Improving Semi-Supervised Domain Adaptation Using Effective Target Selection and Semantics

Anurag Singh[1]*, Naren Doraiswamy[1]*, Sawa Takamuku[2], Megh Bhalerao[1], Titir Dutta[1], Soma Biswas[1], Aditya Chepuri[3]
Balasubramanian Vengatesan[3], Naotake Natori[2]. (* denotes equal contribution)
[1]Indian Institute of Science, Bangalore, [2]Aisin Corporation, Japan [†], [3]Aisin Automotive Haryana Pvt. Ltd, Bangalore[†]

Abstract

Recently, semi-supervised domain adaptation (SSDA) approaches have shown impressive performance for the domain adaptation task. They effectively utilize few labeled target samples along with the unlabeled data to account for the distribution shift across the source and target domains. In this work, we make three-fold contributions, concentrating on the role of target samples and semantics for the SSDA task. First, we observe that choosing a few, but an equal number of labeled samples from each class in the target domain requires a significant amount of manual effort. To address this, we propose an active learning-based framework by modeling both the sample diversity and the classifier uncertainty. By utilizing k-means initialized cluster centers for picking a small pool of diverse unlabeled target samples, we compute a novel classifier adaptation uncertainty term to select the most effective samples from this pool, which are queried to obtain their true labels from an oracle. Second, we propose to weigh the hard target samples more, without explicitly using their predicted, possibly incorrect labels, which guides the adaptation process. Third, we note that irrespective of the domain shift, the semantics of the classes remain unchanged, so they can be effectively utilized for this task. We show that initializing the class-representations or prototypes with the class-semantics helps in bridging the domain gap significantly. These along with adversarially learnt entropy objective results in a novel framework, termed STar (Select TARgets), which sets a new state-of-the-art for the SSDA task.

1. Introduction

With the availability of large scale labeled data and computational resources, deep neural networks have shown impressive performance on various computer vision tasks such as, image classification [14], retrieval [11], segmentation [17], etc. However, when tested on data with a different distribution, these models often fail to generalize well, leading to a sharp decrease in their performance. For most real-life scenarios, the data used for training the model (source domain) and the test data (target domain) follow different distributions (termed as domain shift), which results in this performance degradation. Thus, for deploying a trained neural network in real world, it is extremely important for this model to be able to adapt to the new domain, which is the goal of domain adaptation (DA) [32, 2]. Unsupervised DA aims to learn a classifier using label rich source domain data and then adapt it to a related but unlabeled target domain. Although reasonably successful, these approaches suffer from bias towards the task specific boundaries on the source domain data, due to its label-based strong supervision [28]. Hence, to calculate the disagreement between the source and target distributions for improving the generalization of the network, a small amount of labeled target data is very useful. Recently, semi-supervised DA (SSDA) [26, 13, 24], where a few (1-3 samples per class) labeled samples from the target domain are utilised along with the remaining unlabeled target data have received significant attention and have been used to tackle the conditional shift problem to a certain extent.

In this work, we propose a novel approach focusing on the importance of target sample selection and class semantics for the SSDA task. First, we make the following observations: (1) Since very few labeled samples from the target domain are being utilised, it is important that these samples are effective for addressing the domain shift. However, random selection of such target samples may not ensure the same and may also result in outlier selection; (2) Though few in number, to select equal number of samples from each class requires an expert to manually go through a significant portion of the data, or requires collecting samples from all the classes, both of which require considerable manual effort. To address these, we propose a novel technique to automatically select the target samples [31, 35], which are better suited to the DA task and thus, worth the effort of manual annotation. As in the SSDA approaches, we work in the very low data regime, i.e. the annotation budget is

[†] This work is partly supported through a research grant from Aisin.

less than or equal to 3 samples per class. We aim to select the effective target samples on the basis of their diversity, as well as the classifier uncertainty for the domain adaptation task. First, the unlabeled target pool is grouped into clusters using the k-means algorithm. Observing that not all classes get adapted at the same pace, we propose a novel *classifier adaptation uncertainty* for each class, and leverage it to select the most effective target samples from the diverse target pool obtained in the first step, for annotation by the oracle.

Second, we note that the target samples which are similar to the source domain are easily and correctly classified. So in order to facilitate the domain adaptation process, the target samples which are farther away from the source samples and are likely to be incorrectly labeled should be given more weightage. So we propose to weigh the entropy term for the difficult samples, without explicitly using their predicted labels, which may be incorrect.

Finally, we observe that though the domain changes, class-semantics remain unchanged across domains, and this can be utilized as a unifying factor for the different domains. For this, we propose to initialize the domain-invariant class-representatives which can classify data from either source or target, with semantic-attributes of that class. These attributes can be automatically obtained from the class labels using Word2Vec [19] or GloVe [23] embeddings. This not only helps to make the class-representatives semantically meaningful, but also helps to maintain the inter-class distances, in addition to aiding the domain adaptation.

Incorporating all these techniques with an adversarially trained entropy objective gives the proposed framework, termed as STar (**S**elect **Tar**get). It is tested extensively on three benchmark datasets, and achieves the state-of-the-art results on all of them. The contributions in this work are:
(1) We propose a novel *active learning based* SSDA framework, to identify the most effective samples from the target domain and utilize their labels to improve the generalization of the classifier.
(2) We propose a novel *classifier adaptation uncertainty* for each class, by observing that the rate of domain adaptation varies across the different classes.
(3) We propose to weigh the hard target samples, without explicitly using their predicted, possibly incorrect labels.
(4) We incorporate semantic initialization to unify the different domains for effective domain adaptation.
(5) We show the effectiveness of our method for 3 datasets, namely, DomainNet, Office-Home and Office-31.

