Rethinking ImageNet Pretraining in Domain Adaptation

Junzhi Ning

February 21, 2022

Abstract

Since the release of Imagenet, thanks to its huge size of labeled images and categories available, the performance of Neural Networks trained on this dataset has enjoyed the significant improvements in terms of their accuracy and generality. Upon the increasing use of pretrained weights, The benchmark of UDA methods also benefit from the initialization weights pf pretrained model on ImageNet. However, much of previous work has yet explored the pretraining impact of the classes available in ImageNet on the problem of Domain Adaptation tasks. In this project, we ignore the classes in ImageNet that are relevant to classes in the dataset on which we evaluate the UDA on and analyze the effect of masking such classes on the results of UDA tasks.

1 Introduction

Unsupervised domain adaptation(UDA) is a study area of transfer learning dedicated to construct the model that is able to achieve the tasks by adapting a target domain from the source domain. Today, two main approaches of UDA are Domain-invariant feature learning and domain mapping. Domain-invariant feature learning[16] aims to align the source and target domain by generating a feature representation common to both domains. This approach typically defines a divergence used to measure the distance between two domains at the level of feature representation, and then attempts to find the most appropriate one that can minimize that divergence. Alternatively, the domain mapping[13]in UDA designs to find the best mapping directly from one domain to another so that the input feature can be mapped into the domain that has known labels to train a classifier.

In recent years, upon the burgeoning development of neural network-based model and the increasing success of Generative Adversarial model[4], the adversarial-based methods have also been introduced into the field of UDA[1, 10, 3]. However, the NN-based models often under-perform in UDA tasks when the training data from the source and target domains is insufficient. Current methods mostly address this problem by using the pre-trained networks from the larger training dataset as the initialization weights, and it serves as a starting point of many existing models to tackle problems in applications. Specifically, in computer vision tasks, pre-trained networks with various architectures from ImageNet[2] are usually preferred. Nonetheless, due to the large number of classes in ImageNet, the overall effect brought by pre-trained networks on the performance of the UDA tasks remains unclear, and it is possible for the pre-trained networks to play a non-trivial role in reducing the domains gap, thereby overestimating the intrinsic functionality of current domain adaptation structure.

A recent study [7] shows that removing a portion of classes in the pretrained model from the ImageNet dataset, up to 20 %, can still achieve comparable or even better performance in transfer learning. However, the experiment from this paper only studies the effect of the number of classes from ImageNet dataset used in the pretrained model, the choice of removing or ignoring certain classes when using the pretrained model weights in specific UDA classification tasks like Office31 dataset has yet been

explored.

Motivated by the above discussion, we will investigate how ImageNet pretraining affect the domain adaptation methods. Particularly, we will look into muting some chosen ImageNet class labels of same or similar types in the UDA benchmark datasets when obtaining the pre-trained model.

2 Related Work and Concepts

In this section, we review basic and important concepts throughout our study and experiments.

2.1 Alex-net

Alexnet which is a name of CNN architecture was proposed by Alex Krizhevsky in the paper[6] of 2012, it participated ImageNet Large Scale Visual Recognition Challenge in 2012 and achieved top-5 error rate of 15.3%.The depth of the alexnet model is key to its high performance. It contains 8 main layers with learn-able weights. The first five layers are Convolutional layers, for each of convolutional layers, it is followed by one RELU activation function and one layer of max-pooling, these five layers serve as the feature extractor of the network. The last three layers are fully connected, and the output layer produces 1000-softmax float values in aim of achieving the property to simulate the probability distribution.

In our experiments, we choose this as the main architecture and we modify the output layer of the Alexnet to adjust the number of masked class labels for imagenet pretraining and then apply it to be the feature extractors of UDA and SSL methods in fine-tuning phase.

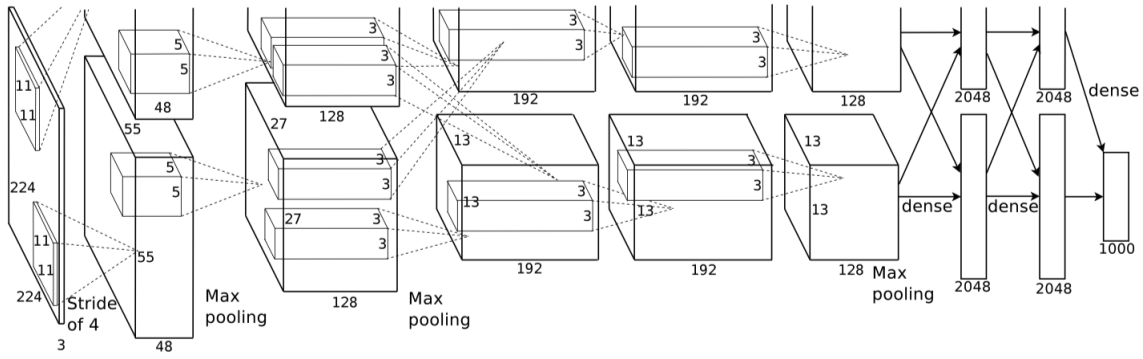

Figure 1: Graphical illustration of Alex-net [6]

