FlexAdapt: Flexible Cycle-Consistent Adversarial Domain Adaptation

Akhil Mathur*†, Anton Isopoussu†, Fahim Kawsar†, Nadia B. Berthouze* and Nicholas D. Lane‡
*University College London, UK; †Nokia Bell Labs, UK; ‡University of Oxford, UK

***Abstract*—Unsupervised domain adaptation is emerging as a powerful technique to improve the generalizability of deep learning models to new image domains without using any labeled data in the target domain. In the literature, solutions which perform cross-domain feature-matching (e.g., ADDA), pixel-matching (CycleGAN), and combination of the two (e.g., CyCADA) have been proposed for unsupervised domain adaptation. Many of these approaches make a strong assumption that the source and target label spaces are the same, however in the real-world, this assumption does not hold true. In this paper, we propose a novel solution, FlexAdapt, which extends the state-of-the-art unsupervised domain adaptation approach of CyCADA to scenarios where the label spaces in source and target domains are only partially overlapped. Our solution beats a number of state-of-the-art baseline approaches by as much as 29% in some scenarios, and represent a way forward for applying domain adaptation techniques in the real world.**

I. Introduction

Recent advancement in deep neural networks has led to significant improvements in many perception tasks in machine learning. At the same time, it has been shown that even minor deviations between the training and test data distributions can degrade the performance of deep neural networks [15]. This in turn poses a serious challenge to the generalizability of deep neural networks in real-world scenarios where the existence of such *domain shift* is very likely. Research in the field of *domain adaptation* has taken significant strides towards adapting deep neural network to different but related domains, even in the absence of labeled data from the target domain [4]. Such unsupervised domain adaptation approaches in the area of computer vision broadly follow two approaches. (i) *Feature-level adaptation*, wherein the features extracted from the task-specific network are aligned across the source and target domains by minimizing a distance metric such as the maximum mean discrepancy (MMD) [10]. (ii) *Pixel-level adaptation* involves aligning the source and target distributions in the raw pixel space, by learning a mapping or translation function between the two domains. Recently, a new approach called CyCADA was proposed which combines the feature-level and pixel-level adaptation into a single architecture and provides state-of-the-art results on a number of visual adaptation tasks [7].

State-of-the-art domain adaptation techniques such as CyCADA assume that label spaces across the source and target domains are identical, even though their underlying data distributions might be different. In practice, however, it might be challenging to find source domains which are identical to the intended target domain. For example, as shown in Figure 1, a developer may want to train a model to recognize various everyday objects from a wearable camera. However, as collecting large-scale training data could be expensive, they may seek to adapt an existing dataset (i.e. a source domain) to their target domain. A possible option could be to use large-scale, labeled datasets such as ImageNet-1k or images from Amazon as the *source* domain and transfer their knowledge to the wearable-camera *(target)* domain. However, existing domain adaptation methods do not easily extend to this scenario of *partial domain adaptation* where the source domain (ImageNet-1k in this example) is a super-set of the target domain. More specifically, the presence of outlier classes in the adaptation task can lead to negative transfer [13], a phenomenon wherein the outliers even degrade the transfer of knowledge between shared classes.

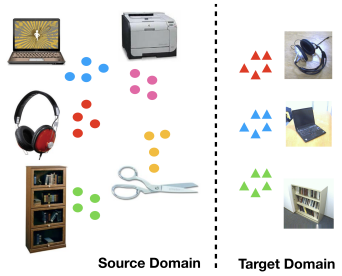

Fig. 1: A typical scenario of partial domain adaptation where the source and target label spaces are not identical. Here the source domain consists of two outlier classes (scissor and printer) which may cause negative transfer in the adaptation task.

The contributions of this paper are two-fold: *firstly*, we present a systematic study to evaluate the performance of CyCADA – which is currently the state-of-the-art domain adaptation technique combining feature-level and pixel-level adaptation – under various scenarios of partial domain adaptation. Our results reveal that label mismatch significantly degrades the accuracy of CyCADA both under pixel and feature adaptation settings (as much as 45% accuracy drop), and in extreme scenarios the performance of CyCADA even falls below that of the source domain classifier. In other words, contrary to its intended goal, CyCADA starts to have negative impact on the model performance in the target domain under extreme scenarios of label mismatch.

