Active Adversarial Domain Adaptation

Jong-Chyi Su[*1] Yi-Hsuan Tsai[2] Kihyuk Sohn[†2] Buyu Liu[2]
Subhransu Maji[1] Manmohan Chandraker[2,3]

[1]UMass Amherst [2]NEC Laboratories America [3]UC San Diego

Abstract

We propose an active learning approach for transferring representations across domains. Our approach, active adversarial domain adaptation (AADA), explores a duality between two related problems: adversarial domain alignment *and* importance sampling *for adapting models across domains. The former uses a domain discriminative model to align domains, while the latter utilizes the model to weigh samples to account for distribution shifts. Specifically, our importance weight promotes unlabeled samples with large uncertainty in classification and diversity compared to labeled examples, thus serving as a sample selection scheme for active learning. We show that these two views can be unified in one framework for domain adaptation and transfer learning when the source domain has many labeled examples while the target domain does not. AADA provides significant improvements over fine-tuning based approaches and other sampling methods when the two domains are closely related. Results on challenging domain adaptation tasks such as object detection demonstrate that the advantage over baseline approaches is retained even after hundreds of examples being actively annotated.*

1. Introduction

The assumption that the training and test data are drawn from the same distribution may not be true in practical applications of machine learning and computer vision. Consequently, a predictor trained on the source domain $\mathcal{S}$ may perform poorly when evaluated on the target domain $\mathcal{T}$ different from the source. This *covariate shift* problem is common in many problems, *e.g*., the seasonal distribution of natural species may change in a camera trap dataset, or the image resolution can change from one dataset to another.

Many domain adaptation (DA) methods have been proposed to address this issue [10, 13, 36, 37, 63, 64, 65]. The covariate shift assumes that the marginal distribution $p(x)$ of the data changes from $\mathcal{S}$ to $\mathcal{T}$, while the conditional label distribution $p(y|x)$ remains the same. Domain adaptation methods operate by minimizing the differences of the marginal distributions of x in the source domain $p_{\mathcal{S}}(x)$ and target domain $p_{\mathcal{T}}(x)$ by projecting the data through an embedding $\Phi(x)$, *e.g*., a deep network, while at the same time being predictive of the distribution $p_{\mathcal{S}}(y|x)$ in the source domain. By matching the marginals, the covariate shift is reduced, thus improving the generalization of the model on the target domain compared to an "unadapted" model.

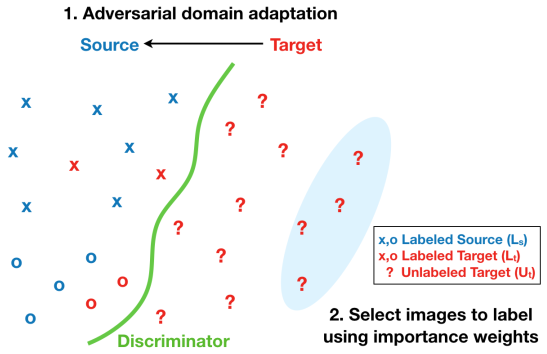

1. Adversarial domain adaptation
Source
Target
x,o Labeled Source (L_s)
x,o Labeled Target (L_t)
? Unlabeled Target (U_t)
Discriminator
2. Select images to label using importance weights

Figure 1: Source and target domain data are shown in blue and red. Circle and cross represent class labels, while question marks are unlabeled data. We employ adversarial training to align features across the source and target domain, and use discriminator predictions to compute the *importance weight* for sample selection of active learning.

While domain adaptation provides a good starting point, the performances of unsupervised DA methods often fall far behind their supervised counterparts [7, 61]. In such cases, some labeled data from the target domain may bring in performance benefits. However, obtaining ground-truth annotations can be laborious and naïvely collecting annotated data could be inefficient. In this work, we aim to answer the following questions: 1) how to select data to label from the target domain effectively, and 2) how to perform adaptation given these labeled data from the target domain.

[*] Partial work done while at NEC Labs.
[†] Currently at Google Cloud AI.

To this end, we propose *Active Adversarial Domain Adaptation* (AADA) that exploits the relation between domain adaptation and active learning to answer those questions. Addressing our second question, we propose to adopt domain adversarial learning [13] between the union of labeled data from source/target and unlabeled target data, when the amount of labeled target data is small. However, after several rounds of active selection to accumulate many labeled data from the target domain, performing adversarial adaptation becomes counter-productive and simple transfer learning approaches (*e.g.*, fine-tuning) serve the purpose.

Inspired by the importance weighted empirical risk minimization [58, 59], we address our first question by proposing a sample selection criterion composed of the two cues: the *diversity cue* and the *uncertainty cue*. The diversity cue is from the *importance* $w(x) = p_T(x)/p_S(x)$ where it can be estimated efficiently from the domain discriminator based on domain adversarial learning [15]. This allows one to sample unlabeled targets that are different from the labeled ones. The uncertainty cue is a lower bound to the empirical risk, which in our case is in the form of entropy of classification distribution. This promotes unlabeled data with low confidence for the next round of annotation. The overall framework of our AADA is illustrated in Figure 1.

In experiments, we first validate the effectiveness of our approach on digit classification from SVHN to MNIST in Section 4, showing significant improvements over other baselines on domain adaptation, transfer learning, and active learning. Second, we conduct experiments for object recognition on the Office [50] and VisDA [43] datasets with larger domain shifts in Section 5. Last, we extend our method to object detection, adapting from the KITTI dataset [14] to the Cityscapes dataset [9]. The proposed AADA outperforms the fine-tuning baseline by 6% when only 50 labeled images from the target domain are available.