2. Related Work

Here, we discuss the related work in DA, active learning in DA and approaches which utilize semantic attributes.

Domain Adaptation: Initial works in unsupervised domain adaptation (UDA) utilized diverse statistical methods like Mean Maximum Discrepancy (MMD) [29], Correlation Alignment (CORAL) [37] and Geodesic Flow Kernel [9] to address the feature distribution mismatch. Ganin *et al.* [8] proposed domain adversarial training to bring the feature distributions of both domains closer and many recent works in UDA [39, 33, 43] follow up on this work. However this approach does not consider the class-wise conditional distribution shift and approaches trying to address this have been recently proposed, few notable ones being MCD [28] and ADR [27]. Recently, several SSDA methods [26, 13, 24] have been proposed to develop a better generalized classifier. MME [26] uses minimax adversarial training to push the target features towards the corresponding source features, while APE [13] addresses the intra-domain discrepancies among the target distributions. BiAT [24] tries to generate adversarial training samples which can fill the gap between the two domain distributions. All these approaches utilize few labeled samples from each class, which requires significant manual effort. In this work, we address the SSDA task, but let the algorithm automatically choose which target samples should be annotated for effective DA.

Active learning for DA: Different types of active learning approaches have been proposed. In pool-based techniques, uncertainty sampling [1], diversity sampling [30] and combination of both [20] are the most popular algorithms. In uncertainty sampling, the least confident queries are chosen [40]. The most simplest, yet effective method in diverse sampling is clustering [12] the feature space of the sample pool to select the query samples. Sener *et al.* [30] used diverse samples picked through maximizing the Euclidean distance among the pool of unlabeled samples and querying labels for the same. Recently, variational autoencoder based active learning framework has been used for selecting the most informative samples [35, 44].

Classical machine learning techniques have showcased incorporating active learning the DA scenario. [4] used importance weighting [3] and selected target samples with larger distances between features while training using MMD [29]. [36] proposed an active learning method using H-divergence and importance sampling to query the target instances. Unlike these approaches, we work in the low data regime, where we select very few target samples to be labeled, which are effective for the DA task.

Class Semantics or Attributes: Several computer vision tasks like zero-shot learning [7, 41, 6], etc. have benefited immensely from using semantics of the classes or attributes. Since manual attributes are expensive to obtain, usually pre-trained word embedding models are used like Glove [23] and Word2Vec [19]; which can be used to directly obtain the embeddings using just the given class names. To the best of our knowledge, this is the first work which uses semantic attribute initialization for DA task.

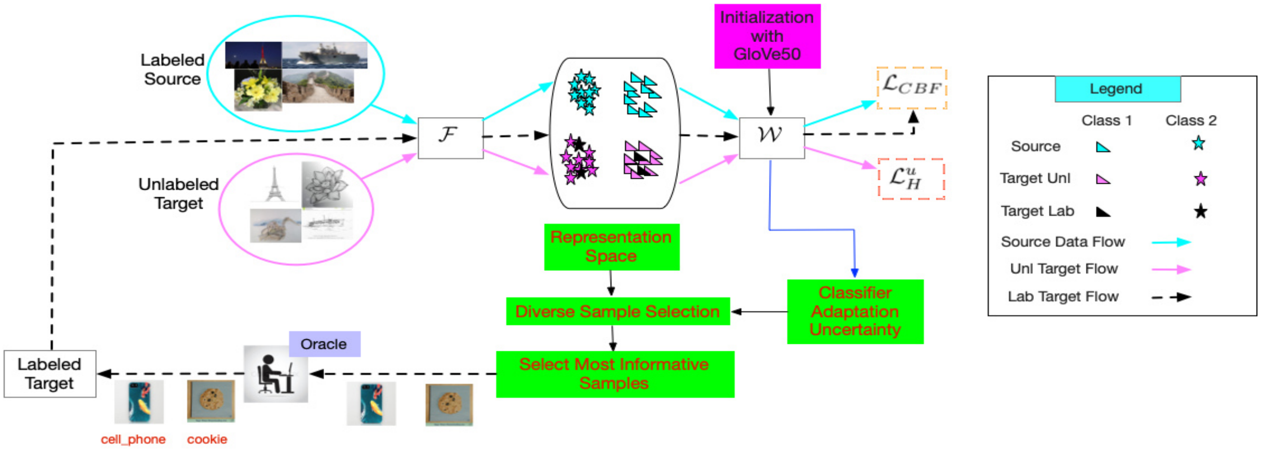

Figure 1: STar framework. $\mathcal{F}$ and $\mathcal{W}$ represent the feature-extractor and the classifier weights respectively. We query labels for the most informative target samples and use them to adapt the class-prototypes. $\mathcal{W}$ is initialized using the semantic information. The labeled data is trained using CB-Focal ($\mathcal{L}_{CBF}$) and unlabelled target data using weighted entropy ($\mathcal{L}_H^u$).

