operator that rotates the input image by an angle $a \in \{0°, 90°, 180°, 270°\}$. The value of a is chosen randomly for each data point and iteration. Note that $f(.)$ and $r(.)$ share all layers except the last one.

Knowledge Distillation (KD) is the method of transferring knowledge from

a teacher network to a student network [44]. We use the same architecture for both student and teacher networks. This is similar to [45,46] except that we do distillation on unlabeled target data rather than labeled data. Similar to [45, 46], our intuition is that this method will reduce the generalization gap caused by replacing one-hot encoding of the ground truth with soft labels. We optimize:

$$\mathcal{L}^t_{KL} = KL(f(x^u_i)||g(x^u_i)$$

Where $KL(.)$ is the KL divergence loss, $g(.)$ is the student network, and $f(.)$ is the teacher network that is frozen during knowledge distillation. Note that $f(.)$ and $g(.)$ share the same network architecture, but the weights in $g(.)$ are initialized from an ImageNet-pretrained network. Also, note that we can choose to distill from multiple teacher networks in an ensemble setting (Table 4.4). In this work, KD(X) refers to a student model trained using knowledge distillation where the teacher is trained using method X.

Entropy Minimization (ENT) was initially proposed in [4] as a form of semi-supervised learning. It works on the assumption that the model should be confident about its prediction on the unlabeled data. Alternately, minimizing entropy leads to a confident model. We can minimize the entropy of the predicted probability on the unlabeled data using following loss function:

$$\mathcal{L}^t_{ent}(f) = \sum_i ent(f(x^u_i))$$

Where $ent(p)$ calculates the entropy of distribution p.

Virtual Adversarial Training (VAT) is a regularization method introduced in [20] that encourages a model to be smooth around each unlabeled data-point. It calculates the adversarial example for each data-point by applying a small perturbation and makes sure that the model's output does not change with the perturbation.

$$\mathcal{L}_{vat}^{t}(f) = \sum_{i} \max_{\|r\|<=\epsilon} (KL(f(x_i^u)||f(x_i^u + r))$$

Where $KL(.)$ is the KL divergence loss, and r is a perturbation to x_i^u that maximizes the KL divergence between the perturbed and non-perturbed input.

Maximum Classifier Discrepancy (MCD) is an adversarial domain adaptation method proposed in [10] that involves alternating between maximizing and minimizing the discrepancy between outputs of two different task specific heads. Each training step is composed of multiple forward passes through the network. This makes the method slow to train. Entropy minimization is used in this method.

Conditional Adversarial Domain Adaptation (CDAN+E) is a form of adversarial domain adaptation proposed in [11] that leverages conditioning of the classifier predictions. It proposes a novel conditional domain discriminator conditioned on the cross-covariance of domain specific feature representations and classifier predictions. CDAN+E is a variant of this method based on entropy conditioning.

Batch Sectral Penalization (BSP+CDAN+E) is proposed in [13] as a regularization method to enhance the discriminability of the features learned using adversarial domain adaptation methods. They show that the eigenvectors of a batch with top singular values are the cause of reduced discriminability. Thus, penalizing

them is a good regularization method. We analyze the application of this regularization to CDAN+E and refer to it as BSP+CDAN+E. Note that BSP+CDAN in [13] is actually BSP+CDAN+E.

Minimax Entropy (MME) is a domain adaptation method proposed in [19] that alternates between updating the class prototypes by maximizing entropy and updating the feature extractor by minimizing entropy. The method is applied in the context of semi-supervised domain adaptation where it was shown to be significanlty better than other state-of-the-art UDA methods.

Drop to Adapt (DTA) is a method proposed in [12] that extends the adversarial dropout method [17] to convolutional layers and applies it to enforce cluster assumption in the setting of domain adaptation. Along with their method, they also add VAT and ENT components to the loss function. Their loss function is complex and consists of 5 different components. Thus, to isolate their core method, we also consider a variant of their method called cDTA+fDTA which does not have the VAT component. It should be noted that cDTA+fDTA still has the ENT component.

