CAFT: Class Aware Frequency Transform for Reducing Domain Gap

Vikash Kumar   Sarthak Srivastava [*]   Rohit Lal [*] Anirban Chakraborty
Department of Computational and Data Sciences
Indian Institute of Science, Bangalore

Abstract

*This work explores the usage of Fourier Transform for reducing the domain gap between the Source (e.g. Synthetic Image) and Target domain (e.g. Real Image) towards solving the Domain Adaptation problem. Most of the Unsupervised Domain Adaptation (UDA) algorithms reduce the global domain shift between labelled Source and unlabelled Target domain by matching the marginal distribution. UDA performance deteriorates for the cases where the domain gap between Source and Target is significant. To improve the overall performance of the existing UDA algorithms the proposed method attempts to bring the Source domain closer to the Target domain with the help of pseudo label based class consistent low-frequency swapping. This traditional image processing technique results in computational efficiency, especially compared to the state-of-the-art deep learning methods that use complex adversarial training. The proposed method **C**lass **A**ware **F**requency **T**ransformation (**CAFT** [1]) can easily be plugged into any existing UDA algorithm to improve its performance. We evaluate CAFT on various domain adaptation datasets and algorithms and have achieved performance gains across all the popular benchmarks.*

1. Introduction

Deep learning has vastly revolutionised the computer vision research. We have witnessed a significant amount of progress since the inception of AlexNet [9] during the ImageNet competition. Deep Learning networks are data-hungry models and require a large number of labelled samples to improve the predictive performance for a given supervised learning task. The generalisation ability of deep features has made it easy to transfer the learning from one task to another with fewer labelled samples from the new task at hand. This method is popularly known as deep transfer learning [20]. In recent times, we witness most of the

state of the art results using different deep learning architectures.

All these learning methods assume that training and test distribution are the same. The generalisation ability of learning algorithms is reported on unseen test datasets having similar distribution as the training dataset. Model performance decreases significantly when the above assumption doesn't hold. For many practical applications, training and test distributions may not be the same. This dissimilarity between training and test distribution is referred to as domain shift or domain gap. We try to reduce this domain gap using Domain Adaptation (DA) algorithms. Training and test distribution are referred to as Source and Target domain, respectively, where the samples in each of the splits are drawn from two different probability distributions. Domain adaptation problem in which we don't have access to any annotation or label information from the Target domain is popularly called Unsupervised Domain Adaptation (UDA) [26]. Data distribution of Source domain and Target domain are different i.e $\mathcal{D}_{\mathcal{S}}(X_s, Y_s) \neq \mathcal{D}_{\mathcal{T}}(X_t, Y_t)$ where $\mathcal{D}_{\mathcal{S}}$ is Source data distribution and $\mathcal{D}_{\mathcal{T}}$ is Target data distribution. We want to build models using widely available labelled Source domain datasets and adapt this learning to Target domain dataset. Our ultimate goal is to bridge the domain shift between Source and Target domain. Therefore, the applications built using the Source dataset can also be deployed in the Target domain where we don't have any access to annotation corresponding to the Target domain.

Most of the UDA methods [2, 19] attempt to reduce the global domain gap by matching marginal Source $\mathcal{D}_{\mathcal{S}}(X_s)$ and Target $\mathcal{D}_{\mathcal{T}}(X_t)$ distribution and assumes that conditional distribution are same i.e $\mathcal{D}_{\mathcal{S}}(Y_s/X_s) = \mathcal{D}_{\mathcal{T}}(Y_t/X_t)$. Since alignment happens between marginal source and target distribution, negative transfer [25] will also take place. Han Zhao Et. Al [29] also shows in their study that marginal alignment of the Source and Target distribution doesn't guarantee the joint alignment of $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$. This results in a sub-optimal solution. Sub-domain adaptation [31] tries global and local alignment to resolve this problem. Difficulty in domain alignment increases as the domain gap between Source and Target increases and vice-versa. Our

---

[*]Equal contribution
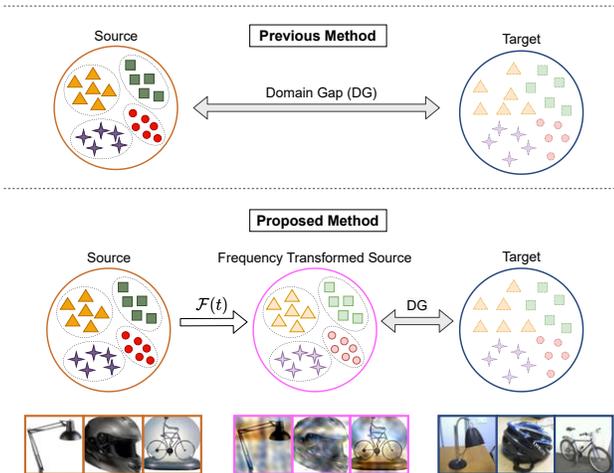[1]Code: https://github.com/vclab-dev/CAFT