Finally, we summarize our contributions as follows:

- An active learning framework by integrating domain adversarial learning and active learning for continuous semi-supervised domain adaptation.
- Improved classification performance with domain adversarial learning, while the discriminator prediction yields better importance weight for sampling.
- A connection between our sampling method and importance weight with domain adversarial training.
- Reduced labeling cost on target domain on object classification and detection tasks.

2. Related Work

2.1. Domain Adaptation

Domain adaptation (DA) aims to make the model invariant to the data from the source and target domain. For example, [10] uses unlabeled data to measure the inconsistency between source and target domain classifiers. Deep domain adaptation has been successful in recent years. The key idea is to measure the domain discrepancy at a certain layer of deep networks using domain discriminator [4, 13] or maximum mean discrepancy (MMD) kernel [36, 37, 63, 65] and train CNNs to reduce the discrepancy. Approaches that combine techniques from semi-supervised learning, such as entropy minimization [16, 31], are proposed to enhance classification performance [37, 71]. It has also been applied to more complicated vision tasks such as object detection [7, 23, 21] and semantic segmentation [18, 19, 61, 62], where the annotation cost is more expensive and how to select images to label become more crucial.

Different from the above-mentioned unsupervised DA, we explore the case where the budget is available to annotate a few labeled examples in the target domain. Comparing to the method of [40] which discusses how to train the model given few labeled targets with uniform distribution, we focus on *how to select target samples to label without knowing any prior distribution* of the target labels.

2.2. Active Learning

Active learning aims to maximize the performance with a limited annotation budget [8, 55]. Thus, the challenge is to quantify the *informativeness* of unlabeled data [27] so that they are maximally useful when annotated. Many sampling strategies based on uncertainty [32, 53], diversity [11, 20], representativeness [68], reducing expected error [48, 67] and maximizing expected label changes [12, 25, 66] are studied and applied to vision tasks such as classification [45, 24], object detection [26], image segmentation [38, 60, 66], and human pose estimation [35]. Among these, uncertainty sampling is simple and computationally efficient, making it a popular strategy in real-world applications.

Learning-based active learning methods [22, 29] are proposed recently by formulating a regression problem for the query procedure and learning strategies based on previous outcomes. Deep active learning methods [54, 57] are studied for image classification and named-entity recognition. [39, 72] propose to use generative models to synthesize data for training, but the performance is largely dependent on the quality of synthetic data, limiting their generality.

2.3. Active Learning for Domain Adaptation

Different from the aforementioned methods, we aim to unify active learning and domain adaptation. Chattopadhyay *et al.* [6] train the domain adaptation model with importance weights [2] and select samples by solving linear programming for minimizing the MMD distances between features. However, it is not clear how to incorporate this strategy with advanced techniques such as deep models and domain adversarial training.

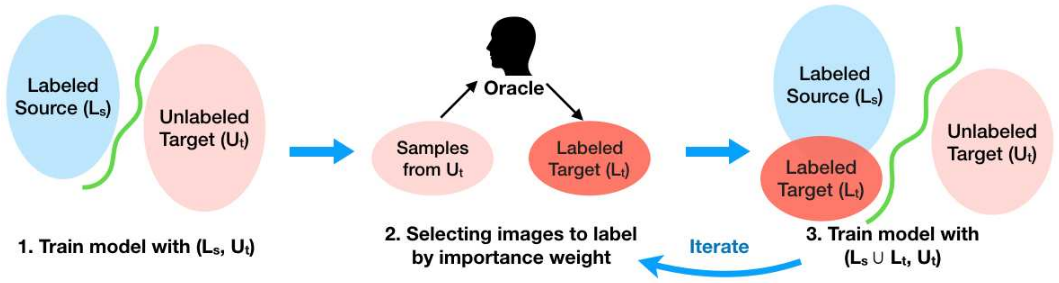


Figure 2: Our proposed algorithm AADA. We start from an unsupervised domain adaptation setting with labeled source L_s and unlabeled target U_t data and train the model with domain adversarial loss. In each following round, we first select samples using importance weight from the unlabeled target domain to obtain annotations. We then re-train the model with labeled data $L_s \cup L_t$ and unlabeled data U_t.

The most relevant work is ALDA [46, 51], which demonstrates its effectiveness in sentiment and landmine classification tasks. ALDA trains three models, a source classifier w_{src}, a domain adaptive classifier u_ϕ, and a domain separator w_{ds}. It first selects unlabeled target samples using u_ϕ, and decide whether to acquire the label from w_{src} (without cost) or the oracle (with cost) using w_{ds}. u_ϕ is then updated with the obtained labeled data.

In addition to using the deep model, the proposed AADA is different from ALDA in several ways. First, our discriminator not only helps sample selection but also trains the recognition model adversarially to reduce the domain gap. Moreover, we combine diversity in the form of discriminator prediction and uncertainty in the form of entropy. To the best of our knowledge, we are the first to jointly tackle DA and active learning using neural networks on vision tasks.

3. Proposed Algorithm

In this section, we introduce our active adversarial domain adaptation (AADA). We begin with the background of domain adversarial neural networks in Section 3.1, and then we motivate our sampling strategy by importance in Section 3.2. The algorithm and its theoretical background under the semi-supervised domain adaptation setting are provided in Section 3.3.