In various experiments, we combine the above unsupervised losses with the supervised loss by weighted addition. Each loss has its own hyperparameter weight λ_{ent}, λ_{vat}, λ_{ssl}, λ^s_{sup}, and λ^t_{sup}. We assume $\lambda^t_{sup} = 1$ and report the other hyperparamters in the implementation details 4.6. When we add two methods, we refer to the final model as their summation, e.g., ENT+VAT. KD(X) refers to training a new student with a teacher trained using method X.

We use following official source code repositories in our study:

- For MME and ENT methods on DomainNet dataset, we use: https://github.com/VisionLearningGroup/SSDA_MME
- For MCD, we use: https://github.com/mil-tokyo/MCD_DA
- For cDTA+fDTA and DTA, we use: https://github.com/postBG/DTA.pytorch
- For BSP+CDAN+E and CDAN+E, we use: https://github.com/thuml/Batch-Spectral-Penalization

Chapter 4: Experiments and Results

Note that this section uses many abbreviations. To make it simpler: UDA and SSDA are tasks, DomainNet and VisDa-17 are datasets, and all other abbreviations are the methods described in the method section.

We describe UDA and SSDA experiments in detail and discuss the results. We downloaded the official code for MME, BSP+CDAN+E, MCD, CDAN+E, and DTA. We reran the methods and were able to successfully reproduce the results reported in the respective papers. We use these results for calculating degradation. For experiments on the uncurated dataset, we only change the unlabaled target data in all methods to uncurated data. We implement ENT, VAT, ROT, and KD methods ourselves. Implementation details can be found in Section 4.6. Note that the implementation of ENT method for SSDA is from the official code for MME method [19]. We follow the evaluation protocol for SSDA in [19] and evaluate our method on AlexNet and ResNet for 1-shot and 3-shot settings. We used ResNet34 for all SSDA experiments following [19] and ResNet101 for all UDA experiments. The baseline for SSDA experiments is called S+T which does not use any unlabeled target data while for UDA it is called source only.