Figure 1. **Method Overview**. (*Top Row*) represents general Unsupervised Domain Adaptation (UDA) settings with labelled Source images and unlabelled Target images. UDA algorithms tries to overcome the depicted domain gap. (*Bottom Row*) The proposed method bridges this gap using pseudo label based low-frequency swapping and then applies the UDA methods over this reduced domain gap images.

method tries to bridge the domain gap between Source and Target domain using a class aware frequency transform to circumvent the negative transfer. Fourier Domain Adaptation (FDA) [27] showed that transferring the Target style to Source during adaptation using Fourier transformation bridges the domain gap and improves the adaptation for segmentation task. Directly applying the FDA [27] method for classification causes two problems. Firstly, Target samples are selected randomly, so it is prone to negative transfer because it ignores the fact that within-class domain shift is not the same as global domain shift. Secondly, the transformed samples are prone to have artifacts which result in the poor class-discriminative features. To address the negative transfer and reduce sub-domain gap, we propose pseudo label based class-aware transformation of the Source sample. We also propose retraining of original Source samples and the transformed Source samples to address the second problem. It helps to fill in for the loss of class-discriminative features. Hence, the proposed method explicitly reduces the domain shift. We can visualize the overview of the proposed approach in Fig.1. It will help the adaptation methods improve their performance because they will solve a relatively easier task. In summary, our proposed CAFT framework comprises of the following key components -

- A computationally inexpensive pre-adaptation step that limits the negative transfer and explicitly tries to swap the Source image style with that of the Target image using class aware Fourier transform.
- Pseudo label based class-aware sample selection for

style transfer from Source to Target. It results in sub-domain gap reduction.
- The retention of the original sample along with the transformed sample which helps to account for the lost class-discriminative features during style transfer.

2. Related Work

Unsupervised Domain Adaptation (UDA) methods [26] try to learn domain invariant representation through various methods such as minimizing the divergence, adaptation through reconstruction or adversarial training methods. Techniques for minimizing the divergence includes maximum mean discrepancy (MMD) [5], correlation alignment (CORAL) [18], contrastive domain discrepancy [8] etc. These methods minimize the domain gap by minimizing the distance between the statistics of Source domain distribution and Target domain distribution. These methods are prone to give a sub-optimal solution since they try to align the marginal distribution. DRCN [3] tries adaptation through reconstruction, which looks for the latent domain representation capable of classifying the Source domain and reconstruct the Target and Source domain. It will ensure meaningful latent representation suitable for both Source and Target domain. Adversarial learning based solutions [2, 10, 11] uses a two-player zero-sum game optimization strategy similar to GAN [4]. RevGrad [2] used feature extractor, domain discriminator and classifier. The discriminator tries to distinguish between the Source domain and Target domain. The gradient reversal layer ensures domain invariant representation from the feature extractor.

Image data augmentation is a widely adopted method when it comes to reduce the domain shift that exists between the Source and Target domain. Such methods including norm-VAE [24] and AdaIN [7] try to use learning-based style transfer approach in order to implement augmentation based domain gap reduction. These techniques pose a cost in terms of computational power and time. Due to its dependency on choice of hyperparameter, it is prone to be unstable very difficult to train. Existing GAN based methods like CycleGAN [30] requires substantial computational cost and takes a lot of time to converge. On the other hand, our proposed method CAFT doesn't depend on any trainable parameter and relatively free of hyperparameter. Hence, it is fast and computationally inexpensive.