Secondly, motivated by these surprising findings, we propose a new end-to-end framework known as FlexAdapt which minimizes the effect of outlier source classes on both feature-level and pixel-level adaptation of CyCADA. We note that

while solutions for partial domain adaptation have been recently studied for feature-alignment architectures [3], there is no solution proposed for complex domain adaptation architectures such as CyCADA which combine feature- and pixel-level adaptation in the same model. In addition, a key design goal for any solution is to minimize its computational overhead on CyCADA, which by itself is a complex architecture consisting of eight deep neural network components and an extremely resource-intensive training process.

Our proposed solution, FlexAdapt, is an intuitive and practical end-to-end framework that is easy-to-integrate with CyCADA with minimal overhead, and addresses the label mismatch challenge by automatically identifying the source outlier classes and re-weighing their contribution in each of the adaptation stages of CyCADA. Experiments on three image adaptation tasks show that our model can outperform CyCADA and other baselines by as much as 29%.

II. Related Work

Research in the field of *domain adaptation* aims to adapt and generalize machine learning models to new domains without requiring extensive labeled data from the target domains [4], [10], [3], [7], [6], [5], [15]. Domain adaptation has been studied both in supervised and unsupervised settings; in the supervised setting, a small amount of labeled data is used to guide the adaptation process [8], [11]. The unsupervised setting [5], [15] however is more practical, albeit challenging, as acquiring labeled data in the real-world is expensive.

This paper focuses on the task of unsupervised domain adaptation. In the area of computer vision, unsupervised adaptation techniques fall under two categories: a) *Feature-level adaptation*, wherein the features of the task networks are aligned across the source and target domains by minimizing a distance metric such as the maximum mean discrepancy (MMD) [10]; and (b) *pixel-level adaptation* wherein the underlying data distributions of source and target distributions are aligned in the pixel space. [7] proposed CyCADA, a combination of feature-level and pixel-level adaptation into a single architecture and showed state-of-the-art performance on a number of visual adaptation tasks.

In particular, we study the performance of CyCADA under the scenario of partial domain adaptation, wherein the source domain consists of classes that are not present in the target. This scenario leads to a known phenomenon of negative transfer [13], [14], [2]. Recently, [3] have studied this problem in the context of feature-alignment techniques such as ADDA and showed promising results. In this work, we aim to explore partial domain adaptation for more complex, hybrid architectures such as CyCADA and propose a novel architecture which generalizes CyCADA to real-world scenarios of non-identical source and target label spaces.

III. FlexAdapt

In this section, we present *FlexAdapt*, our solution to solving a domain adaptation problem *where the source and target label spaces are not identical, and the target label space is not known*. To this end, we first review the construction of cycle-consistent adversarial domain adaptation [7] (CyCADA) which our system extends. We then review the partial domain adaptation (PDA) problem setting, and analyse the effects of label set discrepancy on CyCADA to motivate our approach. In particular, we focus on the case where the source dataset has a bigger label space than the target. This is a typical scenario in many image domain adaptation tasks, as well as personalization tasks, where data from many users is adapted to a personalized model for a single user. In the latter scenario, it is particularly advantageous to allow for an unknown target label space since each users requirements can vary, and explicitly requesting that information from the user each time a model is trained may lead to a bad user experience.

A. Cycle-Consistent Adversarial Domain Adaptation

The idea of incorporating pixel-level and feature-level alignment into a single architecture for unsupervised domain adaptation was recently proposed by [7]. The proposed solution CyCADA combines CycleGANs [16] with ADDA (adversarial domain adaptation) [15] and outperforms a number of baselines which only do a single type of adaptation. We now briefly discuss the optimization objective of CyCADA and refer the readers to the original paper for more details.

CyCADA optimizes five type of losses in a single architecture that forces a model to learn representations in the target domain which are in several ways consistent with representations in a pre-trained model on the source domain. We denote the labeled source distribution by (X_S, Y_S) and the unlabeled target distribution by X_T, and samples from the two distributions by (x_s, y_x) and target data by (x_t). Further, $G_{\mathrm{S}\to\mathrm{T}}$, D_T, f_S, and f_T refer to the generator from source to target domain, the target domain discriminator, the pre-trained source classifier and the target model respectively. The loss function used for the CyCADA target classifier is composed of the following five losses.