3. Problem Definition and Notations

Here, we address the SSDA task, but unlike other approaches proposed for this problem, where the labeled target samples are given apriori, we propose to automatically select them for improved DA. Towards this goal, we propose a novel algorithm, STar to select a few unlabeled target samples, which if labeled, can effectively improve the DA task. In UDA, the training data consists of labeled source data $\mathcal{D}_s = \{x_i^s, y_i^s\}$, $i = 1, 2, ..., N_s$, and unlabeled target data $\mathcal{D}_{ut} = \{x_i^{ut}\}$ where $i = 1, 2, ..., N_{ut}$. Here, x_i is the input data, and $y_i \in \{1, ..., C\}$ denotes its class label, where C is the total number of classes, assumed to be same for source and target domains. The proposed approach is used to query the most informative samples from this unlabeled pool $\mathcal{D}_{ut}$ during the DA process at regular intervals. These queried samples and their labels provided by the oracle are the labeled target data for SSDA denoted by $\mathcal{D}_{lt} = \{\mathbf{x}_i^{lt}, y_i^{lt}\}$ where $i = 1, 2, \ldots N_{lt}$. As these labeled targets are obtained, they are removed from $\mathcal{D}_{ut}$ and infused into DA training with the additional label information.

4. Proposed STar Algorithm

The proposed approach is based on computing a domain-invariant representation (class-prototype) for each class as in several other DA approaches [42, 26]. The distance between such class-prototypes (initially computed from the labeled sources) and the target samples are minimized strategically towards the goal of efficient adaptation. The proposed approach makes three main contributions: (A) Automatic selection of target samples to be labeled by the oracle; (B) Automatic selection of hard target examples to be weighted for efficient DA and (C) Semantic initialization to unify the domains using their common semantics. We will now describe each of these three contributions in details.

4.1. Selection of target samples to be labeled

For any classifier, its weights can be considered as the class-representatives or prototypes. In this work, we choose the target samples during the DA process in an online manner, as the network adapts the class-prototypes computed from source samples towards target data distribution. Inspired by the rich literature in active learning, we aim to choose target samples, which are not only diverse and representative of the target distribution, but also more informative. Here, since the final goal is domain adaptation, these factors have to be considered in the context of the adaptation process. Based on the above, the proposed STar chooses target samples based on the following two criteria:
(1) **Classifier adaptation uncertainty**: We observe that all classes do not adapt to their corresponding target domain at the same pace, and some classes adapt at a slower pace than the others. The novel *classifier adaptation uncertainty* ensures that during target selection, more emphasis is given to the classes for which the estimated prototypes are highly uncertain at that stage of training.
(2) **Diversity of selected samples** - This ensures that the selected samples from the pool of unlabeled target are diverse and represent the target distribution well, which is very important for successful adaptation.

Budget consideration: In SSDA, few samples per class (between 1-3) are utilized to aid the adaptation process. To showcase that actively queried target samples can effectively help in the adaptation process, we query labels for samples with the same budget constraint as that in the manually labeled SSDA setting. Thus, the number of targets which are selected, i.e. budget is $\mathcal{B} = K * C$, where

K denotes the number of samples per class used by SSDA approaches. For example, the budget is $\mathcal{B} = 3C$ for 3-shot setting, where 3 labeled targets per class are provided for training. Here, the sample selection module of the STar algorithm is run K-times interactively with the remaining modules, and each time we select C samples from the target domain for labeling by the oracle. Next, we provide detailed explanation of the proposed sample selection module.

Classifier adaptation uncertainty: For the DA task, the class-prototypes initially computed using the labeled source samples slowly get adapted to the target domain. We assume that at a certain stage of training, the behaviour of the class-prototypes gives an indication of the class-wise adaptation process. Intuitively, we aim to select more samples from the classes, whose prototypes have not yet adapted well to the corresponding target data distribution. Towards this goal, we introduce a novel measure, *classifier adaptation uncertainty* to detect those classes for which the adaptation is not yet satisfactory.

Let us denote the set of class-prototypes (i.e., classifier weights) at t^{th} instant (here, it implies iteration) by $W_t = \{w_t^1, \ldots, w_t^C\} \in \mathcal{R}^{d \times C}$. Here, $w_t^c \in \mathcal{R}^d$ is the prototype for c^{th} class at instant t, where $c \in \{1, \ldots, C\}$. To compute the classifier adaptation uncertainty for each class, we first compute the Euclidean distance between corresponding class-prototypes over subsequent instances as adaptation progresses. The distance traversed by the c^{th} class prototype in the feature space between instances $t-1$ to t is computed as

$$\hat{d}_t^c = \sqrt{||w_t^c - w_{(t-1)}^c||^2} \tag{1}$$

To make this distance computation robust, we compute it for L times, and compute the moving average of the same as follows

$$d_t^c = \alpha \hat{d}_t^c + (1-\alpha) d_{(t-1)}^c \tag{2}$$

Here α is a hyper-parameter, whose value is determined experimentally and is maintained between $0.5 < \alpha < 1$, to ensure that the relative change in the prototype locations at the current iteration gets the highest attention.

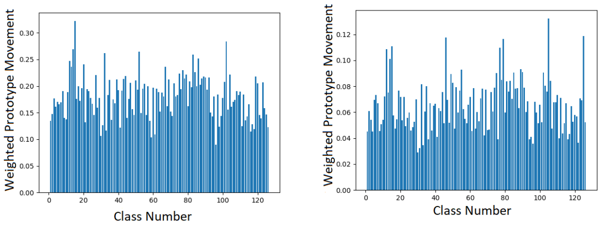

Figure 2: Movement of class prototypes as domain adaptation progresses from 2500^{th} (left) to 7500^{th} (right) iteration.