3. Proposed Method

We present a four-stage algorithm in Figure 2. Stage-1 includes network training using Source samples for few epochs which acts as a classifier for the next stage. In Stage-2, we get the pseudo labels for Target samples using the network obtained from Stage-1. In Stage-3, we do the frequency transformation of Source samples, and finally, the
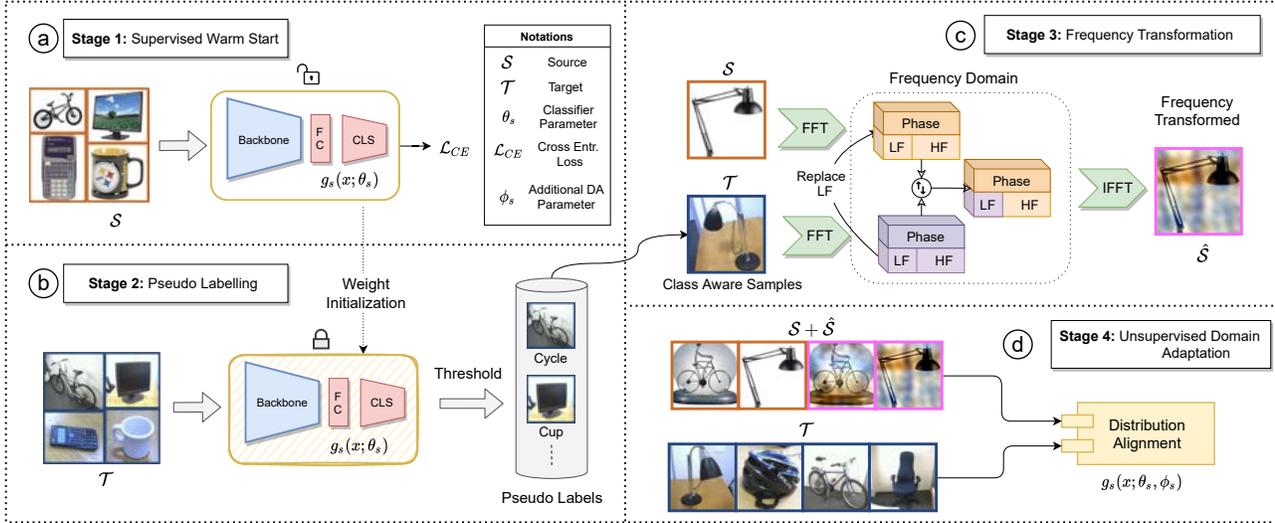
Figure 2. **Architecture of CAFT**. The architecture is divided into four stages. In Stage 1, we train a classifier network using the Source domain and its label. This trained model is then used to generate pseudo labels in Stage 2. We pass the image through the network and select labels whose class confidence value is above a particular threshold. In Stage 3, we transform the source image using frequency domain manipulation with the help of the generated target pseudo labels. The transformed source is closer to the target. The Source $\mathcal{S}$, Transformed Source $\hat{\mathcal{S}}$ (with labels) and Target $\mathcal{T}$ images (unlabelled) are then passed through the Domain Adaptation network in Stage 4.

adaptation happens in Stage-4. We will discuss these stages in detail in the below sub-sections.

3.1. Warm Start of Network

During Stage-1, we train the network $g_s(x_s; \theta_s)$ in a supervised manner as shown in Fig. 2(a) using Source data. Here θ_s represents the network parameter for both backbone and classifier network. We will use this network in the next stage to get the pseudo label for the Target sample.

3.2. Pseudo Labeling

In this iterative stage, we initiate the pseudo label calculation using the trained network in Stage-1. We create a dictionary with key-value pair in the form of a pseudo-class label and samples associated with that pseudo class label. Initially, the pseudo labels will be very noisy. We include only those samples to the dictionary whose prediction probabilities are greater than the pre-defined threshold. Class Probability threshold helps us to filter out the noisy predictions. A maximum number of keys can go up to K, where K is the total number of classes. The quality of pseudo label improves as the training progresses. This is the reason we need to keep updating the dictionary.

3.3. Frequency Transformation of Source Sample

Notation: We have used the following notations. $\mathcal{S}$, $\mathcal{T}$, $\hat{\mathcal{S}}$ are Source domain samples, Target domain samples and frequency transformed Source domain samples respectively. $F[k,l]$, $f[m,n]$, $\mathcal{F}$, $\mathcal{F}^{-1}$ are element in frequency domain at $[k,l]^{th}$ index, element in image space at

$[m,n]^{th}$ index, Fourier transform and inverse Fourier transform respectively.

In general, class-discriminative features are domain invariant because they remain consistent across the domain. On the other hand, domain variant features captures the domain-related information and keep varying across the domain. Domain shift arises mainly due to domain variant features. Image style is one of the domain-dependent components, and hence it also adds to domain shift. In our proposed approach, we explicitly try to swap the Source style to the Target style so that the domain shift between transformed Source and Target is reduced.

FDA [27] tries to minimize the domain gap by replacing the low-frequency component from the Target to the Source. This is because the low-frequency component can be inferred as the style of the domain. The main problem with FDA [27] is a selection of Target samples. It selects a random set of images from the Target domain for performing the style transfer, assuming that style is uniformly distributed across the domain, i.e. $\mathcal{Z} \sim \mathcal{U}(a, b)$ Where $\mathcal{Z}$ is a random variable representing style. Uniform distribution hypothesis for the style doesn't hold true in many practical situation i.e $\mathcal{Z} \nsim \mathcal{U}(a, b)$ which accounts for inter-class variations. However, It is safe to assume that style of intra-class for a particular domain can be represented with uniform distribution i.e $\mathcal{Z}_i \sim \mathcal{U}\{a_i, b_i\}$ for $i = 1$ to K where K is the total number of classes for any domain $\mathcal{D}$. Hence, directly applying the FDA [27] without taking inter-class information into account for solving the classification task is not desirable. We can also infer the same from the Fig. 3.