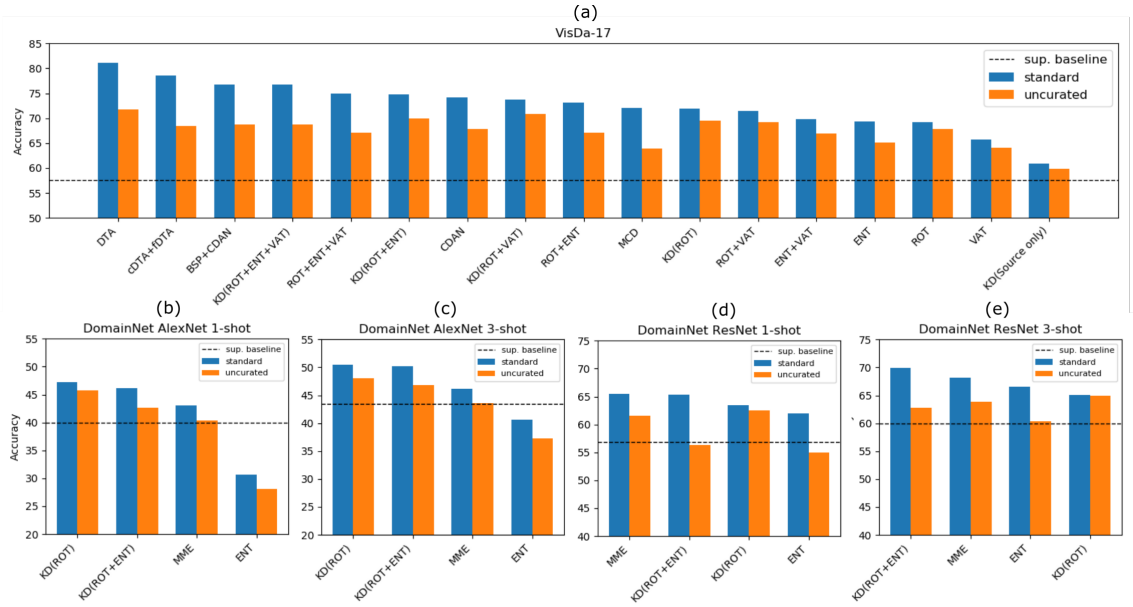


Figure 4.1: Comparison of standard vs. uncurated dataset for various methods. Top: VisDa-17 with ResNet101. Bottom: DomainNet with Alexnet and ResNet34. The methods are sorted by the accuracy on the standard set. All methods degrade when the curation bias is reduced while some methods like KD(ROT) have less degradation. Note that the y-axis is truncated around the baseline to better show the differences.

Data	Method	AlexNet		ResNet	
		1-shot	3-shot	1-shot	3-shot
Standard	S+T	40.0	43.4	56.9	60.0
	DANN [19]	40.4	42.4	58.4	60.7
	ADR [19]	39.2	42.7	57.6	60.4
	CDAN [19]	39.1	41.0	62.5	66.5
	ENT [19]	29.1	39.8	62.6	67.6
	MME [19]	44.2	48.2	**66.4**	68.9
	KD(ROT+ENT)	46.2	50.2	65.3	**69.9**
	KD(ROT)	**47.2**	**50.4**	63.5	65.1
Uncurated	ENT	28.1	37.3	55.0	60.4
	MME	40.3	43.6	61.5	63.8
	KD(ROT+ENT)	42.7	46.8	56.3	62.8
	KD(ROT)	**45.7**	**48.0**	**62.5**	**64.9**

Table 4.1: Results on DomainNet for SSDA task with following settings: 1-shot and 3-shot, AlexNet and ResNet34 architectures, and standard and uncurated datasets. We report the average accuracy over all seven possible pairs of source-target domains: (R to C, R to P, P to C, C to S, S to P, R to S, P to R). The results for each pair is presented in the tables A.1,A.2,A.3 and A.4 of Appendix A. We do experiments with both standard and uncurated datasets. Interestingly, KD(ROT), which is a very simple method compared to SOTA models, performs very well in the uncurated setting. The relative degradation is reported in table 4.2.

4.1 Analysis of entropy minimization

Entropy minimization always hurts in uncurated datasets. Table 4.2 shows the results for degradation in accuracy when going from standard to uncurated dataset on DomainNet. Both ENT and KD(ROT+ENT) degrade the most for all combinations of models and number of shots. Interestingly, for ResNet, 1-shot experiment on uncurated data, the results of KD(ROT+ENT) and ENT are worse than baseline (S+T), which does not use any unlabaled data (Table 4.1 and Figure 4.1.) Further, all methods with ENT in Table 4.6 degrade at least more than 6%. Note that MCD, CDAN+E, BSP+CDAN+E, cDTA+fDTA, and DTA also contain entropy minimization component and degrade by around 10%. This clearly shows the limitations of applying entropy minimization in the case of uncurated data.

Entropy helps ResNet but not AlexNet on standard datasets. Tables 4.1 and 4.6 show that on both standard datasets, ResNet architecture benefits from adding ENT method to any other method. However, for AlexNet architecture, the ENT method itself is worse than the S+T baseline (no unlabeled data) (see Fig. 4.1). Hence for AlexNet, adding ENT to any other method degrades the performance on the standard dataset (for DomainNet). We do not know the reason for this behaviour and note that it has been documented in [19] too. Studying this behaviour can be an interesting future work.

Method	AlexNet		ResNet	
	1-shot	3-shot	1-shot	3-shot
ENT	-8.2%	-8.1%	-11.3%	-9.3%
MME	-6.3%	-5.6%	-6.0%	-6.5%
KD(ROT+ENT)	-7.6%	-6.8%	-13.8%	-10.2%
KD(ROT)	**-3.2%**	**-4.8%**	**-1.6%**	**-0.3%**

Table 4.2: Degradation of the accuracy when changing the unlabeled data from standard to the uncurated one for DomainNet. Degradation percentage is relative to the standard dataset. All methods degrade when going from standard to uncurated data.