3.1. Domain Adaptation

In this section, we introduce the learning objective of our domain adaptation model. For simplicity, we describe the model in the image classification task. We denote X as the input space and $Y = \{1, ..., L\}$ as the label space. The source data and (unlabeled) target data are drawn from the distribution $p_{\mathcal{S}}(x)$ and distribution $p_{\mathcal{T}}(x)$ respectively. We adopt the domain adversarial neural network (DANN) [13], which is composed of three components: *feature extractor* G_f for the input x, *class predictor* G_y that predicts the class label $G_y(G_f(x)) \rightarrow \{1, ..., L\}$, and *discriminator* G_d that classifies the domain label $G_d(G_f(x)) \rightarrow \{0, 1\}$. We use 1 for the source domain and 0 for the target domain. The objective function of the discriminator G_d is defined as:

$$\begin{aligned}\mathcal{L}_d = \ &\mathbb{E}_{x\sim p_{\mathcal{S}}(x)}\big[\log G_d(G_f(x))\big] \\ &+ \mathbb{E}_{x\sim p_{\mathcal{T}}(x)}\big[\log(1 - G_d(G_f(x)))\big],\end{aligned} \tag{1}$$

where G_f, G_y, G_d are parameterized by $\theta_f, \theta_y, \theta_d$, respectively. To perform domain alignment, features generated from G_f should be able to fool the discriminator G_d, and hence we adopt an adversarial loss to form a min-max game:

$$\min_{\theta_f,\theta_y} \max_{\theta_d} \mathcal{L}_c(G_y(G_f(x)), y) + \lambda \mathcal{L}_d, \tag{2}$$

where $\mathcal{L}_c$ is the cross-entropy loss for classification, y is the class label, and λ is the weight between two losses.

3.2. Sample Selection

Given an unsupervised domain adaptation setting where labeled data is only available from the source domain, the goal of our sample selection is to find the most informative data from the unlabeled target domain. We motivate the sample selection criteria from the idea of importance weighted empirical risk minimization (IWERM) [58], whose learning objective is defined as follows:

$$\min_{\theta_f,\theta_y} \mathbb{E}_{(x,y)\sim p_{\mathcal{S}}(x,y)}\left[\frac{p_{\mathcal{T}}(x)}{p_{\mathcal{S}}(x)}\mathcal{L}_c\big(G_y(G_f(x)), y\big)\right], \tag{3}$$

where $w(x) = \frac{p_{\mathcal{T}}(x)}{p_{\mathcal{S}}(x)}$ is an importance of each labeled data in the source domain. The formulation indicates which data is more important during optimization. First, the data with higher empirical risk $\mathcal{L}_c\big(G_y(G_f(x)), y\big)$, and second, the one with higher importance, *i.e.*, larger density in the target distribution $p_{\mathcal{T}}(x)$ but lower in the source $p_{\mathcal{S}}(x)$.

Unfortunately, applying this intuition to come up with a sample selection strategy is non-trivial. This is because the target data is mostly unlabeled and the empirical risk cannot be computed before annotation. Another problem is that the importance estimation of high-dimensional data is difficult [59]. We take advantage of domain discriminator to

Algorithm 1 AADA

```
Input: labeled source $L_s$; unlabeled target $U_t$;
        labeled target $L_t = \emptyset$; budget per round $b$
Model: $\mathcal{M}$={$G_f$, $G_y$, $G_d$}; feature extractor $G_f$;
        class predictor $G_y$; discriminator $G_d$
Train $\mathcal{M}$ with $(L_s, U_t)$
for round $\leftarrow$ 1 to MaxRound do
    Compute $s(x)\ \forall x \in U_t$ via (5)
    Select a set of $b$ images $z$ from $U_t$ according to $s(z)$
    Get labels $y_z$ from oracle
    $L_t \leftarrow L_t \cup (z, y_z)$
    $U_t \leftarrow U_t \setminus (z, y_z)$
    Train $\mathcal{M}$ with $(L_s \cup L_t, U_t)$
```

resolve the second issue. Note that, with adversarial training, the optimal discriminator [15] is obtained at

$$G_d^*(\hat{x}) = \frac{p_{\mathcal{S}}(x)}{p_{\mathcal{S}}(x) + p_{\mathcal{T}}(x)} \Rightarrow w(x) = \frac{1 - G_d^*(\hat{x})}{G_d^*(\hat{x})}, \tag{4}$$

where $\hat{x} = G_f(x)$. Next, assuming cross-entropy as an empirical risk, we resolve the first issue by measuring the entropy of unlabeled data, which is a lower bound to the cross-entropy.[1] Finally, our sample selection criterion $s(x)$ for unlabeled target data is written as follows:

$$s(x) = \frac{1 - G_d^*(G_f(x))}{G_d^*(G_f(x))} \mathcal{H}(G_y(G_f(x))). \tag{5}$$

Two components in the measure are interpreted as follows: 1) *diversity* cue $(1-G_d^*(G_f(x)))/G_d^*(G_f(x))$, and 2) *uncertainty* cue $\mathcal{H}(G_y(G_f(x)))$. The diversity cue allows us to select unlabeled target data which is less similar to the labeled ones in the source domain, while the uncertainty cue suggests data that the model cannot predict confidently.

3.3. Active Adversarial Domain Adaptation

Based on the two objectives of domain adaptation and sample selection, we explain the role of these two components in their collaboration for active learning for domain adaptation purposes.

Collaborative Roles. For domain adaptation, the goal is to learn domain-invariant features via (2) that better serves as a starting point for the next sample selection step. During the adversarial learning process, a discriminator is learned to separate source and target data, and thus we can utilize its output prediction as an indication for selection via the importance weight in (5). By iteratively performing adversarial learning and active learning, the proposed method gradually selects informative samples for annotations guided by the domain discriminator, and then these selected samples are used for supervised training to minimize the domain gap, in a collaborative manner.