Let T denote the instant at which one set of target samples are selected for labeling, Let the weighted measure of change in prototype location computed at instant T for all the classes is given by $D_T^c = \{d_T^1, \ldots, d_T^C\}$. This distance measure indicates how the classifier weights or prototypes of all the classes are adapting to the target domain. From the distance measures of all the 126 classes in DomainNet data [22], at two different instances in the adaptation process in Fig. 2, we observe that it varies for different classes and also decreases gradually as adaptation progresses. At instant T, a higher value of d_T^c indicates that the c^{th} class-prototype has higher movement, which in turn implies that the class prototype is still uncertain and complete domain adaptation has not taken place. Therefore, selecting samples from these classes would be more beneficial for the adaptation process. To ensure this, we find the classes which have higher values of d_T^c. First, the mean value of the weighted distance across all the classes is calculated as $m_T = \text{mean}(d_T^1, ..., d_T^C)$. For classes with weighted distance greater than m_T, we normalize d_T^c with the calculated mean-value m_T, and we define the classifier adaptation uncertainty of that class as

$$\begin{aligned} \beta_u^c &= \frac{d_T^c}{m_T}, \qquad \text{if} \qquad d_T^c > m_T \\ &= 1 \qquad \text{otherwise} \end{aligned} \tag{3}$$

For classes with the weighted distance measure lesser than the mean, the adaptability rate is considered to be 1. This ensures that though the classes with higher uncertainty are given higher weightage for selecting target samples, we would also like to select samples from the other classes for successful domain adaptation. We will discuss how this uncertainty factor is utilized to automatically select target samples to be labeled in details later.

Diverse sample selection: In addition to selecting samples from the classes with higher uncertainty in its prototype position, we also aim to select samples which capture the diversity in the target domain for better generalization. Towards that goal, we select an initial pool of diverse unlabeled target samples from the dataset $\mathcal{D}_{ut}$ with k-means. Specifically, given the entire unlabeled target data $\mathcal{D}_{ut}$, we initially group all these samples into $N_{cluster} = 2C$ number of clusters. As mentioned before, at each instant, since we aim to pick C number of target samples to be labeled, we choose more number of clusters, so that we can pick the most informative amongst them. The intuition behind doing so is that during the adaptation process, the target samples which are easily adaptable would have drifted towards the source prototype, while the target samples which have higher distribution shift would not have drifted towards the source clusters. To accommodate the intra-class variation present during the adaptation process, we pick $2C$ number of diverse samples using k-means. Thus, the diverse candidate unlabeled target sample pool is given by the target samples closest to the cluster centers.

$$\mathcal{X}_{div}^{ut} = \{x_{i,div}^{ut}\}_{i=1}^{2C} \in \mathcal{R}^{2C\times d} \tag{4}$$

Final selection of unlabeled targets: Finally, we aim to select target samples from the diverse candidate list $\mathcal{X}_{div}^{ut}$, which also account for the classifier adaptation uncertainty (i.e. have higher uncertainty), and thus are best suited for DA. At a particular instant T when the target selection is taking place, the predicted class of $x_{i,div}^{ut} \in \mathcal{X}_{div}^{ut}$, is obtained by identifying its nearest class-prototype $w_T^{\bar{c}}$. Let us denote the distance with the predicted class $\bar{c}$ is given by $d(w_T^{\bar{c}}, x_{i,div}^{ut})$. Assuming that the predicted classes are correct, the most uncertain ones are those samples which are the farthest from the predicted class prototypes. If the predicted class is wrong, the distance of these samples from the correct class prototype is even larger, again implying that they are difficult samples. In addition, we account for the domain adaptation uncertainty of the classes by normalizing the distance of the sample from its nearest prototype by the classifier adaptation uncertainty of that class. Thus the final uncertainty of the sample $x_{i,div}^{ut}$ at instant T, in the initially selected diverse pool is given by:

$$d_T(x_{i,div}^{ut}) = \beta_u^{\bar{c}} * d(w_T^{\bar{c}}, x_{i,div}^{ut}) \tag{5}$$

Next, we sort all the $2C$ samples in $\mathcal{X}_{div}^{ut}$ according to their final uncertainty measure as computed above, and select the C samples with highest uncertainty measure. This set of selected samples at T^{th} instant is referred to as D_{lt}. These selected samples are not only diverse in nature, but are also better suited to improve the domain adaptation task. Once the labels are obtained for these target samples from the oracle, the training process of the remaining modules of the STar framework continues with labeled source $\mathcal{D}_s$, actively acquired labeled targets $\mathcal{D}_{lt}$ and the remaining unlabeled target data $\mathcal{D}_{ut}$. Note that the samples for which the labels are queried are removed from the unlabeled target data pool.

4.2. Weighting Hard Target Samples

It is well known that giving more weight to difficult training samples using techniques like [16], makes classifiers more discriminative. Here, we propose to weigh both hard source & target examples. This helps in learning discriminative classifiers, and the hard targets will help guide the DA as well. For the source data, noting that majority of the real-world data [26, 25] are imbalanced, we use a recent variant of focal loss, namely class-balanced focal loss (CB-Focal loss) [5], which additionally incorporates class imbalance information for improved performance as follows:

$$\mathcal{L}_{CBF} = -\mathbb{E}_{(x,y)\in\mathcal{D}_s\cup\mathcal{D}_{lt}} \frac{1-\delta_1}{1-\delta_1^{n_c}} \sum_{c=1}^{C} (1-p_c)^{\gamma_1} log(p_c) \tag{6}$$

where n_c is the total number of labeled training source and target samples and p_c is probability that the sample belongs to class c, δ_1 and γ_1 are hyper-parameters to be empirically determined. Here, we remove the explicit sample indices to avoid notational clutter. It is not straightforward to compute a similar loss on the unlabeled target data, since it requires the knowledge of the class labels. But since the final classification is on the target domain, incorporating such a loss for the target is very important.