Task Loss: For a K-way classification, it denotes the cross-entropy with respect to the softmax function σ of the target classifier against the source labels.

$$\begin{aligned}&\mathcal{L}_{\text{task}}\left(f_T, X_S, Y_S\right)\\&= -\mathbb{E}_{x_s\sim X_s}\sum_{k=1}^{K}\mathbb{1}_{[k=y_s]}\log\sigma\left(f_T^{(k)}\left(G_{\mathrm{S}\to\mathrm{T}}(x_s)\right)\right)\end{aligned}$$

Adversarial Loss: It denotes the domain adversarial loss on the translation functions and is primarily intended to enforce pixel-level alignment across image domains.

$$\begin{aligned}&\mathcal{L}_{\text{GAN}}\left(G_{\mathrm{S}\to\mathrm{T}}, D_T, X_T, X_S\right) = \mathbb{E}_{x_t\sim X_T}\left[\log D_T(x_t)\right]\\&+\mathbb{E}_{x_s\sim X_s}\left[\log\left(1 - D_T\left(G_{\mathrm{S}\to\mathrm{T}}\left(x_s\right)\right)\right)\right]\end{aligned}$$

Cycle Loss: This imposes a L1 penalty on the reconstruction error in a cycle and enforces the two generators to be inverse

of each other.

$$\begin{aligned}&\mathcal{L}_{\text{Cycle}}\left(G_{\text{S}\to\text{T}}, G_{\text{T}\to\text{S}}, X_S, X_T\right)\\&=\mathbb{E}_{x_s\sim X_s}\left[\|G_{\text{T}\to\text{S}}(G_{\text{S}\to\text{T}}(x_s))-x_s\|\right]\\&+\mathbb{E}_{x_t\sim X_T}\left[\|G_{\text{S}\to\text{T}}(G_{\text{T}\to\text{S}}(x_t))-x_t\|\right]\end{aligned}$$

Semantic Loss: This loss encourages high semantic consistency before and after image translation, by minimizing the cross entropy between translated and untranslated data.

$$\begin{aligned}&\mathcal{L}_{\text{sem}}\left(G_{\text{S}\to\text{T}}, G_{\text{T}\to\text{S}}, X_S, X_T, f_S\right)\\&=\mathcal{L}_{\text{task}}\left(f_S, G_{\text{T}\to\text{S}}\left(X_T\right), p\left(f_S, X_T\right)\right)\\&+\mathcal{L}_{\text{task}}\left(f_S, G_{\text{T}\to\text{S}}\left(X_S\right), p\left(f_S, X_S\right)\right)\end{aligned}$$

Feature Loss: Finally, in addition to the pixel-level losses above, CyCADA also optimizes a feature-space loss based on the principle of adversarial learning. Here D_{feat} refers to a domain discriminator for source and target features.

$$\mathcal{L}_{\text{feat}}=\mathcal{L}_{\text{GAN}}\left(f_T, D_{feat}, f_S(G_{\text{S}\to\text{T}}(X_S)), X_T\right))$$

By jointly optimizing these losses, CyCADA learns a classifier on the target domain without the need for labeled data.

B. Negative Transfer from Source Outliers

While the CyCADA construction has the apparent assumption that a natural bijection between the source and target data manifolds should exist, its empirical evaluation has shown that it can perform domain adaptation even when there are drastic changes in image manifolds (e.g., between Street View House Numbers and MNIST datasets). However, there are other categories of domain shift which may still hamper the performance of domain adaptation techniques:

- Strong source outliers: The source domain contains samples whose labels are outside the target label set, i.e. belonging to $\mathcal{C}_S \backslash \mathcal{C}_T$, where C_S and C_T denote the source and target label spaces.
- Strong target outliers: The target domain contains labels outside the source label set, i.e. belonging to $\mathcal{C}_T \setminus \mathcal{C}_S$.
- Weak source and target outliers: Even when the source and target domain share the same label space, there may be some samples in source domain which have no natural pairing in the target domain. For example, a real world digit recognition dataset may contain non-arabic numerals. They are not outliers in the semantic sense, but it is unrealistic to expect completely unsupervised adaptation methods to work.