One may still obtain a discriminator without adversarial learning and it can be easily learned to separate samples across two different domains. However, learning a discriminator in this way can be problematic for active learning. First, this discriminator may give identically high scores to most target samples. Thus it lacks the capability of selecting informative ones. Moreover, the learned classifier and this discriminator may focus on different properties if they are not learned jointly. If this is the case, the informative samples that current discriminator selects are not necessarily beneficial for classifier update. We provide more evidence for the necessity of adversarial training in Section 4.3.

Active Learning Process. Our overall active learning framework is illustrated in Figure 2. We start our AADA algorithm by learning a DANN model in an unsupervised domain adaptation setting as described in Section 3.1, and then use the learned discriminator to perform the initial round of sample selection from all unlabeled target samples based on (5). Once obtaining the selected samples, we acquire their ground-truth labels.

For the following rounds, we have a small set of labeled target data $L_t \sim p_{\mathcal{T}}(x, y)$, a set of labeled source data $L_s \sim p_{\mathcal{S}}(x, y)$, and the remaining unlabeled target data $U_t \sim p_{\mathcal{T}}(x)$. Thus, the learning setting is different from the initial stage as we now have labeled domains L_s and L_t. To accommodate labeled data from both domains, we revisit an analysis of domain adaptation [1, 3] whose generalization bound is given as:

$$\epsilon_{\mathcal{T}}(\hat{h}) \leq \epsilon_{\mathcal{T}}(h_{\mathcal{T}}^*) + \gamma_\alpha + d_{\mathcal{H}\Delta\mathcal{H}}(L_s \cup L_t, U_t) \\ + 4\sqrt{\left(\frac{\alpha_s^2}{\beta_s} + \frac{\alpha_t^2}{\beta_t}\right)\left(\frac{d\log(2m) - \log(\delta)}{2m}\right)}, \tag{6}$$

with $\gamma_\alpha = \epsilon_{\mathcal{T}}(h) + \alpha_s\epsilon_{\mathcal{S}}(h) + \alpha_t\epsilon_{\mathcal{T}}(h)$, m is the number of labeled examples, d is VC-dimension of hypothesis class and h is the hypothesis (*i.e.*, classifier). α=(α_s, α_t) is a weight vector between the errors of labeled source and labeled target, while β=(β_s, β_t) is a proportion of labeled examples for source and target domains. Assuming zero error on the labeled examples (*i.e.*, $\epsilon_{\mathcal{S}}(h) = \epsilon_{\mathcal{T}}(h) = 0$), the bound is the tightest when $\alpha_s = \beta_s$ and $\alpha_t = \beta_t$.

This leads us training a new model that adapts from all labeled data $L_s \cup L_t$ to unlabeled data U_t with uniform sampling of individual examples from labeled set to ensure the tightest bound. Thus, we use uniform sampling of labeled source and target examples for sampling batches during training unless otherwise stated. Then, we select candidates from the remaining unlabeled target set U_t based on the new discriminator G_d and new classifier G_y following the same importance sampling strategy for the next round of training. The overall algorithm is shown in Algorithm 1.

[1] $H(p, q) = D_{KL}(p||q) + H(p) \geq H(p)$.

4. Experiments on Digit Classification

As discussed above, our proposed method aims to address two questions: 1) how to select images to label from U_t to yield the most performance gain? and 2) how to train a classifier given $\{L_s, L_t, U_t\}$? Our experiments then consists of our explorations for both components. In this section, we first perform detailed experiments in a mix-and-match way on the digit classification task from SVHN [41] to MNIST [30]. Specifically, we explore the following **training schemes**:

1) Adversarial Training: we train the classifier via (2) using $(L_s \cup L_t, U_t)$.

2) Joint Training: we train the classifier in a supervised way using $L_s \cup L_t$. Note that we still train a discriminator for sample selection but without adversarial training.

3) Fine-tuning: we train a classifier using L_s and then fine-tune it on L_t, both in a supervised way. Discriminator is trained in a similar manner to Joint Training.

4) Target Only: we train our classifier with L_t only.

The **sampling strategies** we explored are:

1) Importance Weight: we select samples based on the proposed importance weight $s(x)$ (5).

2) K-means Clustering: we perform k-means clustering on image features $G_f(x), \forall x \in U_t$, where the number of clusters is set to b in each round. For each cluster, we select one sample which is the closest to its center.

3) K-center (Core-set) [54]: we use greedy k-center clustering to select b images z from U_t such that the largest distance between unlabeled data $U_t \setminus z$ and labeled data $L_t \cup z$ is minimized. We use L2 distance between image features $G_f(x)$ for the measurement.

4) Diversity [11]: for each unlabeled sample in U_t, we compute its distance to all samples in L_t and obtain the average distance. Then we rank unlabeled samples w.r.t. its average distance in descending order and select the top b samples. L2 distance is applied on features $G_f(x)$.

5) Best-versus-Second Best (BvSB) [24]: we use the difference between the highest and the second highest class prediction as the uncertainty measure., *i.e.*, $\max_i G_{y_i}(\hat{x}) - G_{y_j}(\hat{x})$, where class j has the second highest prediction.