2.2 Unsupervised Domain Adaptation (UDA)

In a classification task of UDA[11, 15], given n_s labeled samples as a set $D_s = \{x_j, y_j\}_{j=1}^{n_s}$ from source domain with probability distribution of $P_s(X, Y)$ and n_t unlabeled samples as a set $D_t = \{x_j\}_{j=1}^{n_t}$ from target domain with probability distribution of $P_t(X)$ which is the marginal probability distribution of X without considering the predefined label set Y and we assume $\forall x \in D_t$, each x has an unknown corresponding label $y' \in Y$, the main goal of UDA classification task is to build a classifier C that minimizes the target risk $\mathbb{E}_{(x,y')\sim P_t(X,Y)}|C(x) - y'|$.

2.2.1 Domain-adversarial Neural Network(DANN)

DANN was originally proposed in the paper [3] of 2016 by Yaroslav Ganin and his colleagues to tackle the problem of Domain adaptation. It is based on the main ideas of adversarial-based network and Domain-invariant feature learning. Through an adversarial approach, the feature representation generated by the feature extractor can not be discriminated effectively by the domain classifier, then the label predictor employs the obtained feature representation to produce the corresponding class label for both source and target domains. (See Figure 2)

To be more specific, the feature representation generated by the DANN has the following properties:

- discriminative for the classification task on the source domain.
- indiscriminative with respect to the divergence between the source and target domains.

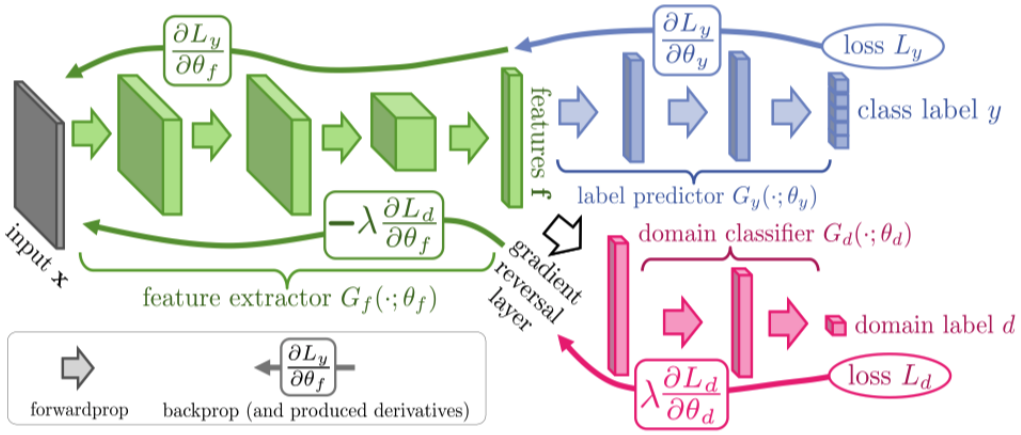

Figure 2: Graphical illustration of DANN [3]

To complement the study of the effect of pretrained network weight on UDA, we also briefly mention the notion of semi-supervised learning in order to see if it has similar scale of dependence on pretrained network in SSL setting, but it will not be the main focus of our experiments.

A research[15] in 2021 suggests that the SSL methods outperform existing UDA methods on the UDA benchmark and therefore should be promoted as baselines in future. Since the research paper also employs the pretrained weights as initialization for training SSL methods, we are also interested in finding out the impact of feature representation of existing classes in pretrained networks on the SSL methods.

2.3 Semi-supervised Learning

In a classification of SSL, given n_l labeled samples as a set $D_l = \{x_i, y_i\}_{i=1}^{n_l}$ and n_u unlabeled samples as a set $D_u = \{x_i\}_{i=1}^{n_u}$ from probability distributions of $P(X)$ and $P(X, Y)$ respectively, typically assuming the size of D_u is less than the size of D_l, the main objective of SSL is to find the a mapping C such that it minimizes the risk $\mathbb{E}_{(X,Y)\sim P(X,Y)}|C(x) - y|$.