Algorithm 1: CAFT: Class Aware Frequency Transform for Reducing Domain Gap

---

Stage 1: *Warm Start*
Input: Network parameter $g_s(x; \theta_s)$; Source Data $\mathcal{D}_s$; Maximum Stage-1 Epoch E_1^{max};
while *epoch* $< E_1^{max}$ **do**
    $\{x_i^s, y_i^s\}_{i=1}^B \sim \mathcal{D}_s \triangleright$ *Randomly sample a batch of source image*
    Update θ_s, ϕ_s by minimizing $\mathcal{L}_{CE}$
end
Stage 2: *Pseudo Labelling*
Input: Target domain data $\mathcal{D}_t$; Maximum Epoch E_2^{max}; Pre-trained model for stage-1. Dictionary for mapping from
  pseudo label pl to target sample: $Dict : \{Key : Val\}$
Initialization: Network remains frozen; $g_{tj}(x; \theta_t) \leftarrow g_s(x; \theta_s)$
$\{x_i^t\}_{i=1}^B \sim \mathcal{D}_t \triangleright$ *Sample a batch of target image*
$\{y_i^{pl}\}_{i=1}^B = \arg\max\{g_t(x_i^t; \theta_t) > Threshold\}_{i=1}^B \triangleright$ *select sample pseudo label y^{pl} if score greater than threshold*
Update $Dict : \{Key : Val\}$ with Key $= y^{pl}$, Val = associated target sample
Stage 3: *Frequency Transformation*
Input: $\{x_i^s, y_i^s\}_{i=1}^B \sim \mathcal{D}_s \triangleright$ *Sample a batch of source image*
$\{x_i^t, y_i^{pl}\}_{i=1}^B \sim Dict(Key, Val)$ where $y_i^s = y_i^{pl} \triangleright$ *Sample a source label conditioned batch of target image from*
  dictionary
Find Fourier transform of source sample: $\mathcal{F}(\mathcal{S}) = \mathcal{F}(x_i^s)$
Find Fourier transform of selected target sample: $\mathcal{F}(\mathcal{T}) = \mathcal{F}(x_i^t) \triangleright$ Make Sure that $y_i^s = y_i^{pl} \forall i$ (sample index)
Swap source low frequency with that of target: $\tilde{f}\{\mathcal{F}(\mathcal{S}), \mathcal{F}(\mathcal{T})\} \triangleright$ *Figure 2(c)*
Get the Transfromed source after applying inverse Fourier Transform: $\hat{\mathcal{S}} = \mathcal{F}^{-1}[\tilde{f}\{\mathcal{F}(\mathcal{S}), \mathcal{F}(\mathcal{T})\}] \triangleright$ Refer Eq. 3
Stage 4: *Domain Adaptation*
Input: original Source $\mathcal{S}$; Transformed source $\hat{\mathcal{S}}$; Unlabelled Target $\mathcal{T}$; Adaptation Network Parameter $g_s(x; \theta_s, \phi_s)$
while *epoch* $< E_2^{max}$ **do**
    Update Adaptation network parameter θ_s, ϕ_s using adaptation loss
end
Repeat Stage-2 to 4 for all epochs with updated model parameter
Inference On Trained Model
$\{x_i\}_{i=1}^B \sim \mathcal{D}_t \triangleright$ *Sample a batch of target image*
$\{y_i^{test}\}_{i=1}^B = \arg\max\{g_t(x_i^t; \theta_s, \phi_s)\}_{i=1}^B \triangleright$ get predicted target label

---

From Stage-2, we get the pseudo label for the Target samples. We store Target samples as key-value pairs where key corresponds to the pseudo-class labels. We randomly sample Target images from the stored dictionary based on the Source sample true class label i.e $\mathcal{T}_b \sim \mathcal{T}_{pl}\{X_T, Y_T^{pl}\}$ such that Y_S is equal to Y_T^{pl} for each batch of Source where $\mathcal{T}_b$ is batch of Target domain samples, Y_T^{pl} is Target pseudo label, Y_S is Source true label. It ensures class aware representation of Target during transform. We calculate Fast Fourier transform (FFT) for the Source samples and Target samples for the current batch. FFT algorithm is efficient implementation of Discrete Fourier transform (DFT) whose 2-D expression is represented in equation (1) and its inverse representation is in equation (2) .

$$\mathcal{F} = F(k, l) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[n, m] \exp\{-j2\pi(\frac{kn}{N} + \frac{lm}{M})\}$$

(1)

$$\mathcal{F}^{-1} = f(m, n) = \frac{1}{KL} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} F[k, l] \exp\{j2\pi(\frac{kn}{K} + \frac{lm}{L})\}$$

(2)

Where M and N are number of rows and columns in the 2-D image. $f[n, m]$ is the pixel value at m^{th} row and n^{th} column index. Similarly, $F[k, l]$ is the frequency value at k^{th} row and l^{th} column index. $\mathcal{F}$ and $\mathcal{F}^{-1}$ are Fourier and inverse Fourier transform respectively. Once we calculate the FFT for both Source and Target samples, we need to transfer the style of the Target to the Source, which is present in the form of low-frequency magnitude component. We replace the low-frequency magnitude component of the Source with that of the Target as shown in Fig. 2(c). We call it frequency transformed Source $\hat{\mathcal{S}}$. It ensures a smaller domain shift between $\hat{\mathcal{S}}$ and Target domain $\mathcal{T}$ in image space. Overall, Frequency transformation procedure can be summarized as

follows:
$$\hat{\mathcal{S}} = \mathcal{F}^{-1}[\tilde{f}\{\mathcal{F}(\mathcal{S}), \mathcal{F}(\mathcal{T})\}] \qquad (3)$$

Where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are Fourier transform and inverse Fourier transform respectively. Function $\tilde{f}$ represents the low frequency swapping operation as shown in Fig. 2(c). We name this process as **C**lass **A**ware **F**requency **T**ransformation (CAFT).