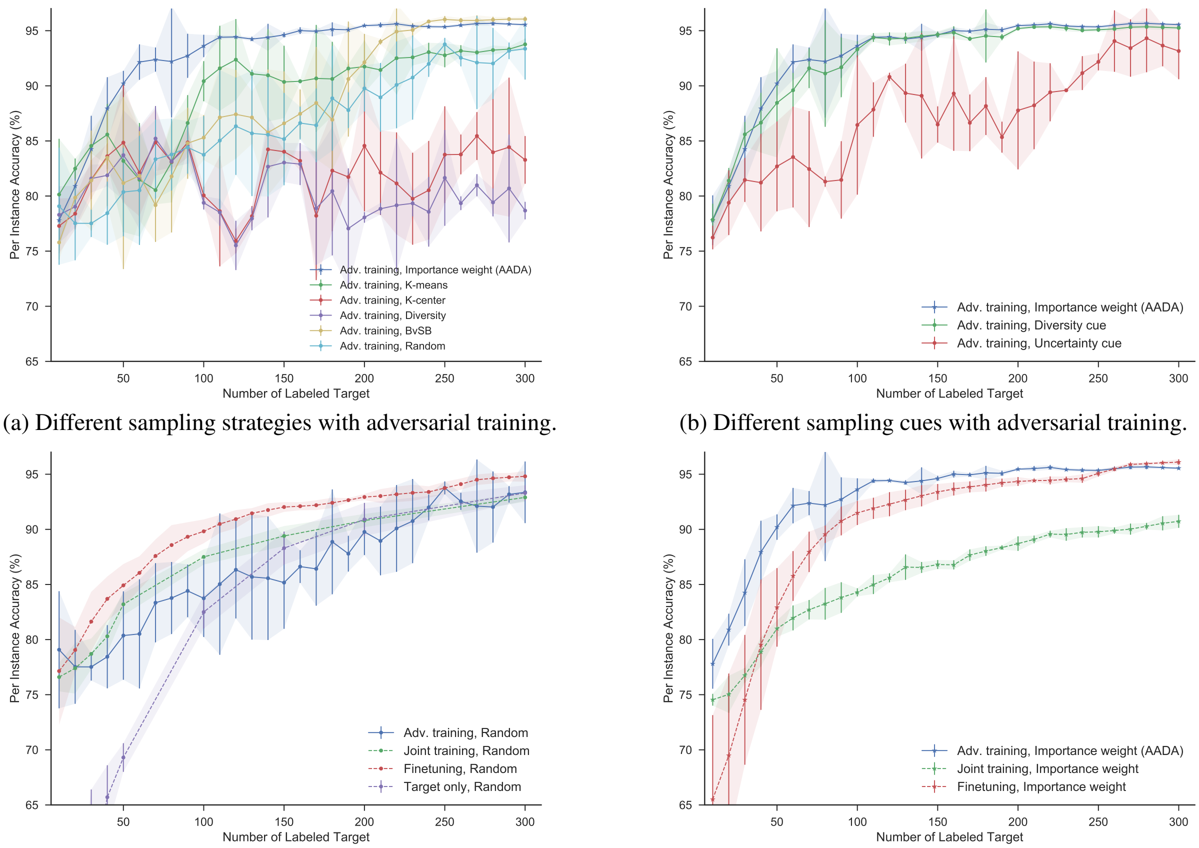
6) Random Selection: we select samples uniformly at random from all the unlabeled target data U_t.

Our AADA uses importance weight for sample selection, and adversarial training as the training scheme. We note that other unsupervised DA methods can be orthogonal to our approach, *e.g.*, one can use improved DANN such as CyCADA [18] for initialization but still use our criteria for selecting samples to label. Here we focus on sample selection and only use the vanilla adversarial training. We also note that different sampling methods do not compete with AADA as they can be combined with our method. For example, BvSB can be used as an alternative uncertainty measurement as opposed to entropy in (5).

Experimental Setting. Commonly in the active learning literature [38, 60], we simulate oracle annotations by using the ground-truth in all our experiments. We consider an adaptation task from SVHN to MNIST, where the former and latter are initially considered as labeled source L_s and unlabeled target U_t respectively. SVHN contains 73,257 RGB images and MNIST consists of 60,000 grayscale images, both from the digit classes of 0 to 9. Not only differ in color statistics, the images from two datasets also experience different local deformations, making the adaptation task challenging. For this task, we use the variant of LeNet architecture [18] and add an entropy minimization loss $\mathcal{L}_{ent} = \mathcal{H}(G_y(G_f(x))$ for regularization [37] during training. For each round, we train the model for 60 epochs using Adam [28] optimizer with learning rate $\{2 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}\}$ for 20 epochs each. The batch size is 128 and $\lambda = 0.1$. We set budget to 10 in each round and perform 30 rounds, eventually selecting 300 images in total from the target domain. We carry our experiments with five different random seeds and report the averaged accuracy after each round. We use PyTorch [42] for our implementation.

4.1. Comparison of Sampling Methods

We start from comparing different sampling method combined with adversarial training. As shown in Figure 3a, importance weight often outperforms its active sampling counterparts. It can achieve 95% accuracy with 160 samples after 16 rounds while the random selection baseline requires two times more annotations to have similar performance. Moreover, our proposed method consistently improves performance when more samples are selected and annotated, whereas other baselines generate unstable performances. One reason for such observation is that the class distribution of the selected samples in each round is not uniform. If the selected targets are heavily biased towards few classes, the "mode collapse" issue due to adversarial training gives high test accuracy on those classes but low accuracy on others, causing the overall lower accuracy. However, sampling with importance weight makes the result more stable after each round. As a reference, AADA performs similarly as random selection (97.5% accuracy) with 1000 labeled targets. The performance saturates at around 99.0% accuracy with 5000 labeled targets and achieves 99.5% accuracy with all 73,257 labeled targets.



(a) Different sampling strategies with adversarial training.