2.3.1 Pseudo-label method

The Pseudo-label method was introduced in the paper [8] of the Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks in 2013. It trains the network with labeled and unlabeled data simultaneously. During the training iterations, the unlabeled data will being gradually labeled with the class that has the highest probability and confidence. The measure of confidence and probability can be calculated through applying the Sigmoid Unit or soft-max Unit to the final output. In the original paper, it mentions that The Pseudo-label method in principle equivalent to Entropy Regularization [5] and the resulting effect of minimizing the entropy for unlabeled data exhibits the favour to a low-separation between classes. In our study, we utilize this method to conduct the experiments for the aspect of SSL setting.

3 Experiments

3.1 Brief Summary of Workflow

For the first part of our experiment, we manually scrutinise through the label list of the pre-trained dataset of ILSVRC and attempt to filter out class labels that are similar or exactly matched with categories presented in the dataset of Office31. Then, by ignoring those chosen classes in ILSVRC face-blurred dataset, we adopt Alexnet models to classify the ILSVRC face-blurred dataset. At the same time, the same number of class labels in the ILSVRC face-blurred dataset is randomly chosen to train another alexnet model in order to replicate the training environment of the pretrained model on the chosen masked label list, this serves as a control group to evaluate the difference.

For the second part of our experiment, we adopt two Alexnet models pretrained on the ILSVRC face-blurred dataset as the feature extractors after removing the last few layers of the models. The feature extractors are then embedded into DANN and Pseudo-label methods respectively to perform the fine-tuning in domain adaptation tasks on three domains of Office31. We set most of hyper-parameters according to either the original papers or available machine learning libraries, we will look into details later in this section.

3.2 Datasets

3.2.1 ImageNet (ILSVRC face-blurred)[2]

ImageNet is a dataset with a size of over 14 million labeled images, categorising into more than 20000 classes. The images from the ImageNet dataset were collected from online resources, and they were manually labeled for uses in computer vision applications and research. Since 2010, in light of Pascal Visual Object Classes (VOC) challenge, the competition known as The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) takes place annually. A subset of images from the ImageNet database, consisting of more than 1.2 million training labeled images of 1000 category, roughly 50 thousand validation images and 150 thousand testing images are chosen as their competition datasets.

For the purpose of our experiments, we use the latest version of ILSVRC ImageNet face-blurred dataset for training our pretrained models.

3.2.2 Office31

The office31 is a dataset consisting of 31 object classes in three domains. The 31 object classes are common tools and items in daily office setting. The Amazon domain contains a total of over 2800

Figure 3: Examples of ILSVRC Image-Net face-blurred dataset images

images with a resolution, most images are captured by amazon online merchants. The DSLR domain consists of around 500 images with a high resolution of 4288×2848, each class contains 5 images and each object are captured from different angles. For the Webcam domain, 795 low resolution images are shared across 31 categories with significant noise and color. Due to availability of multiple domains in the dataset, the Office31 are widely adopted in field of transfer learning for comparison purposes and evaluation of model performance.

For the purpose of our experiments, we use Office31 dataset to perform the benchmark evaluation for UDA classification.

(a) Amazon domain (b) Dslr domain (c) Webcam domain

Figure 4: Examples of images from Office31

3.3 Setups

Here, we adopt the public Pytorch library [12] of Image-Net example as a starting point of our experiment. We create three separate datasets from the ILSVRC Image-Net face-blurred dataset of Table 1.
To enhance the models' robustness and generality, we add image transformations to the training images before feeding to the models, these following transformations are Horizontally flipping an image with a given probability, resizing an image of 256 × 256, cropping an image at a random point and resizing it into a size of 224 × 224.

The implementation of Alex-net in our experiments does not follow the default version in pytorch library, but instead we use the modified version from the paper [10], this version of Alexnet is easier

Data-sets	No.classes	No.masked classes	No. Training images	No.validation images
Original	1000	0	1281066	49997
Masked	958	42	1226767	47897
Masked Random	958	42	1227158	47897

Table 1: Masked label list attached in Appendix 7.
Note that every class label has the same number of validation images.

to converge and also covers the implementation of Local Response Normalization layer of which the default version in pytorch omits. Three alexnet models are trained on the three datasets as pretrained models 1 using the following setting 2.