4.2 Analysis of knowledge distillation

Knowledge distillation is not very sensitive to curation bias. For knowledge distillation, the teacher and student do not need to use the same dataset (standard or uncurated). We experiment with varying this datasets to measure robustness of KD method to the curation bias in Table 4.3. We observe that KD always improves over the teacher model regardless of what dataset is used for distillation. Interestingly, when the teacher is trained on the standard dataset, doing KD on the uncurated deadset does not degrade the accuracy much (last column of top section in Table 4.3.)

Teacher Dataset	Teacher Method	Teacher	Student on Std.	Student on Unc.
Standard	ROT	69.2	71.9	70.7
	ROT+ENT	73.1	74.8	74.1
	ROT+VAT	71.5	73.7	73.4
	ROT+ENT+VAT	74.9	76.7	75.9
Uncurated	ROT	67.9	70.7	69.5
	ROT+ENT	67.1	69.1	69.9
	ROT+VAT	69.2	72.1	70.8
	ROT+ENT+VAT	67.1	68.7	68.7

Table 4.3: Effect of the dataset type on KD for VisDa-17 (ResNet101). Results of distillation using standard and uncurated data separately. We do not observe a large degradation in accuracy when we change the dataset of distillation from standard to uncurated. This shows that distillation is robust to curation bias.

Method	P to R	S to P	C to S	P to C	R to P	R to C	R to S	Mean
ROT_{sup1}	60.4	44.3	41.5	49.7	49.9	51.6	40.1	48.2
ROT_{sup2}	60.3	45.2	41.9	49.6	49.8	51.4	40.5	48.3
$ROT_{sup+rot}$	59.2	44.0	42.6	49.9	49.7	52.4	41.3	48.4
$ens(ROT_{sup1}, ROT_{sup2})$	62.3	47.0	43.7	52.1	51.9	53.5	42.3	50.4
$ens(ROT_{sup1}, ROT_{sup+rot})$	61.9	46.8	44.4	52.8	52.0	54.3	43.2	50.8
$KD(ens(ROT_{sup1}, ROT_{sup+rot}))$	63.4	48.3	45.5	54.1	53.1	55.6	44.2	52.0

Table 4.4: The effect of pretraining for DomainNet. We use 3-shot standard DomainNet and AlexNet for these experiments. We initialize our method with three different pretrained models (ROT_{sup1}, ROT_{sup2}, and $ROT_{sup+rot}$). The first two are regular supervised pretrainings on ImageNet and the last one uses both, supervised and rotation prediction loss, on ImageNet pretraining. We show that adding RotNet to the pretraining helps slightly to generalize better. We also show that using this model as one of the teachers in ensembling and distillation helps by almost 4 points.

Distilling from an ensemble is simple with knowledge distillation. As shown above, knowledge distillation is a simple yet effective method. Moreover, it is straightforward to distill a student from an ensemble of teachers. If the teachers in an ensemble make errors that are orthogonal to each other, then the ensemble will have a higher accuracy than any of the teachers. One way of coming up with orthogonal teachers is to use different self-supervised tasks as auxiliary losses. We show the results for this experiment in Table 4.4. Interestingly, the student model reaches 52% (4 points higher than MME which is state-of-the-art) for AlexNet on 3-shot SSDA.

Method	AlexNet	ResNet34
S+T	43.4	60.0
ROT+ENT	48.6	68.5
JIG+ENT	44.1	59.5

Table 4.5: Jigsaw vs. RotNet as the self-supervised task. We use solving jigsaw puzzles instead of rotation prediction for the SSL task. We see that even though Jigsaw might be better than RotNet for domain generalization, as shown in [4], Jigsaw is significantly worse than RotNet for Semi-Supervised Domain Adaptation (SSDA) that we studied. We report mean accuracy for all 7 pairs on standard DomainNet dataset for 3-shot setting, and both AlexNet and ResNet34 architectures.

