Stochastic Adversarial Gradient Embedding for Active Domain Adaptation

Victor Bouvier[1,2], Philippe Very[3], Clément Chastagnol[4], Myriam Tami[1], and Céline Hudelot[1]

[1] Université Paris-Saclay, CentraleSupélec, `firstname.name@centralesupelec.fr`
[2] Sidetrade, `vbouvier@sidetrade.com`
[3] Talan, `philippe.very@talan.com`
[4] IQVIA, `clement.chastagnol@iqvia.com`

Abstract. Unsupervised Domain Adaptation (UDA) bridges the gap between a labelled source domain and an unlabelled target domain. In this paper, we improve adaptation by guiding the model with actively annotated target data. This problem, named Active Domain Adaptation (ADA) is of practical interest as it is sometimes possible to annotate a small budget of target data in many applications. We introduce **S**tochastic **a**dversarial **g**radient **e**mbedding (Sage), an embedding for estimating the impact of annotating a target sample on adaptation. Sage measures the variation of the transferability loss gradient, before and after annotation. Additionally, we investigate various procedures for incorporating a small subset of labelled target samples when learning domain invariant representations. Our experiments on challenging benchmarks demonstrate that a small effort of active annotation with Sage improves adaptation substantially. Importantly, with a comparable labelling budget, Sage performs better than its semi-supervised counterpart while having more realistic assumptions.

Keywords: Domain Adaptation · Active Learning · Invariant Representations.

1 Introduction

When provided with a large amount of labelled data, deep neural networks have dramatically improved the state-of-the-art in various applications [19]. However, when deployed in the real world, where data may slightly differ from the training data, deep models often fail to generalize out of the training distribution [5]. Nevertheless, deep nets can learn data representations transferable to new tasks or new domains if some labelled data from the new distribution are available [36]. Acquiring a sufficient amount of labelled data is often impossible, and large scale annotation is often cost-prohibitive. In contrast, unlabelled data are much more convenient to obtain. This observation has motivated the field of *Unsupervised Domain Adaptation* (UDA) [23,26] for bridging the gap between a labelled *source domain* and an unlabelled *target domain*. Learning

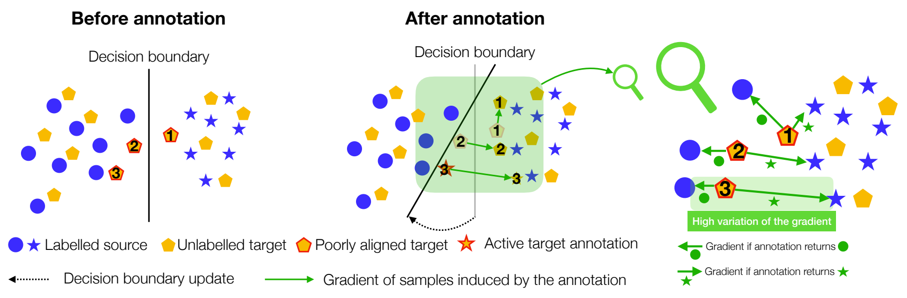


Fig. 1. Effect of annotation of a target sample selected by Sage (*best viewed in colors*). Binary classification problem ($\bullet$ *vs* $\star$) where source samples are blue and target samples are orange. Before annotation, the class-level alignment is not satisfactory leading to a potential negative transfer (poorly aligned target samples tagged as **1**, **2** and **3**). We estimate which sample should be primarily annotated by measuring the variation of the representations' transferability gradient, before and after annotation. We observe the highest variation is obtained for target sample **3**, which is sent to an oracle. The oracle annotation returns class $\star$, validating the suspicion of negative transfer. This leads to an update of the decision boundary, which pushes **1**, **2**, and **3** into class $\star$, resulting in a better class-level alignment of representations.

domain *Invariant Representations* has led to significant progress [13,21,22]. By fooling a discriminator trained to separate the source from the target domain, the feature extractor removes domain-specific information from representations. Thus, a classifier trained from those representations with source labelled data is expected to perform reasonably well in the target domain [6].