No.Epochs	learning rate	Optimizer	Batch size	Weight decay	LR scheduler
100	0.01	SGD	256	0.0005	Every 30 epochs decay of 0.1

Table 2: Hyperparameters and setting of pre-trained models.
Note that SGD stands for Stochastic Gradient Descent

After obtaining the three pre-trained models of Alex-net, we remove the last fully connected layers of these models as feature extractors(backbone of the new models)to train new DANN and Pseudo-label models using the fine-tuning techniques on six domain adaptation tasks respectively in Office31 dataset: A $\rightarrow$ W, A $\rightarrow$ D, W $\rightarrow$ D, W $\rightarrow$ A, D $\rightarrow$ A, D $\rightarrow$ W. During the fine-tuning phase, we replace the full connected layers in the pretrained models with one bottleneck module and one full connected layer.The learning rate for these new layers is 10 times of the learning rate in the backbone of the new models for DA tasks. In other words, we do not freeze the certain layers in the stage of fine-tuning, rather we update all weights with different learning rates in all new models.

In terms of the Pseudo-label method, we choose the probability-cut-off as 0.95, this means that the unlabeled data x will only pick up the predicted class label as its "true" label if output values of the model after applying the soft-max functions is greater or equal to 0.95.

Apart from performing the fine-tuning on the DANN and Pseudo-label, we conduct the same experiment on the source-only method as a baseline comparison for investigation. Table 3 summarizes our choices of hyperparameters in the fine-tuning phase.

The implementation of DANN and Pseudo-label model are modified based on the public libraries from [15], both of them are under MIT license.

Model types	Source-only	DANN	Pseudo-label
Number of Epochs	50		
Iterations per epoch	1000		
FC layers Learning rate	0.0001	0.01	
Backbone Learning rate	0.00001	0.001	
Optimizer	SGD with a momentum of 0.9		
Batch Size	32		
γ (learning gamma)	0.0003	0.001	
β (learning decay)	0.75		
Learning Rate Scheduler	$LR \times (1 + \gamma \times p)^{-\beta}$		
Weight Decay	0.0005	0.001	

Table 3: Hyper-parameters and setting of Fine-tuning Models
Note that p is the training process from 0 to 1

3.4 Results

Figure 5: Test Loss and Top-1 Accuracy of Pre-trained Models

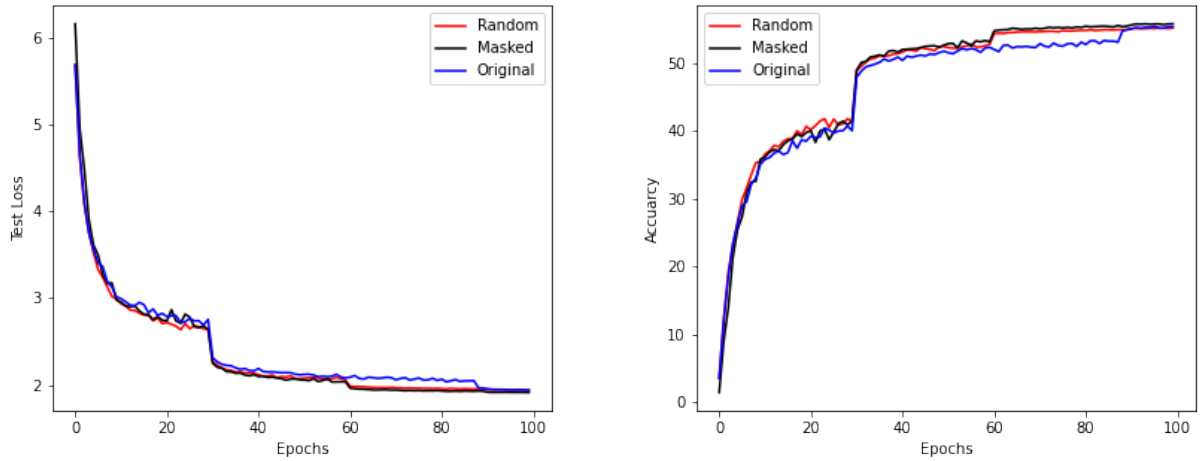

Table 4: Results of Pre-trained Models(Alex-net)

Data-set on which Model trained	Best Top 1 accuracy	Best Top 5 accuracy	Train Loss	Test Loss
Original	55.18	78.27	2.336	1.949
Masked	55.11	78.93	2.331	1.917
Random	55.72	78.41	2.328	1.945

Table 5: Summary of Transductive Results on Office31(Alex-net)