In this work, we focus on the challenge of *strong source outliers* and study how they affect the performance of domain adaptation. This setting is known as 'partial domain adaptation', as we are interested in transferring the knowledge from a subset of (overlapping) source classes to the target domain. We will call source and target samples which are not strong outliers as T-regular and S-regular respectively. Further, we denote the probability mass of the T-regular subset by

$$r(S,T)=\mathbb{P}_S\left(x_s \text{ is regular}\right). \tag{1}$$

Intuitively, as the number of source outlier classes increase, probability mass of the T-regular subset $r(S,T)$ decreases.

The presence of source outlier classes in the adaptation task can lead to negative transfer [13], a phenomenon wherein the outliers even degrade the transfer of knowledge between shared classes. Let us illustrate this by showing how the learning theoretical bound fail in the case of partial domain adaptation. Ben-David et al. [1] established the following bound for the target domain error $Err_T(h)$ under domain adaptation:

$$\text{Err}_T(h)\leq \text{Err}_S(h)+\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(S,T)+\lambda_H(S,T), \tag{2}$$

where $\text{Err}_S(h)$ and $\text{Err}_T(h)$ are the classification errors of a classifier $h\in H$ on domains S and T, $\lambda_H(S,T)$ is the minimum joint classification error, and

$$d_{\mathcal{H}\Delta\mathcal{H}}(S,T)=2\sup_{h,h'\in\mathcal{H}}\left|\mathbb{P}_S\left(h\neq h'\right)-\mathbb{P}_T\left(h\neq h'\right)\right| \tag{3}$$

is the $\mathcal{H}$-discrepancy between S and T. The bound was generalised to domain adversarial learning in [5], namely the discrepancy can be written as follows and is maximised by the discriminator loss.

$$d_{\mathcal{H}\Delta\mathcal{H}}(S,T)\leq 2\sup_{D_{\text{feat}}}\left|\mathbb{P}_S(D_{\text{feat}}=1)+\mathbb{P}_T(D_{\text{feat}}=0)-1\right| \tag{4}$$

Now we show the effect of source outlier classes on the domain discrepancy (Eq. 4) which bounds the target error. By partitioning the source domain into T-regular and T-outlier subsets, S_T^{reg} and S_T^{out}, respectively, we can write the upper bound in Equation (4) as

$$\begin{aligned}&2\sup_{D_{\text{feat}}}\Big|r(S,T)\mathbb{P}_{S_T^{\text{reg}}}\left(D_{\text{feat}}=1\right)+\mathbb{P}_T\left(D_{\text{feat}}=0\right)\\&+\left(1-r(S,T)\right)\mathbb{P}_{S_T^{\text{out}}}\left(D_{\text{feat}}=1\right)-1\Big|.\end{aligned} \tag{5}$$

As the regular fraction $r(S,T)$ of the training set becomes smaller (i.e., number of source outlier increases), the outlier term

$$\left(1-r(S,T)\right)\mathbb{P}_{S_T^{\text{out}}}\left(D_{\text{feat}}=1\right), \tag{6}$$

begins to dominate the discriminator loss. This leads to a lack of convergence, where the optimization of feature encoders and the domain discriminator both focus on the outlier subset and thereby fail to effectively perform the transfer of shared information across domains. Later in Section IV, we empirically show that this phenomenon is observed even for modest values of $r(S,T)$, which we vary between 1 and 0.5.

C. Partial Cycle-Consistent Domain Adaptation

When there is a mismatch between source and target label spaces, existing domain adaptation techniques fail to generalize because of the negative transfer caused by the source outlier classes $\mathcal{C}_S \setminus \mathcal{C}_T$. Intuitively, as the source outlier space grows (which is likely in practice), the adverse effect of negative transfer is also expected to increase. Indeed, this has been identified as a key problem with feature-space domain

adaptation [3]. However, it remains unclear how this solution extends to hybrid architectures such as CyCADA which jointly adapt models in the pixel and feature-space.

Algorithm 1: Partial Cycle-Consistent Domain Adaptation

Result: f_T: An adaptation of the source model f_S for the target domain.

Input : f_S: Pre-trained source domain classifier
(X_s, Y_s): Labeled data from the source domain
X_t: Unlabeled data from the target domain
N: Number of training epochs

Condition: $C_t \subseteq C_s$: Target label space is a subset of the source label space.