3.4. Domain Adaptation

At the start of Stage-4, we have transformed Source samples $\hat{\mathcal{S}}$ along with original Source samples $\mathcal{S}$ and Target samples $\mathcal{T}$. The middle circle in the bottom row of Fig.1 represents the transformed source images. We observe a finite reduction in the domain shift between transformed source sample $\hat{\mathcal{S}}$ and Target sample **T** compared to that between the original Source samples $\mathcal{S}$ and Target sample $\mathcal{T}$. During the transformation process, the $\hat{\mathcal{S}}$ samples also lose their class-discriminative features during inverse Fourier transform. It happens because of the replacement of low-frequency magnitude from the Target samples, which combines with the original phase to provide discriminative feature containing natural images. Hence, some artifacts are observed in the transformed source image. These artifacts hurt the classification performance of the trained model on source and target images. We need to ensure that there is no loss in the class-discriminative features if we want to have a non-decreasing model performance. We include original Source samples along with transformed Source samples during adaptation to solve this problem. Finally, we have $\overline{S} = \mathcal{S} \cup \hat{\mathcal{S}}$ as our new Source domain sample and we perform adaptation between $\overline{S}$ and $\mathcal{T}$ using existing domain adaptation algorithms.

3.5. Theoretical Insights

Let $\mathcal{H}$ be a hypothesis space of VC dimension d, $\mathcal{U}_{\mathcal{S}}$, $\mathcal{U}_{\mathcal{T}}$ are unlabeled samples of size m each, drawn from $\mathcal{D}_{\mathcal{S}}$ and D_T respectively, then theorem 2 of [1] states that for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of the samples), for every $h \in \mathcal{H}$:

$$\xi_T(h) \le \xi_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log 2m + \log\frac{2}{\delta}}{m}} + \lambda \tag{4}$$

where $\xi_T(h)$ and $\xi_S(h)$ are classification error for Target and Source domain samples respectively for the given hypothesis h. $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ is a measure of divergence or domain shift between Source and Target domain. λ is the sum of Source error and Target error w.r.t best available hypothesis $h^* \in \mathcal{H}$

For a given a trained Source model we can safely assume that last two terms of the equation (4) will not change. So, we can define $c = 4\sqrt{\frac{2d\log 2m + \log\frac{2}{\delta}}{m}} + \lambda$. Equation (4) can

be expressed in equation (5).

$$\xi_T(h) \le \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + c \tag{5}$$

In order to have a tighter upper bound, we should have smaller divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$. Domain adaptation algorithm tries to minimize divergence which reduces the Target error $\xi_T(h)$. The proposed approach explicitly tries to minimize the domain shift using class aware style transfer, i.e. $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_{\hat{S}}, \mathcal{U}_T) < \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ where $\mathcal{U}_{\hat{S}}$ is unlabelled transformed source sample. Hence, the proposed method will have a tighter target error upper bound. Hence, applying domain adaptation algorithm on reduced divergence should further minimize the target error $\xi_T(h)$ resulting in improved model performance.

3.6. Algorithm

Refer to Algorithm 1 for complete implementation of CAFT method. Stage wise architecture details is given in Fig. 2.

4. Experiments

The Proposed approach is model agnostic and can be plugged into any adaptation method for improving its performance. We evaluate the proposed method on popular RevGrad [2] and DeepCoral [19] using Office-31 dataset. We also evaluated on DSAN [31] using Office-31 and Office-Home datasets. These are widely reported domain adaptation benchmarks for classification.

4.1. Dataset

Office-31 [16] : It contains 3 domains with 31 classes in each domain. The total number of image samples are 4110 images, and the domains are Amazon (*A*), Webcam (*W*) and DSLR (*D*). Amazon dataset is downloaded from amazon.com, and it has a white background. Webcam and DSLR images are captured in an office environment. The difference in resolution acts as a domain gap for Webcam and DSLR. DSLR images have high resolution, and Webcam images have low resolution. There are 6 possible adaptation setting for this dataset which are $A \to W$, $A \to D$, $D \to A$, $D \to W$, $W \to A$ and $W \to D$.

Office-Home [23] : It contains 4 domains with 65 classes in each domain. The total number of image samples are 15588. 4 domains are Art(*A*), Product(*P*), Real World (*R*) and Clipart (*C*). This dataset is one of the most popular domain adaptation benchmarks. Since it has more number classes, it helps us to evaluate the scalability of our proposed solution.