Dataset of pretrained model	Model Type	$A \to D$	$A \to W$	$D \to A$	$D \to W$	$W \to A$	$W \to D$	Avg.
Original	Source-only	44.4	43	29.7	84.8	31.3	90.2	53.9
Masked	Source-only	43.8	41.4	30.8	83.5	28.9	91.8	53.36
Random	Source-only	46.2	44.8	30.2	81.3	29.7	90.4	53.77
Masked	DANN	49.8	52.2	35.5	94.5	38.4	98.6	61.5
Random	DANN	50.6	55.5	36.4	93.6	42.3	98.8	62.86
Masked	Pseudo-label	41.96	44.65	28.8	90.18	31.55	97.38	55.75
Random	Pseudo-label	43.77	43.52	32.44	91.07	32.62	95.78	56.53

Table 6: Summary of Transductive Results at Epoch 1

Dataset of pretrained model	Model Type	$A \to D$	$A \to W$	$D \to A$	$D \to W$	$W \to A$	$W \to D$	Avg.
Masked	Source-only	37.5	33.7	28.1	75.3	22.4	83	46.67
Random	Source-only	35.8	36.5	27.5	73.9	25.5	78.9	46.35
Masked	DANN	51.3	44.4	30.2	89.7	31.2	97	57.3
Random	DANN	51.2	47.2	30.9	89.9	34.9	98.7	58.8

4 Discussions of Results

4.1 Pretrained Models

Pretained models of Alex-net trained on three datasets derived from ILSVRC ImageNet face-blurred dataset converge at 100 epochs with overall top-1 accuracy of around 55 $\pm$ 0.5% and test loss of 1.94 $\pm$ 0.05 %, which are consistent with the results published on paper [14] of top-1 accuracy of 55.83 $\pm$ 0.11 % and top-5 accuracy of 78.55 $\pm$ 0.07 %(See Table 4, Figure 5). However, our results are still behind that of the original paper, but taking into account of non-identical dataset and the coding platform used in implementing this architecture, we believe that there is still a room of improvement of accuracy and test loss after hyper-parameter optimization. The validation accuracy on models with 1000 classes also does not exhibit inferior performance over the models (Random and Masked Datasets) with fewer classes, and it takes more training iterations to reach top-1 accuracy of 55 %.

4.2 Domain Adaptation Tasks

From Table 5 we observe in general that using DANN and source-only methods, the domain adaptation tasks perform better at the pretrained model trained on the random dataset. The average increase in domain transfer accuracy with the respect to the choice of dataset of pretrained models is more obvious to DANN methods, up to 1.36 %. At the more detailed level, the accuracy difference between domains for source-only method does not give a consistent pattern, for example, $W \to D$, $D \to A$ and $D \to W$, these domain adaptation tasks experience a decrease in accuracy when the dataset of pre-trained model changes from Masked to Random. We suspect the results obtained in Table 5 are affected by the certain level of randomness, and it is possible for some models to learn actual features quicker, even though they are slightly different in initialization weights or even have a slight advantage of pretrained weights over others, while fine-tuning the pretrained model weights. Therefore, Transductive Results after fine-tuning might not provide a reliable indicator of the effect of masking relevant classes in pretraining on Domain Adaptation tasks.

To avoid these types of confusing effect in our experiment with a limited modification required on existing setting, we instead examine through the very first training results of Domain Adaptation Tasks(See Table 6).Despite our efforts, we still observe the alternating changing patterns of mean

accuracy on the six domain adaptation tasks.

Looking further into per-class accuracy of several domain adaptation tasks, we found that some class labels in which we attempt to mask produces the opposite changing pattern across the choice of datasets. Here 6 is an example of it. The effect of masking some classes in the pretrained model's dataset does not block the learning of relevant features present in the Office31 dataset used in domain adaptation tasks. We propose two possible reasons for that:

- Masking effect of pretrained models on DA tasks can only be reliably estimated through a considerably large number of repeated experiments.
- Omit or fail to mask some or all relevant class labels in the pretrained dataset that are related to categories in Office31. For instance, should we mask the class label of bottle-cap if we intend to avoid the Neural Network to learn the feature of bottle as well?
- The quality of the pretrained models, DANN and Pseudo-label models underperforms.