One approach is to predict the labels of the targets and either consider these hard labels or compute soft labels, which are then used for computing this loss. Though this idea has been successfully used in some works [45, 46, 15, 34], we did not apply it for our problem for two reasons, (1) In general, since easier samples are correctly classified in comparison to the difficult ones, so label prediction has to be done in multiple stages from easy to difficult. We observe that predicting the labels of the target samples repeatedly increases the computational complexity of the framework significantly; (2) Also, we are more interested in predicting the labels of the difficult targets, which is hard, since these samples are likely to be incorrectly classified. Taking these factors into account, we initially let the model adapt to the target domain by only incorporating CB-focal loss for the labeled samples, by running it for sufficient number of iterations (eg. 8000 iterations for DomainNet data). This is followed by an inference step, i.e. predicting the labels of the unlabeled targets. The predicted label of an unlabeled target example $x_i^{ut} \in \mathcal{D}_{ut}$ is the class whose prototype is the closest with the target feature. We define the class prediction uncertainty of the target sample as follows

$$uc_i^{ut} = 1 - \max_{1\leq c\leq C} p_c^{ut} \tag{7}$$

As in majority of DA approaches, the proposed STar framework also uses entropy loss for the unlabeled targets to ensure that they move closer to one of the class prototypes. Inspired by the focal loss, we propose to weigh the entropy of the unlabeled targets based on their prediction uncertainty. We weight the uncertain unlabeled targets more, to ensure that they are better aligned with one of the class prototypes. However, in order to be confident in the predictions we let the classifier adapt to unlabelled target for τ iterations before we start weighing uncertain examples as follows

$$\alpha_i^{ut} = \begin{cases} (uc_i^{ut})^{\gamma_2} + \delta_2 & \text{if } T > \tau \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

Here, γ_2 and δ_2 are hyper-parameters to be empirically determined. This implies, that the more uncertain samples will have higher weight α_i, and vice versa. Incorporating this weight, the new entropy loss takes the following form:

$$\mathcal{L}_H^u = -\mathbb{E}_{x_i^{ut} \in \mathcal{D}_{ut}} \alpha_i^{ut} \sum_{c=1}^{C} p_c log(p_c) \tag{9}$$

We observe that for SSDA, this works significantly better than directly using the predicted class of the target samples with focal-loss.

4.3. Semantic Initialization

Semantic information represents a domain invariant attribute for samples of both source and target distributions. Since our goal is to compute domain-invariant class prototypes, we propose to initialize them with the class name embeddings, and we observe experimentally that this significantly helps in adaptation. This not only makes the latent space semantically meaningful, but also makes the computation of the prototype movements (1) very efficient, by reducing the latent space dimension (equal to the dimension of the semantic attributes).

The semantic information can be obtained using pre-trained word representations like Word2Vec [19] or GloVe [23]. Here, we use GloVe 50-d embeddings of the class names to initialize the prototypes. Given the class names in the source domain, these embeddings can be automatically generated with no additional information.

4.4. Complete STar framework for SSDA

The proposed STar framework (Fig. 1) consists of a feature extractor $\mathcal{F}$ and a classifier $\mathcal{W}$, with learnable parameters denoted by θ_F and θ_W respectively. We use adversarial entropy minimization objective [26] on the unlabeled target data $\mathcal{D}_{ut}$ to train the feature extractor and classifier in an adversarial fashion. The entropy maximization step drives the prototypes to move towards the unlabeled target features to avoid over-fitting the prototypes to labeled source data, while the entropy minimization step on the feature extractor clusters the unlabeled target features around these prototypes. We use semantic initialization as a means of domain invariant prototype initialization. In this work, since there are no labeled target samples at the beginning, we start the adaptation in an unsupervised manner using the labeled source data $\mathcal{D}_s$ and unlabeled target data $\mathcal{D}_{ut}$. The domain-invariant class prototypes are first estimated by extracting the discriminative features from the labeled source data. When the actively selected target samples along with their labels are acquired at instant T, they are moved from $\mathcal{D}_{ut}$ to $\mathcal{D}_{lt}$, and the final training objective is as follows:

$$\begin{aligned} \hat{\theta}_F &= \underset{\theta_F}{\text{argmin}}\, \mathcal{L}_{CBF} + \lambda \mathcal{L}_H^u \\ \dot{\theta}_W &= \underset{\theta_W}{\text{argmin}}\, \mathcal{L}_{CBF} - \lambda \mathcal{L}_H^u \end{aligned} \tag{10}$$