4.2. Experiment Setup Details

We have used ResNet-50 [6] backbone as feature extractor in all our experiments. Optimisation method used for

Table 1. Accuracy (%) on Office-31 for unsupervised domain adaptation (ResNet50).

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet [6]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DDC [22]	75.8	95.0	98.2	77.5	67.4	64.0	79.7
DAN [12]	83.8	96.8	99.5	78.4	66.7	62.7	81.3
ADDA [21]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN [14]	85.4	97.4	99.8	84.7	68.6	70.0	84.3
MADA [15]	90.0	97.4	99.6	87.8	70.3	66.4	85.2
GTA [17]	89.5	97.9	99.8	87.7	72.8	71.4	86.6
CAN [28]	81.5	98.2	99.7	85.5	65.9	63.4	82.4
iCAN [28]	92.5	98.8	**100.0**	90.1	72.1	69.9	87.2
CDAN [13]	93.1	98.2	**100.0**	89.8	70.1	68.0	86.6
CDAN+E [13]	**94.1**	**98.6**	**100.0**	**92.9**	71.0	69.3	87.7
DSAN [31]	93.6	98.3	**100.0**	90.2	**73.5**	**74.8**	**88.4**
DSAN+CAFT (ours)	92.0	98.0	**100.0**	88.0	**75.0**	75.0	88.0

Table 2. Accuracy (%) on Office-Home for unsupervised domain adaptation (ResNet50 backbone).

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
ResNet [6]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [12]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
JAN [14]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [13]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E [13]	50.7	70.6	**76.0**	57.6	**70.0**	**70.0**	57.4	50.9	77.3	70.9	56.7	81.6	65.8
DSAN [31]	**54.4**	70.8	75.4	**60.4**	67.8	68.0	**62.6**	55.9	78.5	73.8	60.6	**83.1**	67.6
DSAN+CAFT (ours)	**55.2**	69.8	75.0	60.0	72.0	71.0	63.3	57.3	79.1	74.1	60.6	83.0	**68.4**

Table 3. Accuracy on Office-31

Method	A → W	D → W	W → D	Avg
ResNet [6]	68.4	96.7	99.3	88.13
RevGrad [2]	76.4	97.0	100.0	91.13
RevGrad+CAFT (ours)	**79.8**	**98.7**	**100.00**	**92.83**

updating the weights is minibatch stochastic gradient descent with momentum as 0.9 and decreasing learning rate as a function of epoch as shown in equation (6). It was reported in RevGrad [2]:

$$\mu_p = \frac{\mu_0}{(1 + 10p)^\beta} \qquad (6)$$

$$p = \frac{epoch}{total \ number \ of \ epochs} \qquad (7)$$

where $p \in [0, 1]$ and μ_0 as 0.01.

The side of square window size for selecting low-frequency component is $2L * \min(Height, Width)$ and L was chosen to be 0.04 (DSAN, RevGrad) and 0.08 (Deep-Coral). The threshold for assigning pseudo labels was kept at 0.8 and 0.2 for Office-31 and Office-Home datasets, respectively, after analysing one split for a given algorithm and dataset. We use model accuracy for evaluating the proposed approach.

4.3. Results

RevGrad: We show the experiment results for RevGrad [2] in Table 3. The proposed approach improves the existing model accuracy across all the possible splits of Office-31. It results in 1.7% absolute gain in the average accuracy.

DSAN: We show the experiment results for DSAN [31] in Table 1 for Office-31 dataset and in Table 2 for Office-Home dataset. DSAN [31] is one of the state-of-the-art approaches for UDA. The proposed approach improved the accuracy in 8 out of 12 possible splits in for the Office-Home dataset. We get comparable results for the remaining splits as well. It also improves average absolute accuracy by 0.8%. Since the office-home dataset size is large, hence 0.8% gain is non-trivial. We achieve performance gain in 3 out of 6 splits for the Office-31 dataset. Resulting average accuracy is comparable.