Output : f_T: Target domain classifier.

Initialization: $f_T \leftarrow f_S$, $\Upsilon \leftarrow$ Tensor of ones, where $|\Upsilon| = C_s$
for $epoch \leftarrow 1$ **to** N **do**
  $\Upsilon \leftarrow f_S(X_t)$;
  $\Upsilon \leftarrow \Upsilon$ / $max(\Upsilon)$
  foreach *Minibatch* (x_s, y_s), x_t *in* (X_s, Y_s), X_t **do**
    $\tau \leftarrow$ assign a weight of $\Upsilon[y_s]$ to each source sample in the minibatch ;
    Compute $\mathcal{L}_{\text{GAN}}(\text{S} \rightarrow \text{T})$;
    Compute $\mathcal{L}_{\text{semantic}}(\text{S} \rightarrow \text{T})$;
    Compute $\mathcal{L}_{\text{cycle}}(\text{S} \rightarrow \text{T})$;
    Compute $\mathcal{L}_{\text{feature}}(\text{S} \rightarrow \text{T})$;
    Compute $\mathcal{L}_{\text{task\_weighted}}(\text{S} \rightarrow \text{T})$;
    $\mathcal{L}_{\text{GAN}}(\text{S} \rightarrow \text{T}) \leftarrow \tau * \mathcal{L}_{\text{GAN}}(\text{S} \rightarrow \text{T})$
    $\mathcal{L}_{\text{semantic}}(\text{S} \rightarrow \text{T}) \leftarrow \tau * \mathcal{L}_{\text{semantic}}(\text{S} \rightarrow \text{T})$
    $\mathcal{L}_{\text{cycle}}(\text{S} \rightarrow \text{T}) \leftarrow \tau * \mathcal{L}_{\text{cycle}}(\text{S} \rightarrow \text{T})$
    $\mathcal{L}_{\text{feature}}(\text{S} \rightarrow \text{T}) \leftarrow \tau * \mathcal{L}_{\text{feature}}(\text{S} \rightarrow \text{T})$
    Optimize $G_{\text{S}\rightarrow\text{T}}, D_S, G_{\text{T}\rightarrow\text{S}}, D_T, f_T$
    //Generate a batch of fake target samples
    $x_s^{'} \leftarrow G_{\text{S}\rightarrow\text{T}}(x_s)$
    Optimize f_S on ($x_s^{'}$, y_s)
  end
end

At a high-level, our idea is to downweigh the contribution of source outlier classes in the adaptation process. In the context of CyCADA, this raises two challenges: (i) identifying the source outlier classes and (ii) developing an algorithm to adapt CyCADA's loss functions $(\mathcal{L}_{\text{task}}, \mathcal{L}_{\text{GAN}}, \mathcal{L}_{\text{sem}}, \mathcal{L}_{\text{cycle}}, \mathcal{L}_{\text{feat}})$ in the forward and reverse cycles.

Note that as the target label space C_T is not known during training, it is not trivial to find the source outlier classes $\mathcal{C}_S \setminus \mathcal{C}_T$. To identify them, we use the intuition that when a data sample x_i is passed to the source classifier f_S, it outputs a probability distribution over the source label space C_S, denoting the likelihood of x_i belonging to various source classes [3]. Therefore, when a sample x_t from the target domain (belonging to target label space C_t) is passed to the source classifier, we expect low probability outputs corresponding to the source outlier classes $\mathcal{C}_S \setminus \mathcal{C}_T$ and higher probabilities for the shared classes $\mathcal{C}_S \cap \mathcal{C}_T$. These probabilities outputs could be used as class-specific weights to reweigh the contribution of each source class in the adaptation process.