(b) Different sampling cues with adversarial training.

(c) Different training schemes with random sampling.

(d) Different training schemes with importance weight.

Figure 3: Ablation studies on digit classification (SVHN → MNIST). Each data point is the mean accuracy over five runs, and the error bar shows the standard deviation. We show that: (a) sampling using importance weight performs the best when using adversarial training, (b) combining diversity and uncertainty cues performs better for selecting samples, (c) fine-tuning is the best training scheme when random sampling is used, (d) when using importance weight for sampling, adversarial training is the best when there are less than 250 labeled target. Overall, our AADA which uses adversarial training and importance weigh provides the best performance when few labeled targets are available.

4.2. Comparison of Different Cues

We perform an ablation study of the two components in the proposed importance weight (5). The *diversity* cue, *i.e.*, $\frac{1-G_d^*(G_f(x))}{G_d^*(G_f(x))}$, uses the predictions from the discriminator G_d, while the *uncertainty* cue $\mathcal{H}(G_y(G_f(x)))$ uses the predictions from the classifier G_y. As shown in Figure 3b, using diversity cue outperforms that of uncertainty cue, while combining these two yields the best performance. However, the benefits of using different cues may depend on the characteristics of each dataset and will be discussed later.

4.3. Comparison of Training Schemes

We compare different training schemes and show the effectiveness of combining adversarial training with importance weight. First, we provide a study of four training schemes in Figure 3c, all using random sampling. In this case, we find that adversarial training suffers from mode collapse problem and fine-tuning is the best option. Fine-tuning is also the most effective and widely-used method of transfer learning as discovered in the deep learning literature [56, 69].

However, once the imbalance sampling problem can be effectively addressed, *e.g*. using the proposed importance weight, we can benefit from adversarial training. Figure 3d demonstrates the effectiveness of combining adversarial training with importance weight. We can see that it outperforms all the settings in Figure 3c. Moreover, our AADA method demonstrates its effectiveness especially when very few labeled targets L_t are available; on the other hand, when more and more labeled targets are available, fine-tuning seems to be a better option as the benefit of leveraging information from source domain has decreased (as explained in Section 1). In our experiment, using fine-tuning performs better than using adversarial training when there are more than 250 labeled target selected using importance weight.

Comparison with ALDA [51]. For the baseline of using joint training and importance weight, we train the classifier G_y and the feature extractor G_f with $L_s \cup L_t$, and train the discriminator G_d for separating labeled and unlabeled data. The two objectives are trained jointly but not adversarially. This can be seen as an extension of ALDA [51] using deep learning framework, despite some differences such as 1) the use of joint training instead of updating a perceptron, and 2) selecting samples using our proposed importance weight instead of using the margins to the linear classifier.

Interestingly, this baseline (as shown in Figure 3d) is worse than the one using joint training and random sampling (as shown in Figure 3c). This is mainly due to the lack of diversity. Specifically, without the help of adversarial loss, the importance weight can be very confident thus lacks the ability to provide sufficient diverse samples. This problem also remains for the original ALDA [51] method. Again, as shown in Figure 3d, our AADA outperforms this baseline by 7.6% on average of the first 25 rounds, showing that adversarial training not only helps adapt the model but also collaborates with importance weight for sampling.

5. More Experimental Results

In this section, we conduct experiments on object recognition and object detection datasets. Here we focus on comparing different sampling methods and refer the readers to supplementary material for complete comparisons.

5.1. Object Recognition

We validate our idea on the Office domain adaptation dataset [50]. It consists of 31 classes and three domains: amazon (A), webcam (W), and dslr (D), each with $\{2817, 795, 498\}$ images. Specifically, we select dslr (D) as the source domain and amazon (A) as the target one. We further split the target domain using the first $2/3$ images as U_t and the rest as the test set to evaluate all methods. We utilize ResNet-18 [17] model (before the first `fc` layer) pretrained on ImageNet as the feature extractor G_f. On top of it, G_y has one layer while G_d has `fc-ReLU-fc` with 256-256-2 channels. We train our model with SGD for 30 epochs with a learning rate of 0.005. The batch size is 32 and $\lambda = 0.0001$. Budget per-round b is set to 50 and we perform 20 rounds in total. We start the first round with random selection for all the methods as a warm-up.

Figure 4 demonstrates different sampling baselines with adversarial training. Our AADA method performs competitively with BvSB and outperforms all other methods, suggesting that the uncertainty cue is more useful in this dataset. More specifically, AADA outperforms random selection by around 3% from round 10 to round 20, and our AADA can achieve 85% accuracy with 800 labeled targets while random selection requires 200 more to achieve similar performance. Note that BvSB is one of the variants of our method, which also deploys our adversarial training scheme and uncertainty measurement.

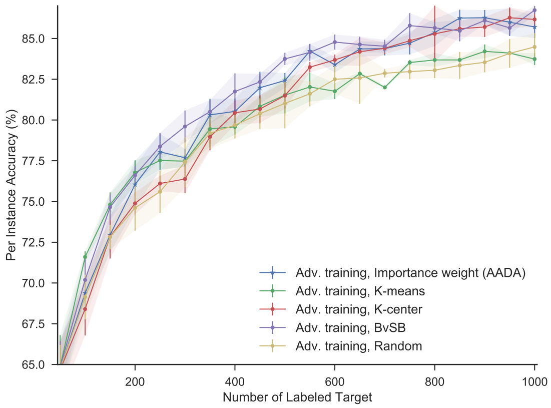


Figure 4: Object classification result (Office D → A). We compare different sampling methods with adversarial training. BvSB and AADA perform the best with 81.3% and 80.7% mean accuracy of 20 rounds separately.