DeepCoral: We show the experiment results for the DeepCoral [19] in Table 4. Our proposed approach improves the existing model performance across all the possible splits of Office-31. It results in 1% absolute gain in the average accuracy.
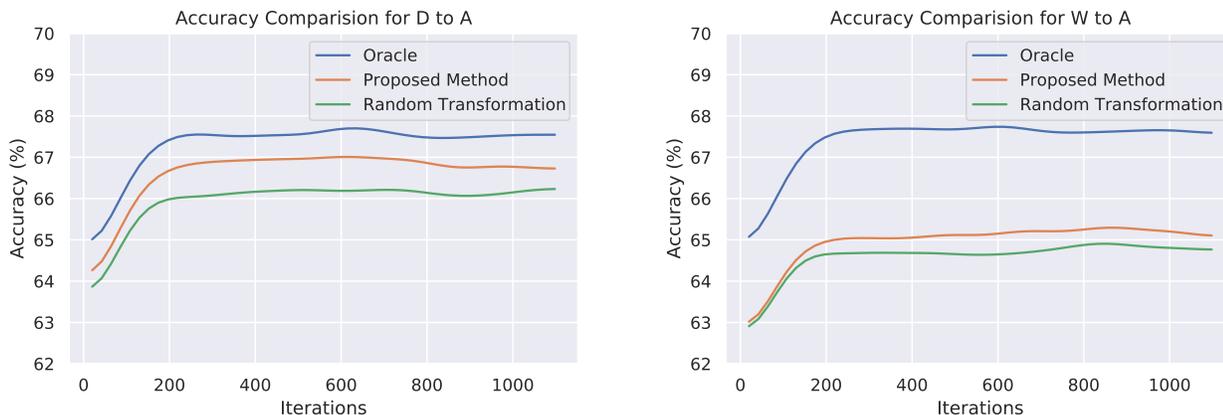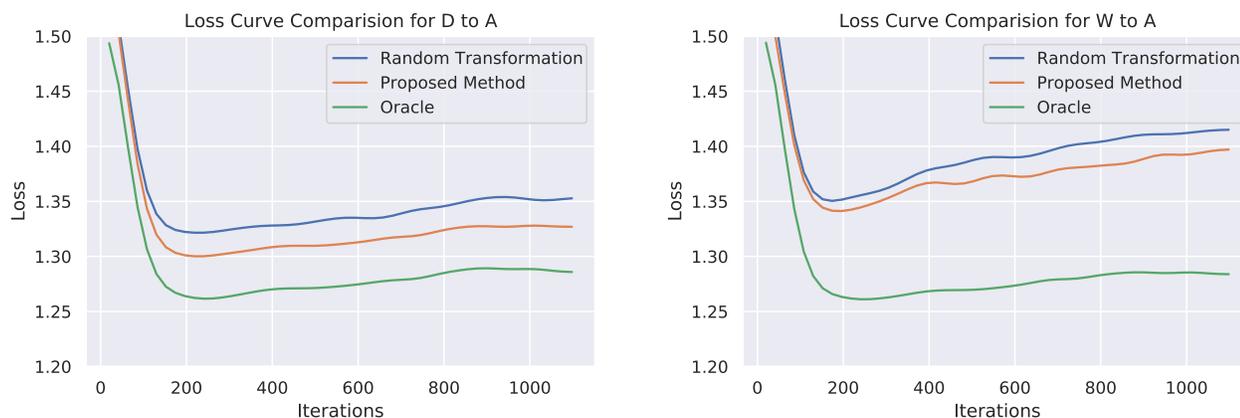
Figure 3. The accuracy plots for Ideal Case Oracle Based Target Labeling against the proposed pseudo label based method and Randomly Selected Image based method for DeepCoral [19]. The superior performance of the proposed method against the randomly selected image based transformation method is evident from the graphs. Due to noisy predictions in the pseudo labels, model performance stays between oracle and random transformation. In oracle case, we assume to have access of the ground truth. In random transformation, we randomly select target samples for CAFT.

Figure 4. The loss plots for Ideal Case Oracle Based Target Labeling against the proposed pseudo label based method and Randomly Selected Image based method for DeepCoral [19]. The convergence of loss for the proposed CAFT approach is faster compared to the random transformation based approach. Oracle converges fastest, which is in line with our proposed hypothesis. The superior performance of the proposed method against the randomly selected image based transformation method is evident from the graphs.

5. Analysis

In this section, we will empirically analyse the proposed approach of class aware Frequency transformation. We will also examine the accuracy and convergence of the proposed approach.

5.1. Analysis of Class Aware Transformation

In our proposed approach, we hypothesise that class aware frequency transformation of the Source sample w.r.t Target sample minimise the domain shift in image space. Hence, it results in ease of adaptation for existing domain adaptation algorithms. Since the Target domain is unlabelled, we use the pseudo label for class aware frequency

transformation. We rely on a Source trained and adapted network to get the pseudo label. Hence, the pseudo label we get is noisy in nature but is better than using a random Target sample for the frequency transformation of the Source sample. We analyse this in Fig. 3(a) and Fig. 3(b) using DSLR$(D) \rightarrow$ Amazon(A) and Webcam$(W) \rightarrow$ Amazon(A) split of Office-31 dataset for DeepCoral [19] method. We observe that pseudo label based transformation results in better accuracy compared to random transformation. We assume that we have access to the actual class label of Target samples (Oracle) and compare the results against the random Target label and pseudo-label-based approach for the analysis purpose. The accuracy of the oracle is highest (as expected) followed by the proposed approach and random