However, as the target data belongs to a different domain, output class probabilities from the source classifier are inherently noisy – therefore, as shown in Equation 7, we compute the per-class weights Υ by averaging the output probabilities over the entire target data to reduce the effect of a noisy classifier. Secondly, we iteratively update the source classifier during the training process by generating fake target samples $x_s^{'}$ from the generator $G_{\text{S}\rightarrow\text{T}}$ and optimizing f_S on the $(x_s^{'}, y_s)$ pair. Further details are provided in Algorithm 1 and Figure 2.

$$\Upsilon = \frac{1}{|X_t|} \sum_{i=1}^{|X_t|} f_S(x_t^i) \tag{7}$$

The training process of FlexAdapt works as follows: at the start of each training epoch, we compute the per-class weights Υ over the source classes using Equation 7 and normalize them. Thereafter, we sample a minibatch of labeled source data (x_s, y_s) and the unlabeled target data x_t and compute weights of each sample in the minibatch based on its source class y_s using $\Upsilon[y_s]$. The losses over the entire batch are computed and then reweighed based on their respective source instances, for example, the losses corresponding to source outlier classes are assigned lower weights whereas losses for shared source and target classes are given higher weights. For the task loss $\mathcal{L}_{\text{task}}$, we use the weighted cross-entropy loss instead of the standard cross-entropy loss. The updated losses are then used to optimize the generators, discriminators, and output domain model f_T. Finally, we fine-tune the source classifier using fake target samples as a way to improve its accuracy in generating probability distributions on target data as shown in Equation 7. All other operations in the CyCADA architecture, including the reverse cycle ($\text{T} \rightarrow \text{S}$) losses remain unchanged.

By automatically re-weighing the contribution of the outliers and shared classes in the adaptation process, FlexAdapt mitigates negative transfer due to label mismatch and also enables transfer of relevant knowledge in the shared label space.

IV. Experiments

We evaluate FlexAdapt on three domain adaptation tasks, in 16 different combinations of source and target label spaces.

Setup. We conduct experiments on three digit classification datasets, namely MNIST [9], USPS and SVHN [12]; each dataset consists of 10 digit classes ranging from 0-9. More specifically, FlexAdapt is evaluated for the following adaptation tasks: *USPS to MNIST*, *MNIST to USPS*, and *SVHN to MNIST*. To evaluate the scenario where the target label space is a subset of the source label space, i.e., $C_T \subseteq C_S$, we systematically reduce the number of classes in the target domain while keeping the number of source classes fixed to 10. Table I shows the various experimental settings, e.g., the $10 \rightarrow 5$ experiment setting denotes 10 classes in the source domain and 5 classes in the target domain. Note that due to

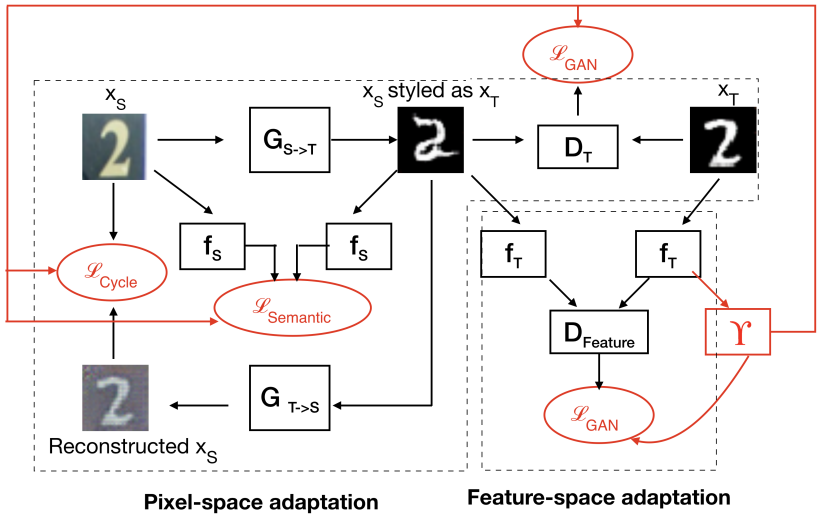

Fig. 2: Architecture of FlexAdapt to extend CyCADA to scenarios of partial domain adaptation. The red parts denote the modifications that FlexAdapt proposes over the original CyCADA architecture. Υ represents the per-class weights and is used to adapt the various losses in the architecture. Task loss $\mathcal{L}_{\text{task}}$ and its adaptation are not shown in the figure.

resource limitations, we do not evaluate all combinations of classes in the target domains. For each experiment setting, we randomly choose three combinations of source and target classes and present their results.

Baselines and Implementation. For all datasets, we use the standard training sets for adaptation, with no label information from the target domain in the training process. The test set of the target domain is used to compute the final classification accuracy, which is the metric we use to compare our approach with baseline techniques. We evaluate FlexAdapt against 4 baseline unsupervised domain adaptation approaches.