4.3 Jigsaw vs. Rotation

Since [5] uses Jigsaw solver in domain generalization setting, here, we study using Jigsaw instead of RotNet in our SSDA setting. We tried $\lambda_{ssl} \in \{0.7, 1.0\}$ and $\lambda_{ent} \in \{0.01, 0.1\}$ for the pair of real to sketch and picked the best combination. These are the parameters used in [5]. We do not do knowledge distillation for these experiments. We list our results in Table 4.5. We found that Jigsaw worked almost similar to the S+T baseline. Note that Jigsaw is shown to be more effective than RotNet in [5], but it is significantly worse than RotNet in our SSDA setting. We empirically conclude that Jigsaw does not generalize well in the case of single source domain.

4.4 Comparison of methods

Most methods are equally good on uncurated data for UDA. From Table 4.6, it can be seen that the performance of a lot of methods lies in the range of 67-70% on uncurated data. Concretely, the standard deviation of the results of all methods on standard dataset is 4.7 but only 2.8 for uncurated dataset. This difference in variance suggests that some methods can exploit the curation bias more, but this boost does not translate to uncurated data. The accuracies saturate on uncurated data. In other words, most methods are only marginally different from each other on a more practical dataset.

KD(ROT) is a competitive baseline. We believe this due to the following reasons: (1) It is conceptually "simple" and easy to implement. It has only one hyperparameter (weight of the rotation prediction loss function) unlike DTA. It does not need multiple forward passes or adversarial training unlike VAT, MCD, or CDAN+E. (2) KD(ROT) is "consistently" better than the baseline on both tasks (UDA in Table 4.6 and SSDA in Table 4.1). As an example, note that ADR is almost 17 points better than the baseline (source only) in Table 4.6 for UDA task, but is either worse or close to the baseline (S+T) for SSDA task in Table 4.1. (3) It achieves "state-of-the-art results" for AlexNet on both, 1-shot and 3-shot, SSDA tasks (Table 4.1.) This indicates that KD(ROT) might be a better method for smaller models. (4) It "degrades the least" when going from standard to uncurated dataset which shows that it is less sensitive to the curation bias.

Method	Reported	Standard	Uncurated	Degradation
Source only	-	57.6	57.6	0.0%
KD(Source only)	-	60.9	59.8	-1.8%
VAT	-	65.7	64.1	-2.4%
ROT	-	69.2	67.9	-1.9%
ENT	-	69.4	65.1	-6.2%
ENT+VAT	-	69.8	66.9	-4.2%
ROT+VAT	-	71.5	69.2	-3.2%
KD(ROT)	-	71.9	69.5	-3.3%
ROT+ENT	-	73.1	67.1	-8.2%
KD(ROT+VAT)	-	73.7	70.8	-3.9%
KD(ROT+ENT)	-	74.8	69.9	-6.6%
ROT+ENT+VAT	-	74.9	67.1	-10.4%
KD(ROT+ENT+VAT)	-	76.7	68.7	-10.4%
DANN [48]	57.4 [13]	-	-	-
MCD [10]	71.9 [13]	72.0	63.9	-11.3%
CDAN+E [11]	73.7 [13]	74.2	67.8	-9.4%
ADR [17]	74.8 [12]	-	-	-
BSP+CDAN+E [13]	75.9 [13]	76.7	68.7	-10.4%
cDTA+fDTA [12]	77.4 [12]	78.5	68.5	-12.7%
CRST [9]	78.1 [9]	-	-	-
DTA [12]	**81.5** [12]	**81.1**	**71.8**	-11.5%

Table 4.6: Results of ResNet101 on standard and uncurated datasets for VisDa-17. cDTA+fDTA refers to DTA [12] without VAT introduced in [12]. The degradation percentage is relative to the standard dataset. Results in standard column are ours.