5.2. Object Detection

Now we focus on object detection task adapting from KITTI [14] to Cityscapes [9]. We use the same setting as [7], which only considers the car object and resizes images to 500 for the shorter edge while keeping the aspect ratio. After discarding images without cars, we obtain 6,221 and 2,824 training images from KITTI and Cityscapes respectively, and we split 500 images from Cityscapes for testing. Mean average precision at 0.5 IoU (mAP@0.5) is our evaluation metric in this task [7, 23]. We adopt Faster-RCNN [47] with the ResNet-50 architecture combining with FPN [33] as the feature extractor, and perform image-level adaptation as proposed in [7]. We select $\{10, 10, 10, 20, 50, 100\}$ images in each round and assume that the cost of labeling one image is the same.

We report our quantitative results in Table 1. Our baselines include adversarial training with other sampling methods and different training schemes with random sampling. Note that BvSB is not included here due to the fact that in the single object category detection scenario, it provides similar measurement as entropy. Overall, using adversarial training and importance weight (AADA) gives the best performance. Specifically, 60.4% accuracy can be achieved with 100 labeled target selected by AADA, while other baselines require about twice as much annotations to achieve similar performance. We further illustrate images selected with AADA within two rounds in Figure 5. As can be seen in this figure, we are able to select diverse images with different semantic layouts.

Training	Sampling	Number of Labeled Target					
		10	20	30	50	100	200
Adversarial	Imp. weight	**49.4**	**53.3**	**54.6**	**57.4**	**60.4**	**62.3**
Adversarial	K-means	49.1	51.7	53.8	56.8	59.2	60.9
Adversarial	Entropy	48.9	50.9	52.3	54.3	58.1	61.0
Adversarial	Random	47.4	49.8	51.6	55.2	58.6	61.7
Joint	Imp. weight	48.5	52.1	53.5	56.2	58.6	60.5
Joint	Random	45.5	48.8	51.8	54.9	59.0	61.6
Fine-tuning	Random	41.0	46.0	48.7	51.4	56.0	59.8
Target only	Random	29.0	38.5	42.1	48.3	53.3	58.8

Table 1: Object detection results (KITTI → Cityscapes). Our AADA method (first row) outperforms all other baselines, including using adversarial training and other sample selection methods, as well as using different training schemes and random sampling.

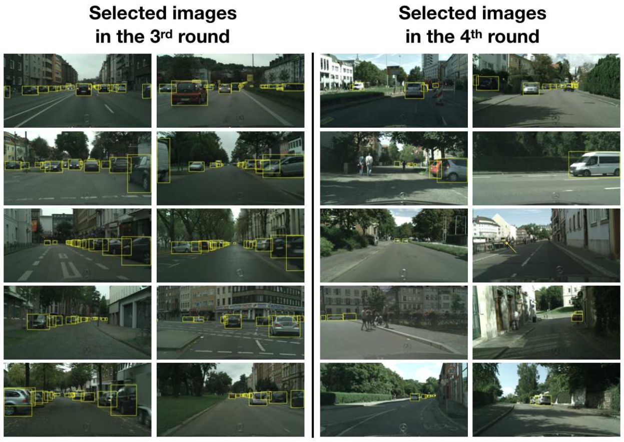

Selected images in the 3rd round
Selected images in the 4th round

Figure 5: Top 10 images selected in the third and the fourth rounds from the target domain (Cityscapes) using AADA. The ground-truth bounding boxes of cars are shown in yellow. Images selected in the third round have more cars and the semantic layouts are different w.r.t. that of the fourth round, showing that diverse samples are selected by AADA.

5.3. VisDA-18 Challenge

We investigate the VisDA-18 domain adaptation challenge [43, 44] as a special case. The source domain is composed of 78,222 synthetic images across 12 object categories rendered from 3D CAD models, while the target domain contains 5,534 real images. We consider the 12-way classification problem following the setting in [44] and the ImageNet pre-trained ResNet-18 [17] model is used as a feature extractor. As mentioned in [44], without using ImageNet pre-training, the accuracy would be very low and unsupervised domain adaption methods do not work. However, ImageNet images are closer to the target domain, this raises our interest to investigate whether images from source domain still help in this scenario.

Our initial trial using adversarial training shows improvement when there is no labeled target $L_t = \emptyset$. However, after having a few labeled targets, using adversarial training does not introduce further improvement (see supplementary material). We argue that, 1) the domain gap (from synthetic to real images) in this dataset is large, thus the benefit of aligning image features from target to source domain is less than adding annotated target images L_t, and 2) due to the use of ImageNet pre-trained model, the target domain (images from MS-COCO [34]) is actually closer to the domain for pre-training (images from ImageNet [49]) than the source domain (synthetic images).

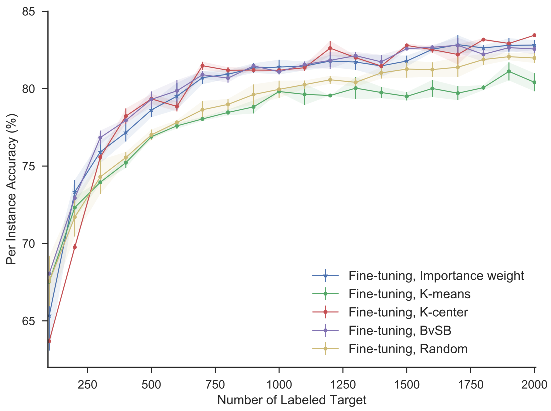


Figure 6: VisDA-18 result (synthetic → real). Here we use fine-tuning as the training scheme and compare different sampling strategies. Using importance weight for sampling performs equally well as BvSB and k-center baselines, and outperforms k-means and random baselines. The mean accuracies after each round are 79.8% and 80.1% for importance weight and BvSB methods separately.