Algorithm 1: Proposed STar algorithm.

```
Input: Base network with feature extractor $\mathcal{F}$, classifier
  $\mathcal{W}$, labeled source data $\mathcal{D}_s$, unlabeled target data $\mathcal{D}_{ut}$,
  labeled target data initialized as $\mathcal{D}_{lt} = \{\phi\}$, update
  iteration $T$, budget $\mathcal{B}$, total training iteration $N_{iter}$.
  Semantic initialization of class prototypes.
Repeat
    • Train $\mathcal{F}$ & $\mathcal{W}$ using $\mathcal{D}_S$, $\mathcal{D}_{lt}$ & $\mathcal{D}_{ut}$, minimizing (10).
    • Extract class prototypes and compute $d_t^c$ according to (2)
      iteratively till update iteration $T$.
    • At update iteration $T$, obtain the classifier adaptation
      uncertainty $\beta_u^c$ according to (3).
    • Perform k-means sampling to obtain $2C$ number of diverse
      target candidate samples as $\mathcal{X}_{div}^{ut}$.
    • Compute the uncertainty for each diverse candidate,
      $d_T(x_{i,div}^{ut})$, according to (5).
    • Sort $d_T(x_{i,div}^{ut})$ to select the most effective $C$ target samples
      as $D_{lt}$.
    • Move the newly labeled target samples to $\mathcal{D}_{lt}$ from $\mathcal{D}_{ut}$.
Until size($\mathcal{D}_{lt}$) $= \mathcal{B}$
Output: Trained $\mathcal{F}$ and $\mathcal{W}$.
```

where $\mathcal{L}_{CBF}$ is the CB-focal loss over the labeled data. The $\mathcal{L}_{CBF}$ in training objective starts with source data $\mathcal{D}_s$. As the data from unlabeled target $\mathcal{D}_{ut}$ is labeled by the oracle, it is moved to labeled target data $\mathcal{D}_{lt}$ and used by $\mathcal{L}_{CBF}$. Thus, as labeled samples from the target domain are acquired, they are seamlessly infused into the domain adaptation process, without restarting the training process. Unlike other SSDA approaches which have access to labeled targets from the beginning of the adaptation process, our approach gets access to labeled targets at different stages in the domain adaptation process. Our model not only weighs the harder samples in labeled data from the start, but also the more uncertain/hard unlabeled target eventually as the model adapts. The feature extractor and classifier thus learnt is used to classify the remaining unlabeled samples.

5. Experimental Evaluation

Here, we provide the experimental details to evaluate the efficacy of the proposed approach for SSDA application. We start with the datasets used and implementation details. **Datasets Used:** In this work, we use three benchmark DA datasets. **DomainNet [22]** is a recently released large scale DA dataset with 6 different domains and a total of 345 classes with over 0.6 million images. Due to the prevalence of noise, experiments are conducted on the partial dataset consisting of 4 domains and 16 classes. Following the recent approaches, we conduct experiments on all the 7 domain adaptation scenarios using this large scale dataset. We also perform experiments on the other benchmark datasets, namely, **Office-31** [25] and **Office-Home** [38]. Office-31 is a relatively smaller dataset consisting of 3 domains (Ama-

Method	R to C		R to P		P to C		C to S		S to P		R to S		P to R		Mean	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
DANN [8]	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
ADR [27]	57.1	60.7	61.3	61.9	57.0	60.7	51.0	54.4	56.0	59.9	49.0	51.1	72.0	74.2	57.6	60.4
CDAN [18]	65.0	69.0	64.9	67.3	63.7	68.4	53.1	57.8	63.4	65.3	54.5	59.0	73.2	78.5	62.5	66.5
ENT [10]	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
BiAT [24]	73.0	74.9	68.0	68.8	71.6	74.6	57.9	61.5	63.9	67.5	58.5	62.1	77.0	78.6	67.1	69.7
MME [26]	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
APE [13]	70.4	76.6	70.8	72.1	**72.9**	**76.7**	56.7	63.1	64.5	66.1	63.0	67.8	76.6	79.4	67.6	71.7
Proposed STar	**74.1**	**77.1**	**71.3**	**73.2**	71.0	75.8	**63.5**	**67.8**	**66.1**	**69.2**	**64.1**	**67.9**	**80.0**	**81.2**	**70.0**	**73.2**

Table 1: Performance on the DomainNet dataset for 1-shot and 3-shot settings using ResNet34 backbone. We observe that the proposed STar not only eliminates the need to manually find target samples to be labeled, but also results in significantly better performance compared to all the state-of-the-art SSDA approaches.

zon, Webcam and DSLR) with 31 classes. Office-Home is a relatively larger and more challenging dataset and consists of 4 domains, namely Real, Clipart, Painting and Art with a total of 65 classes. We evaluate STaR on all the 12 different adaptation scenarios on Office-Home dataset.

Evaluation: For SSDA, the classification accuracy is measured only on the unlabeled target samples. For our approach, the test data consists of samples in $\mathcal{D}_{ut}$, i.e. the samples which are not selected for labeling by the proposed STar algorithm. Since the labeled target samples (and thus the remaining unlabeled ones) may be slightly different for the other SSDA approaches and the proposed STar framework, the exact testing data is slightly different. But the total number of labeled samples and thus the total number of testing data used in all the approaches is exactly the same. To evaluate the effectiveness of the second and third contributions, we also evaluate on the standard SSDA settings, where the labeled targets are also the same.

Implementation Details: For comparison with state-of-the-art approaches, we perform experiments with both Alexnet and Resnet34 backbones. We use SGD optimizer, with starting learning rate of 0.01, momentum of 0.9 and weight decay of 0.0005. We use batch size of 24 and 32 for Resnet34 and Alexnet backbones respectively on labeled samples, and twice the batch size for unlabeled data. We implemented the proposed algorithm in PyTorch [21] using a Geforce GTx 1080 card. The hyper-parameters $\{\alpha, \lambda, \gamma_1, \gamma_2\}$ are experimentally set to $\{0.8, 0.1, 0.5, 2\}$ respectively. We set δ_1 as $\{0.99, 0.9, 0.999\}$ for DomainNet, Office-31 and Office-Home respectively. $\delta_1 = 1$ and $\tau = 10000$ iterations for all experiments.