4.5 Source dataset for self-supervision

Self-supervision for domain adaptation is also studied in [7]. The central hypothesis of [7] is that solving a pretext task (self-supervised) for both the source and target domain will lead to alignment of features between the two domains. Ideally, if we had labels in the target domain as well, then we could just use them in a supervised way for inducing alignment, but since we don't have them we can replace supervised labels with self-supervised labels. This approach is slightly different from our proposed method. In Table 4.7, we perform experiments comparing our approach with [7]. We do not find any supporting evidence for the hypothesis proposed in [7]. Only doing the self-supervised task on the target dataset is sufficient.

4.6 Implementation details

Our code is implemented in PyTorch (1.0) and closely follows the implementation of [19]. Because the focus of this work is on obtaining baselines using methods that are easy to train, we refrain from extensive hyper-parameter tuning.

Semi-supervised domain adaptation. We use AlexNet [49] and ResNet34 [50] pre-trained on ImageNet in all of our experiments. The architectures of feature extractor and supervised classification head are the same as [19] for fair comparison. For self-supervised classification head (rotation prediction), we use a single fully-connected layer for AlexNet and two fully-connected layers with a ReLU between

Model	Method	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
AlexNet	ROT(t)+ENT	52.9	49.6	51.1	41.2	44.3	40.7	60.1	48.6
	ROT(s+t) + ENT	53.1	48.0	50.9	41.7	43.0	41.6	60.6	48.4
ResNet34	ROT(t)+ENT	70.4	69.4	69.9	62.8	65.8	62.3	79.1	68.5
	ROT(s+t) + ENT	70.9	70.0	70.4	61.2	65.0	64.0	77.7	68.5

Table 4.7: Comparison of using both source and target domains vs. only target domain for self-supervised task. We study the importance of using both source and target domains for domain alignment using self-supervised tasks as proposed in [38] and find that this alignment does not help. We use 3-shot setting on standard DomainNet. ROT(t) only uses the target dataset for the RotNet loss while ROT(s+t) uses both source and target datasets. We show that ROT(s+t) is no better than ROT(t) on average.

them for ResNet34. The model is optimized using SGD with a momentum of 0.9 and weight decay of 0.0005. The initial learning rate for feature extractor is 0.001 while for both classification heads it is 0.01. We use the same learning rate annealing schedule as in [48].

We tried $\lambda_{ent} \in \{0.1, 0.01, 0.05\}$ values on Real to Sketch pair for 3-shot setting. We use $\lambda_{ent} = 0.01$ for AlexNet and $\lambda_{ent} = 0.1$ for ResNet34. We use $\lambda_{ssl} = 1, \lambda^s_{sup} = 1$ for all experiments. The training is run for 30k iterations and the checkpoint with best validation accuracy is used for testing.

Unsupervised Domain Adaptation. For a fair comparison with other works, we only use ResNet101 [50]. Apart from weights for losses, all other hyper-parameters are the same as above. We search for $\lambda^s_{sup} \in \{0.5, 1.0\}$ and $\lambda_{ent} \in \{0.05, 0.01, 0.1\}$. We use $\lambda^s_{sup} = 0.5, \lambda_{ent} = 0.01, \lambda_{vat} = 0.01$ for all our experiments when the corresponding losses are used. Also, parameters for the VAT [20] are the same as the original work. The training is run for 30k iterations.

Knowledge Distillation. We start with an ImageNet pre-trained student. We run the training for 10 epochs while dropping learning rate by a factor of 0.1 every 3 epochs. We intentionally keep the number of epochs for distillation small to reduce computational time and keep the experiments simple. We don't use temperature in our experiments.