Table 4. Accuracy on Office-31

Method	$A \rightarrow W$	$D \rightarrow W$	$W \rightarrow D$	$A \rightarrow D$	$D \rightarrow A$	$W \rightarrow A$	Avg
ResNet [6]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DeepCoral [19]	77.7	97.6	99.7	81.1	64.6	64.0	80.8
DeepCoral+CAFT (ours)	**79.4**	**97.9**	**100.0**	**81.5**	**66.6**	**65.7**	**81.8**
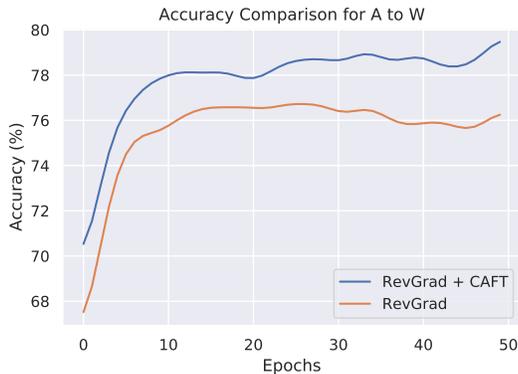
Figure 5. We compare the target domain accuracy from the source (Amazon) to target (Webcam) for adaptation method RevGrad and ours (RevGrad + CAFT). We can see that our approach improved overall accuracy of the adapted model
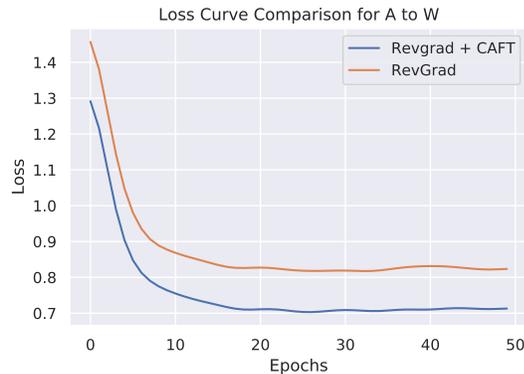

Figure 6. We compare the target domain accuracy from the source (Amazon) to target (Webcam) for adaptation method RevGrad and ours (RevGrad + CAFT). We can see that our approach improved overall accuracy of the adapted model

transformation, which is inline with our proposed hypothesis. We observe the similar trend in the convergence of the loss plot as shown in Fig. 4(a) and Fig. 4(b). Hence, accuracy will further improve if we could improve the pseudo labeling method.

5.2. Accuracy and Convergence on Target Domain

This section analyses the accuracy and loss convergence of the proposed approach compared to the original algorithm. We show the accuracy and loss convergence plot in Fig. (5) and Fig. (6) respectively. We use Amazon(A) $\rightarrow$ Webcam(W) of Office-31 dataset for the RevGrad [2] because this was studied in original work. We can observe that the accuracy curve for the proposed approach stays ahead of our original RevGrad implementation. This is because our proposed approach can ease the adaptation process, which leads to faster convergence. Similarly, we observe from Fig. 6 that the proposed approach converges faster than the original method. These better performances and faster convergence validate our hypothesis that explicitly reducing the domain gap helps to improve and facilitate the adaptation process without requiring extra computing power.

6. Conclusion

This work explored to limit the negative transfer and achieve the explicit domain shift reduction in image space using traditional image processing methods like Fourier transformation. We analysed the effectiveness of Class

Aware Fourier Transform (CAFT) from source samples to target samples with the help of pseudo labels. Along with providing the theoretical insights, we empirically evaluated our proposed hypothesis. The proposed approach yields better or comparable results against the existing domain adaptation baseline methods. CAFT acts as a pre-adaptation step and is independent of the UDA algorithm, utilizing the CAFT transformed source images. Hence it can be plugged into any existing domain adaptation methods easily to improve their existing performance. Unlike adversarial network-based style transfer methods to generate intermediate domain images, our proposed pre-adaptation step is computationally inexpensive and doesn't have many trainable parameters.

Limitations Current method is dependent on accuracy of pseudo labels. A lot of noisy pseudo labels brings down the model's performance due to negative transfer. Additionally, CAFT results in loss of class discriminative features from the source domain images. This leads to a finite decrease in model's predictive performance.

Future Works For the future work, we will investigate the effectiveness of the proposed approach for other tasks such as object detection. Further study can be done on the scalability and robustness of the proposed method across different settings of domain adaptation, such as multi-source domain adaptation and multi-target domain adaptation.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 5

[2] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 1, 2, 5, 6, 8

[3] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014. 2

[5] 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. 2

[6] 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. 5, 6, 8

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2

[8] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 2

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. 1

[10] Jogendra Nath Kundu, Nishank Lakkakula, and R Venkatesh Babu. Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1436–1445, 2019. 2

[11] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2656–2665, 2018. 2

[12] 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. 6

[13] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017. 6

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

[15] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 6

[16] 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. 5

[17] 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. 6

[18] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 2

[19] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 1, 5, 6, 7, 8

[20] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018. 1

[21] 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. 6

[22] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 6

[23] 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. 5

[24] Qian Wang, Fanlin Meng, and Toby P Breckon. Data augmentation with norm-vae for unsupervised domain adaptation. *arXiv preprint arXiv:2012.00848*, 2020. 2

[25] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019. 1

[26] 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. 1, 2

[27] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. 2, 3

[28] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3801–3809, 2018. 6

[29] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019. 1

[30] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

[31] Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2020. 1, 5, 6