For DomainNet, the iteration at which target samples are selected is chosen to be multiples of 2500. For 1-shot, the selection of targets happen at 2500^{th} iteration, for 3-shot at $\{2500^{th}, 5000^{th}, 7500^{th}\}$ iterations. The target selection iteration is based on the convergence of the base approach (i.e. the unsupervised version) for the dataset. We performed experiments by varying this update iteration (500 on both sides) and did not find any considerable change in the results. For Office-31 and Office-Home, the update iteration is considered to be multiples of 200 and 2000 respectively.

5.1. Evaluation on DomainNet dataset

First, we evaluate the proposed STar technique on the large scale DomainNet and the results with ResNet34 backbone for both 1-shot and 3-shot settings are reported in Table 1. The results of all the other approaches are directly taken from [26, 24]. We observe that the proposed STar approach performs considerably better than all the state-of-the-art SSDA approaches. Irrespective of the adaptation scenario, we see that utilizing actively queried target samples, along with semantic initialization and weighting of unlabeled hard targets immensely helps in the adaptation.

5.2. Evaluation on Office-Home and Office-31

The detailed results on Office-Home for 1 & 3 shot scenarios are reported in Tables 2 for AlexNet backbone. The results for Office-31 using AlexNet backbone is reported in Table 3. We observe that for both the datasets, the proposed STar performs significantly better as compared to all the state-of-the-art SSDA approaches for all the settings, thus justifying the effectiveness of the proposed framework.

6. Analysis of the proposed STar framework

Here, we perform ablation study and further analysis.

Importance of each of the contributions: Here, we analyze the importance of the different components in the final performance improvement. We choose 2 domains, namely R to S and C to S from DomainNet dataset for this analysis, with ResNet34 backbone for both 1-shot and 3-shot settings. Table 4 (first three rows) reports the results with all the components for STar, i.e. active sampling of target samples (AS), semantic initialization (SI) and weighting Hard Target Samples (WHTS), and removing one component at a time. We observe that all the components contribute to the good performance of the entire framework.

In the last two experiments we do not use our active learning based target sample selection procedure and in-

Method	R to C	R to P	R to A	P to R	P to C	P to A	A to P	A to C	A to R	C to R	C to A	C to P	Mean
						One-shot							
S+T [26]	37.5	63.1	44.8	54.3	31.7	31.5	48.8	31.1	53.3	48.5	33.9	50.8	44.1
DANN [8]	42.5	64.2	45.1	56.4	36.6	32.7	43.5	34.4	51.9	51.0	33.8	49.4	45.1
ADR [27]	37.8	63.5	45.4	53.5	32.5	32.2	49.5	31.8	53.4	49.7	34.2	50.4	44.5
CDAN [18]	36.1	62.3	42.2	52.7	28.0	27.8	48.7	28.0	51.3	41.0	26.8	49.9	41.2
ENT [10]	26.8	65.8	45.8	56.3	23.5	21.9	47.4	22.1	53.4	30.8	18.1	53.6	38.8
BiAT [24]	-	-	-	-	-	-	-	-	-	-	-	-	49.6
MME [26]	42.0	69.6	48.3	58.7	37.8	34.9	52.5	36.4	57.0	54.1	39.5	59.1	49.2
Proposed STar	**46.5**	**71.6**	**53.6**	**62.0**	**42.3**	**38.0**	**57.8**	**37.8**	**59.8**	**57.6**	**42.5**	**61.0**	**52.54**
						Three-shot							
S+T [26]	44.6	66.7	47.7	57.8	44.4	36.1	57.6	38.8	57.0	54.3	37.5	57.9	50.0
DANN [8]	47.2	66.7	46.6	58.1	44.4	36.1	57.2	39.8	56.6	54.3	38.6	57.9	50.3
ADR [27]	45.0	66.2	46.9	57.3	38.9	36.3	57.5	40.0	57.8	53.4	37.3	57.7	49.5
CDAN [18]	41.8	69.9	43.2	53.6	35.8	32.0	56.3	34.5	53.5	49.3	27.9	56.2	46.2
ENT [10]	44.9	70.4	47.1	60.3	41.2	34.6	60.7	37.8	60.5	58.0	31.8	63.4	50.9
BiAT [24]	-	-	-	-	-	-	-	-	-	-	-	-	56.4
APE [13]	**51.9**	**74.6**	51.2	61.6	47.9	42.1	65.5	44.5	60.9	58.1	44.3	64.8	55.6
MME [26]	51.2	73.0	50.3	61.6	47.2	40.7	63.9	43.8	61.4	59.9	44.7	64.7	55.2
Proposed STar	**51.9**	74.1	**54.0**	**65.1**	**48.5**	**43.0**	**66.0**	**45.8**	**63.0**	**61.5**	**46.1**	**65.3**	**57.03**

Table 2: 1 and 3 shot results on Office-Home dataset using Alexnet backbone.

Method	W to A		D to A	
	1-shot	3-shot	1-shot	3-shot
S+T [26]	50.4	61.2	50.0	62.4
DANN [8]	57.0	64.4	54.5	65.2
ADR [27]	50.2	61.2	50.9	61.4
CDAN [18]	50.4	60.3	48.5	61.4
ENT [10]	50.7	64.0	50.0	66.2
BiAT [24]	57.9	68.2	54.6	68.5
MME [26]	57.2	67.3	55.8	67.8
Proposed STar	**59.8**	**69.1**	**56.8**	**69.0**

Table 3: Evaluation on Office-31 dataset for 1-shot and 3-shot settings using Alexnet backbone.