Chapter 5: Conclusions and Future Work

5.1 Conclusions

We study the effect of curation bias that already exists in two well-known domain adaptation benchmarks using various SOTA methods. We find that reducing the bias is easy and degrades the performance of methods. Particularly, some methods like ENT can exploit this bias effectively. We also find that some simple methods like KD(ROT) are relatively more robust to the curation bias. We believe this is important since the curation bias may not exist in many real world applications, so including it in the benchmark may be misleading for the community.

5.2 Future Work

We attempted to understand the effect of curation for domain adaptation in this work. To the best of our knowledge this is one of the first works to tackle this problem. We expect future datasets in domain adaptation to be designed to have reduced curation bias. More work is also needed to find novel methods that are robust to curation bias.

Appendix A: Expanded results for semi-supervised domain adaptation on DomainNet

Method	Data	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	S	43.3	42.4	40.1	33.6	35.7	29.1	55.8	40.0
DANN	S	43.3	41.6	39.1	35.9	36.9	32.5	53.6	40.4
ADR	S	43.1	41.4	39.3	32.8	33.1	29.1	55.9	39.2
CDAN	S	46.3	45.7	38.3	27.5	30.2	28.8	56.7	39.1
ENT	S	37.0	35.6	26.8	18.9	15.1	18.0	52.2	29.1
MME	S	48.9	48.0	46.7	36.3	39.4	33.3	56.8	44.2
KD(ROT+ENT)	S	**51.0**	**50.5**	47.8	37.7	38.1	38.0	**60.4**	46.2
KD(ROT)	S	49.3	49.9	**48.3**	**39.7**	**44.3**	**40.0**	58.7	**47.2**
ENT	U	30.6	33.8	25.2	22.2	16.7	17.4	50.8	28.1
MME	U	41.2	43.0	39.4	34.1	39.7	30.9	53.4	40.3
KD(ROT+ENT)	U	46.4	47.2	43.7	34.3	35.8	33.3	57.9	42.7
KD(ROT)	U	**47.7**	**48.9**	**46.5**	**37.4**	**42.7**	**38.1**	**58.5**	**45.7**

Table A.1: AlexNet 1-shot: We report the accuracies for standard and uncurated DomainNet datasets on 1-shot setting and AlexNet. This is an expansion of the Table 4.1.

Method	Data	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	S	47.1	45.0	44.9	36.4	38.4	33.3	58.7	43.4
DANN	S	46.1	43.8	41.0	36.5	38.9	33.4	57.3	42.4
ADR	S	46.2	44.4	43.6	36.4	38.9	32.4	57.3	42.7
CDAN	S	46.8	45.0	42.3	29.5	33.7	31.3	58.7	41.0
ENT	S	45.5	42.6	40.4	31.1	29.6	29.6	60.0	39.8
MME	S	55.6	49.0	51.7	39.4	43.0	37.9	60.7	48.2
KD(ROT+ENT)	S	**54.7**	50.9	**53.0**	42.3	**46.3**	41.8	**62.1**	50.2
KD(ROT)	S	54.5	**51.3**	52.9	**43.2**	**46.3**	**43.2**	61.5	**50.4**
ENT	U	40.5	40.9	33.3	30.7	31.6	26.9	57.5	37.3
MME	U	45.8	44.3	43.5	37.1	42.0	34.7	57.8	43.6
KD(ROT+ENT)	U	48.9	**50.0**	49.0	39.2	41.8	38.1	**60.5**	46.8
KD(ROT)	U	**51.3**	48.3	**50.2**	**41.7**	**43.8**	**40.5**	60.3	**48.0**

Table A.2: AlexNet 3-shot: We report the accuracies for standard and uncurated DomainNet datasets on 3-shot setting and AlexNet. This is an expansion of the Table 4.1.