As FlexAdapt uses both pixel-level and feature-level adaptation, we compare it against a pixel-adaptation baseline (CycleGAN) [16], a feature adaptation baseline (ADDA) [15], and against CyCADA's hybrid adaptation architecture. To the best of our knowledge, there are no prior solutions for partial adaptation of hybrid-architecture such as CyCADA, therefore we compare our results against PADA, which is a partial domain adaptation algorithm for feature-adaptation architectures [3]. Like FlexAdapt, PADA also operates on the same principle of reweighing the contribution of outliers in the feature adaptation process.

Experiment Setting ($S \rightarrow T$)	Target Classes	Experiment Setting ($S \rightarrow T$)	Target Classes
$10 \rightarrow 10$	0,1,2,3,4,5,6,7,8,9	$10 \rightarrow 7$	1,2,4,5,7,8,9 0,2,3,5,6,8,9 0,1,3,4,6,7,8
$10 \rightarrow 9$	0,2,3,4,5,6,7,8,9 0,1,2,4,5,6,7,8,9 0,1,2,3,4,6,7,8,9	$10 \rightarrow 6$	1,2,3,6,7,8 0,2,5,6,8,9 0,1,3,4,7,9
$10 \rightarrow 8$	0,1,2,3,5,7,8,9 0,1,3,4,5,6,8,9 0,1,2,3,4,5,6,7	$10 \rightarrow 5$	1,3,5,7,9 0,2,4,6,8 0,3,4,7,8

TABLE I: Experiment settings to simulate source and target label mismatch. The first column $S \rightarrow T$ refers to the number of classes in the source domain and target domain respectively. All 10 digit classes (0-9) are used in the source domain.

For a fair comparison across methods, we use the network architectures proposed in CyCADA paper for the generator, discriminator and task network in all our experiments. For CycleGAN, CyCADA, and FlexAdapt, we train the generators and discriminators for 50 epochs with batch size of 100 and learning rate of 2e-4. For the feature adaptation methods, we train for 200 epochs with a learning rate of 1e-5. With the exception of FlexAdapt and PADA where losses are adapted for partial domain transfer, all other techniques use the standard losses with equal weighing. Finally, as FlexAdapt makes adaptations to CyCADA both in the pixel-space and feature-space, we also report the performance of each of these adaptations separately.

Results Our findings are shown in Tables II, III, and IV. Interestingly, we observe that under scenarios of label mismatch, CyCADA shows a trend of higher accuracy losses than ADDA and CycleGAN. In other words, label mismatch is an even severe problem for hybrid adaptation architectures such as CyCADA. Further, we observe that FlexAdapt outperforms all adaptation baselines in most of the label mismatch scenarios, more so as the amount of label mismatch increases. In scenarios with 0 or minimal label mismatch, our results are similar to existing domain adaptation baselines. For example, in the USPS→MNIST experiment with 5 source outlier classes, the performance of CyCADA drops to 51.33% which is significantly lower than the source classifier itself. This demonstrates one example of severe negative transfer that may happen due to source outliers. FlexAdapt, on the other hand, achieves an accuracy of 80.1% which is 14% higher than the source classifier and 29% higher than CyCADA.

Further, in Figure 3, we plot the performance of CyCADA and FlexAdapt as the source outlier classes increase. We observe a significant difference in the performance as the number of outlier classes increase, thereby establishing the promise of FlexAdapt.

Method	10 → 10	10 → 9	10 → 8	10 → 7	10 → 6	10 → 5
Source-only	70.1	65.7	68	65.33	63.0	66
ADDA	89.1	**90.0**	75.6	71.4	56.3	34.6
PADA	88.9	80.0	71.5	68	64.3	52.3
CycleGAN	95.3	78.6	80.6	79.3	58.4	57.3
CyCADA	**96.4**	89.0	79.5	73.4	58.33	51.33
FlexAdapt-pixel	96.3	80.6	79	79.5	76	75.3
FlexAdapt-feature	96.3	**90.0**	84.5	86.2	72.7	72.7
FlexAdapt-all	**96.4**	89.7	**84.7**	**89.3**	**77.8**	**80.1**