Based on the above observations, we use fine-tuning as our training scheme on VisDA-18, and compare different sampling strategies in Figure 6. We set $b = 100$ and perform 20 rounds in total. Using importance weight for sampling performs on a par with BvSB and K-center, and outperforms K-means and random selection baselines.

6. Conclusion

We propose AADA, a unified framework for domain adaptation and active learning via adversarial training. When few labeled targets are available, the domain adversarial model helps improve the classification; meanwhile, the discriminator can be utilized to obtain the importance weight for active sample selection in the target domain. We conduct extensive ablation studies and analyses, and show improvements over other baselines with different training and sampling schemes on object recognition and detection tasks. In the future, we will consider extending our work to other settings such as open set [52], partial [5], and universal [70] domain adaptation.

Acknowledgements The project is supported in part by NSF #1749833.

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*, 79(1-2):151–175, 2010. 4

[2] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(Sep):2137–2155, 2009. 2

[3] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NeurIPS*, 2008. 4

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NeurIPS*, 2016. 2

[5] Z. Cao, L. Ma, M. Long, and J. Wang. Partial adversarial domain adaptation. In *ECCV*, 2018. 8

[6] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye. Joint transfer and batch-mode active learning. In *ICML*, 2013. 2

[7] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018. 1, 2, 7

[8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In *NeurIPS*, 1995. 2

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 7

[10] H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59. Association for Computational Linguistics, 2010. 1, 2

[11] S. Dutt Jain and K. Grauman. Active image segmentation propagation. In *CVPR*, 2016. 2, 5

[12] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *ECCV*, 2014. 2

[13] 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*, 17(1):2096–2030, 2016. 1, 2, 3

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 7

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2, 4

[16] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005. 2

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7, 8

[18] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2, 5

[19] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[20] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems (TOIS)*, 27(3):16, 2009. 2

[21] H.-K. Hsu, W.-C. Hung, H.-Y. Tseng, C.-H. Yao, Y.-H. Tsai, M. Singh, and M.-H. Yang. Progressive domain adaptation for object detection. In *CVPR Workshops*, 2019. 2

[22] W.-N. Hsu and H.-T. Lin. Active learning by learning. In *AAAI*, 2015. 2

[23] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. *arXiv preprint arXiv:1803.11365*, 2018. 2, 7

[24] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*. IEEE, 2009. 2, 5

[25] C. Kading, A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler. Active learning and discovery of object categories in the presence of unnameable instances. In *CVPR*, 2015. 2

[26] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu. Localization-aware active learning for object detection. *arXiv preprint arXiv:1801.05124*, 2018. 2

[27] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007. 2

[28] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[29] K. Konyushkova, R. Sznitman, and P. Fua. Learning active learning from data. In *NeurIPS*, 2017. 2

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[31] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 2

[32] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994. 2

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 7

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 8

[35] B. Liu and V. Ferrari. Active learning for human pose estimation. In *ICCV*, 2017. 2

[36] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013. 1, 2

[37] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*, 2016. 1, 2, 5

[38] W. Luo, A. Schwing, and R. Urtasun. Latent structured active learning. In *NeurIPS*, 2013. 2, 5

[39] C. Mayer and R. Timofte. Adversarial sampling for active learning. *arXiv preprint arXiv:1808.06671*, 2018. 2

[40] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. In *NeurIPS*, 2017. 2

[41] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011. 5

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NeurIPS Autodiff Workshop*, 2017. 5

[43] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, K. Saenko, X. Roynard, J.-E. Deschaud, F. Goulette, T. L. Hayes, et al. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *CVPR Workshops*, 2018. 2, 8

[44] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. *arXiv preprint arXiv:1806.09755*, 2018. 8

[45] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang. Two-dimensional active learning for image classification. In *CVPR*, 2008. 2

[46] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, 2010. 3

[47] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 7

[48] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001. 2

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 8

[50] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2, 7

[51] A. Saha, P. Rai, H. Daumé, S. Venkatasubramanian, and S. L. DuVall. Active supervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112. Springer, 2011. 3, 7

[52] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. Open set domain adaptation by backpropagation. In *ECCV*, 2018. 8

[53] T. Scheffer, C. Decomain, and S. Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer, 2001. 2

[54] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 2, 5

[55] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012. 2

[56] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR workshop*, 2014. 6

[57] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017. 2

[58] M. Sugiyama, M. Krauledat, and K.-R. MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005, 2007. 2, 3

[59] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NeurIPS*, 2008. 2, 3

[60] Q. Sun, A. Laddha, and D. Batra. Active learning for structured probabilistic models with histogram approximation. In *CVPR*, 2015. 2, 5

[61] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2

[62] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, 2019. 2

[63] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 1, 2

[64] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1

[65] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 1, 2

[66] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012. 2

[67] S. Vijayanarasimhan and A. Kapoor. Visual recognition and detection under bounded computational resources. In *CVPR*, 2010. 2

[68] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *European Conference on Information Retrieval*, pages 393–407. Springer, 2003. 2

[69] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014. 6

[70] K. You, Z. Cao, M. Long, J. Wang, and M. I. Jordan. Universal domain adaptation. In *CVPR*, 2019. 8

[71] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, 2018. 2

[72] J.-J. Zhu and J. Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017. 2