Method	Components			R to S		C to S	
	AS	SI	WHTS	1 shot	3 shot	1 shot	3 shot
Actively Labeled Targets	✓	✓	✓	64.1	67.9	63.5	67.8
	✓	✓		62.5	66.8	62.5	65.2
	✓			62.1	66.0	62.3	64.3
Manually Labeled Targets		✓	✓	62.7	64.2	59.4	64.3
		✓		62.0	63.1	58.4	63.9

Table 4: Ablation Study using Resnet34 backbone on DomainNet dataset for R to S and C to S settings.

stead use the labeled target examples used by the other SSDA approaches [26, 24, 13]. We observe that except for R to S, for the other settings, even with the same labeled examples, using only two of the proposed contributions, we achieve similar or better performance as compared to the state-of-the-art. This validates the usefulness of semantic-initialization, weighting hard target and selection of informative target samples for the SSDA task.

Per-class performance analysis: We observe that there is significant amount of data imbalance (both in source and target domain) in the datasets, specially DomainNet. Due to this imbalance, classes with lesser number of target samples may have difficulty in adapting their class-prototypes. Here, we perform per-class accuracy analysis on DomainNet for R to S, 3-shot scenario. We compare the per-class accuracies on the basis of number of target samples per class. The average of class-wise accuracies for 30-classes with lowest number of target samples is calculated and compared with another SSDA method MME [26]. For the lowest 30-classes, STar performs extremely well with an average accuracy of 56.2% as compared to 40.8% for MME. For the 30 classes with highest number of target samples, the average of per-class accuracies is 69.2% for STar and 61.9% for MME. This gives a clearer picture of the adaptation process, since due to presence of less number of target samples in several classes, even a poorer accuracy for those classes does not reflect in the overall accuracy. This also shows the effectiveness of STaR to deal with the inherent data imbalance.

7. Conclusion

In this work, we proposed a novel SSDA framework, STar. Observing that few labeled target samples are extremely crucial for generalization, instead of the traditional approach of selecting equal number of target samples per class, we propose an active learning based strategy, for selecting the most informative target samples to label. We also propose semantic initialization and weighing of hard target samples for facilitating the adaptation process. Extensive experiments on several datasets justify the effectiveness of the proposed framework.

References

[1] William H Beluch, Tim Genewein, Andreas Nurnberger, and Jan M Kohler. The power of ensembles for active learning in image classification. In *CVPR*, 2018.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. In *Machine learning, 79(1-2)*, page 151–175, 2010.

[3] S. Bickel, M. Bruckner, and T. Scheffer. Discriminative learning under covariate shift. In *Journal of Machine Learning Research*, 2009.

[4] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye. Joint transfer and batch-mode active learning. In *ICML*, 2013.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *ICCV*, 2019.

[6] Titir Dutta, Anurag Singh, and Soma Biswas. Adaptive margin diversity regularizer for handling data imbalance in zero-shot sbir. In *European Conference on Computer Vision*, pages 349–364. Springer, 2020.

[7] Titir Dutta, Anurag Singh, and Soma Biswas. Styleguide: Zero-shot sketch-based image retrieval using style-guided image generation. *IEEE Transactions on Multimedia*, 2020.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *The Journal of Machine Learning Research*, 2016.

[9] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[10] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005.

[11] Wan Ji, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *ACMM*, 2014.

[12] Tapas Kanugo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. In *IEEE transactions on pattern analysis and machine intelligence*, 2002.

[13] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[15] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*, 2019.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[18] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013.

[20] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *ICML*, 2004.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[22] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[24] Jiang Pin, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, 2020.

[25] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[26] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

[27] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, , and Kate Saenko. Adversarial dropout regularization. In *ICLR*, 2018.

[28] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[29] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and Fukumizu. K. equivalence of distance-based and rkhs-based statistics in hypothesis testing. In *The Annals of Statistics*, 2013.

[30] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

[31] Burr Settles. Active learning literature survey. 2009.

[32] Tasfia Shermin, Guojun Lu, Shyh Wei Teng, Manzur Murshed, and Ferdous Sohel. Adversarial network with multiple classifiers for open set domain adaptation. *IEEE Transactions on Multimedia*, 2020.

[33] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

[34] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *AAAI*, 2019.

[35] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, 2019.

[36] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, and Subhransu Maji. Active adversarial domain adaptation. In *WACV*, 2019.

[37] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[38] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, , and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[39] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, 2018.

[40] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *IJCNN*, 2014.

[41] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[42] Fan Yang, Yang Wu, Zheng Wang, Xiang Li, Sakriani Sakti, and Satoshi Nakamura. Instance-level heterogeneous domain adaptation for limited-labeled sketch-to-photo retrieval. *IEEE Transactions on Multimedia*, 2020.

[43] Fan Yang, Ke Yan, Shijian Lu, Huizhu Jia, Don Xie, Zongqiao Yu, Xiaowei Guo, Feiyue Huang, and Wen Gao. Part-aware progressive unsupervised domain adaptation for person re-identification. *IEEE Transactions on Multimedia*, 2020.

[44] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qingming Huang. State-relabeling adversarial active learning. In *CVPR*, 2020.

[45] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[46] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017.