TABLE II: Results of the USPS→MNIST adaptation experiment

Method	10 → 10	10 → 9	10 → 8	10 → 7	10 → 6	10 → 5
Source-only	82.2	73.0	76.7	75.6	74.7	73.4
ADDA	90	**89.4**	81	76.2	48.3	40.4
PADA	88	87.3	86.6	79	78.6	75
CycleGAN	95.6	86	88.6	87.4	82	76.3
CyCADA	95.6	87.7	78.6	71	57.1	54
FlexAdapt-pixel	95.1	84	87.3	86.7	85.3	80
FlexAdapt-feature	95.7	89	87.3	90	83.4	78.2
FlexAdapt-all	**95.7**	89.2	**87.4**	**90**	**86.3**	**83.3**

TABLE III: Results of the MNIST→USPS adaptation experiment

V. Discussion and Future Work

Our results clearly demonstrate the adverse effect of negative transfer caused by source outlier classes on various domain adaptation techniques, and show that FlexAdapt can mitigate this negative transfer to a large extent. As a future work, we plan to extend our evaluation to larger vision datasets (e.g., ImageNet-1k) and other modalities (e.g., audio).

While we believe that our study is an important first step towards generalizing CyCADA to real-world scenarios, there still remain a number of related challenges. In addition to the source outlier problem that we studied, another equally important problem is of open-set domain adaptation, i.e., how to make domain adaptation techniques work in the presence of target outlier classes. Further, it is also possible that even when the label spaces are identical, there may not always be a natural pairing between all source and target classes, which may again lead to weak negative transfer. As a future work, we plan to investigate these challenges in the context of CyCADA and other domain adaptation architectures.

VI. Conclusion

We presented a solution to generalize the state-of-the-art unsupervised domain adaptation method namely CyCADA to

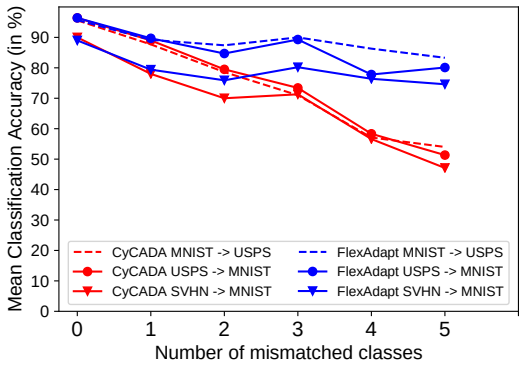

Fig. 3: Comparison of CyCADA and FlexAdapt as the number of source outlier classes are increased.

Method	10 → 10	10 → 9	10 → 8	10 → 7	10 → 6	10 → 5
Source-only	67.1	69	71	66.5	68.1	67.3
ADDA	75.2	75.7	66	67	43.4	43
PADA	77	74.6	68.2	71	70.1	69.2
CycleGAN	70.1	69.3	66	70	66.3	63.3
CyCADA	**90**	78	70	71.3	56.6	47.1
FlexAdapt-pixel	83	70	71	74.2	70.1	68
FlexAdapt-feature	87	79.2	73.7	78.2	73	70.7
FlexAdapt-all	89	**79.4**	**75.9**	**80.2**	**76.4**	**74.6**

TABLE IV: Results of the SVHN→MNIST adaptation experiment

scenarios where the source and target label spaces are not identical. Our proposed solution downweighs the contribution of the source outliers, thereby reducing the effect of negative transfer in the adaptation process. Through adaptation experiments on three datasets, we show that FlexAdapt outperforms CyCADA and a number of other domain adaptation baselines. In summary, this work makes a significant contribution to the vision of taking domain adaptation solutions to the real-world, wherein it is not always feasible to have identical label spaces between source and target domains.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, 2010.

[2] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[3] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in *ECCV*. Springer, 2018.

[4] H. Daume III and D. Marcu, "Domain adaptation for statistical classifiers," *Journal of artificial Intelligence research*, vol. 26, 2006.

[5] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, 2016.

[7] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, 2018.

[8] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," in *Proceedings of CVPR*, 2017.

[9] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[10] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *arXiv preprint arXiv:1502.02791*, 2015.

[11] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, 2010.

[14] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *NIPS 2005 workshop on transfer learning*, vol. 898, 2005.

[15] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.