	type	Source-only		Dann	
Classes		Mask	Random	Mask	Random
Bike	A-D	76.19	52.38	90	71.4285
Bottle	A-D	12.5	0	50	50
desk_lamp	A-D	28.57	42.85	64	35.7142
keyboard	A-D	60	60	70	80
desktop_computer	A-D	0	0	13	13
monitor	A-D	0	13.63	45	18
Projector	A-D	13.04	13.04	56	17.39
Mug	A-D	37.5	50	62.5	75
Pen	A-D	0	20	100	100
speaker	A-D	34.615	23.0769	23.0769	19
mouse	A-D	66.6	66.6	75	91.6

Figure 6: Examples of A $\rightarrow$ D domain-adaptation task across source-only and dann methods for chosen class labels

4.3 Limitations of our work

Due to the limitation of time constraints, our work is limited to explore one Neural Network architecture, one relatively small DA dataset of Office31 and two DANN and SSL methods on pretraining effect. Much of the work still remains to obtain more statistically reliable estimates of our experiment to draw the strong conclusion in the future.

5 Conclusions

In this work, we investigate the pretraining effect of the choices of classes available in ImageNet dataset with the respect to the domain adaptation tasks. In our experiments, we preliminarily show that masking similar or same classes on the dataset of pretrained model related to Domain adaptation dataset we evaluate on does not have a conspicuous effect in terms of boosting or degrading the performance of domain adaptation tasks measured by accuracy.

6 Acknowledgements

This project was undertaken using Spartan platform [9] of High Performance Computing at the university of Melbourne, I am grateful of the University of Melbourne providing such resources to fellow researchers and students like me.

I would like to express my deep gratitude to Dr Mingming Gong, my supervisor, who kindly provided immerse help and timely guidance over the 8-weeks summer course of this project. Also, I would like to thank Dongting for providing the help when I faced technical difficulties in using HPC .

References

[1] Konstantinos Bousmalis et al. "Unsupervised pixel-level domain adaptation with generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3722–3731.

[2] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[3] Yaroslav Ganin et al. "Domain-adversarial training of neural networks". In: *The journal of machine learning research* 17.1 (2016), pp. 2096–2030.

[4] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[5] Yves Grandvalet and Yoshua Bengio. *Entropy Regularization.* 2006.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[7] Michal Kucer and Diane Oyen. "Transfer learning with fewer ImageNet classes". In: (2021).

[8] 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*. Vol. 3. 2. 2013, p. 896.

[9] Greg Sauter Linh Vu Lev Lafayette and Bernard Meade. "Spartan Performance and Flexibility: An HPC-Cloud Chimera," in: *arXiv preprint arXiv:2103.06191* (2021).

[10] Mingsheng Long et al. "Conditional adversarial domain adaptation". In: *Advances in neural information processing systems* 31 (2018).

[11] Mingsheng Long et al. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[12] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[13] Ashish Shrivastava et al. "Learning from simulated and unsupervised images through adversarial training". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2107–2116.

[14] Kaiyu Yang et al. "A study of face obfuscation in imagenet". In: *arXiv preprint arXiv:2103.06191* (2021).

[15] Yabin Zhang et al. "Semi-supervised models are strong unsupervised domain adaptation learners". In: *arXiv preprint arXiv:2106.00417* (2021).

[16] Han Zhao et al. "On learning invariant representations for domain adaptation". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7523–7532.

Appendix

ImageNet class label	Associated Office31 class label
cup	bottle, mug
comic book	paper notebook, ring-binder
wine bottle	bottle
water bottle	bottle
typewriter keyboard	desktop computer, keyboard, calculator
tray	letter tray
table lamp	desk lamp
spotlight	desk lamp
screen	monitor, desktop computer
rocking chair	desk chair
projector	projector
printer	printer
pop bottle	bottle
pill bottle	bottle
pay phone	phone
notebook	paper notebook
mouse	mouse
mountain bike	bike
moped	bike
monitor	monitor, desktop computer
microphone	headphones, speaker
medicine chest	bookcase, file cabinet
loudspeaker	speaker
laptop	desktop computer, monitor
lampshade	desk lamp
hand-held computer	mobile phone, phone
fountain pen	pen
football helmet	bike-helmet
folding chair	desk chair
dial telephone	calculator, headphone
desktop computer	desktop, computer monitor
crash helmet	bike-helmet
computer keyboard	keyboard, desktop computer
coffee mug	mug, trash can, bottle
china cabinet	file cabinet, book case
cellular telephone	phone, speaker, calculator
bottlecap	bottle
bookcase	bookcase, file cabinet
binder	ring-binder, punchers
bicycle	bike
beer bottle	bottle
barberchair	desk chair

Table 7: Masked label list for ImageNet.