Method	Data	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	S	55.6	60.6	56.8	50.8	56.0	46.3	71.8	56.9
DANN	S	58.2	61.4	56.3	52.8	57.4	52.2	70.3	58.4
ADR	S	57.1	61.3	57.0	51.0	56.0	49.0	72.0	57.6
CDAN	S	65.0	64.9	63.7	53.1	63.4	54.5	73.2	62.5
ENT	S	65.2	65.9	65.4	54.6	59.7	52.1	75.0	62.6
MME	S	**70.0**	**67.7**	**69.0**	56.3	**64.8**	**61.0**	**76.1**	**66.4**
KD(ROT+ENT)	S	65.6	70.4	64.8	**58.1**	62.6	60.0	75.8	65.3
KD(ROT)	S	63.6	66.5	60.8	57.5	63.4	58.3	74.6	63.5
ENT	U	52.4	59.8	53.2	49.4	54.8	45.3	70.1	55.0
MME	U	59.9	64.2	60.4	56.0	**63.2**	54.5	72.6	61.5
KD(ROT+ENT)	U	56.1	61.9	49.8	50.8	57.4	49.0	69.2	56.3
KD(ROT)	U	**60.7**	**65.9**	**61.9**	**57.6**	61.7	**56.1**	**73.6**	**62.5**

Table A.3: ResNet 1-shot: We report the accuracies for standard and uncurated DomainNet datasets on 1-shot setting and ResNet. This is an expansion of the Table 4.1.

Method	Data	R to C	R to P	P to C	C to S	S to P	R to S	P to R	Mean
S+T	S	60.0	62.2	59.4	55.0	59.5	50.1	73.9	60.0
DANN	S	59.8	62.8	59.6	55.4	59.9	54.9	72.2	60.7
ADR	S	60.7	61.9	60.7	54.4	59.9	51.1	74.2	60.4
CDAN	S	69.0	67.3	68.4	57.8	65.3	59.0	78.5	66.5
ENT	S	71.0	69.2	71.1	60.0	62.1	61.1	78.6	67.6
MME	S	**72.2**	69.7	**71.7**	61.8	66.8	61.9	78.5	68.9
KD(ROT+ENT)	S	71.6	**70.8**	71.2	**64.1**	**67.4**	**63.6**	**80.7**	**69.9**
KD(ROT)	S	64.0	67.0	65.0	60.9	62.6	60.3	75.6	65.1
ENT	U	58.2	63.2	59.8	52.7	61.5	52.3	75.2	60.4
MME	U	63.1	66.6	**64.5**	57.1	**65.5**	54.5	75.0	63.8
KD(ROT+ENT)	U	64.0	66.7	58.9	57.2	61.6	55.4	**76.1**	62.8
KD(ROT)	U	**64.8**	**67.5**	63.9	**60.0**	64.7	**57.5**	**76.1**	**64.9**

Table A.4: ResNet 3-shot: We report the accuracies for standard and uncurated DomainNet datasets on 3-shot setting and ResNet. This is an expansion of the Table 4.1.

Bibliography

[1] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[2] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[3] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *ArXiv*, abs/1710.06924, 2017.

[4] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[5] Fabio M. Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] Xu Jiaolong, Xiao Liang, and Antonio M. López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019.

[7] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision, 2019.

[8] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

[9] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019.

[10] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2017.

[11] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2017.

[12] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[13] 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, 2019.

[14] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[15] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[16] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogério Schmidt Feris, Bill Freeman, and Gregory W. Wornell. Co-regularized alignment for unsupervised domain adaptation. In *NeurIPS*, 2018.

[17] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *International Conference on Learning Representations*, 2018.

[18] Pedro H. O. Pinheiro. Unsupervised domain adaptation with similarity learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8004–8013, 2017.

[19] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[20] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1979–1993, 2017.

[21] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018.

[22] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[23] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[24] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[25] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[26] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

[27] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[28] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[29] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[30] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019.

[31] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding, 2019.

[32] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[33] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

[34] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

[36] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

[37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[38] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

[39] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[40] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[41] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[42] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2959–2968, 2019.

[43] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.

[44] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[45] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet classification through label progression, 2018.

[46] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *ICML*, 2018.

[47] A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7130–7138, 2017.

[48] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 1180–1189. JMLR.org, 2015.

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[50] 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.