However, those methods perform significantly worse than their fully supervised counterparts. To this purpose, Semi-Supervised Domain Adaptation (SSDA) has been studied in [31] through a Mini-Max Entropy objective (MME). Nevertheless, assuming that at least one target labelled sample represents a class, thus involving information about target labels, SSDA is built on assumptions that are unlikely to be met in practice. A more realistic scenario would be to guide adaptation by selecting for annotation a pool target unlabelled instances. This new paradigm referred to as *Active Domain Adaptation* (ADA), is often encountered in real-world applications. To our knowledge, only a few prior works address ADA [9,27,30,34]. In particular, the recent work of Su et al. [34] is the first that uses domain adversarial learning for *Active Learning* (AL).

In this paper, we address ADA reserving the annotation budget for target samples for which their annotations are likely to guide adaptation. In contrast to [34], which selects a diverse set of poorly adapted target samples based on a classical criterion of uncertainty, we estimate the impact of annotation on the representations' transferability. To this purpose, we introduce **S**tochastic **a**dversarial **g**radient **e**mbedding (Sage), an embedding of target samples, whose norm estimates precisely this impact. Our approach also promotes diversity in the annotation. We follow [3] and select target samples for which Sage spans on diverse directions using the `k-means++` initialization [1]. Since access to some labelled data from the target domain brings us back to SSDA, we investigate the role of MME in this context.

We organize the rest of the paper as follows. First, we provide a brief overview of Domain Adversarial Learning for UDA. Importantly, we expose a soft-class conditioning adversarial loss, which reflects the transferability error of domain invariant representations [7]. Second, we present the details of Sage while providing theoretical insights that AL can improve representations' transferability in the third section. Finally, we conduct an empirical study on several benchmarks that support our ADA approach.

2 Background

Notations. Let us consider three random variables; X the input data, Z the representations and Y the labels, defined on spaces $\mathcal{X}$, $\mathcal{Z} \subset \mathbb{R}^m$ where m is the dimension of the representation, and $\mathcal{Y}$ such that $|\mathcal{Y}| = C$ for some integers C, respectively. We note realizations with lower cases, x, z and y, respectively. Those random variables may be sampled from two and different distributions: the *source* distribution p_S *i.e.*, data where the model is trained and the *target* distribution p_T *i.e.*, data where the model is evaluated. Labels are one-hot encoded *i.e.*, $y \in [0,1]^C$ with $\sum_c y_c = 1$ where C is the number of classes. We use the index notation S and T to differentiate source and target quantities. We define the hypothesis class $\mathcal{H}$ as a subset of functions from $\mathcal{X}$ to $\mathcal{Y}$ which are the composition of a representation class Φ (mappings from $\mathcal{X}$ to $\mathcal{Z}$) and a classifier class $\mathcal{F}$ (mappings from $\mathcal{Z}$ to $\mathcal{Y}$) *i.e.*, $h := f\varphi := f \circ \varphi \in \mathcal{H}$ where $f \in \mathcal{F}$ and $\varphi \in \Phi$. For $D \in \{S, T\}$ and a hypothesis $h \in \mathcal{H}$, we introduce the error in domain D, $\varepsilon_D(h) := \mathbb{E}_D[\ell(h(X), Y)]$ where ℓ is the L^2 loss $\ell(y, y') = ||y - y'||^2$ and $h(x)_c$ is the probability of x to belong to class c. We note the source domain data $(x_i^S, y_i^S)_{1 \leq i \leq n_S}$ and the target domain data $(x_j^T)_{1 \leq j \leq n_T}$.

Domain Adversarial Learning. The seminal works from [13,21], and their theoretical ground [6], have led to a wide variety of methods based on domain invariant representations [22,20,10]. A representation φ and a classifier f are learned by achieving a trade-off between source classification error and domain invariance of representations by fooling a discriminator trained to separate the source from the target domain:

$$L(\varphi, f) := L_S(\varphi, f) - \lambda \cdot \inf_{d \in \mathcal{D}} L_{\text{INV}}(\varphi, d) \tag{1}$$

where $L_S(\varphi, f) := \mathbb{E}_S[-Y \cdot \log(f\varphi(X))]$ is the cross-entropy loss in the source domain, $L_{\text{INV}}(\varphi, d) := \mathbb{E}_S[\log(1 - d(\varphi(X)))] + \mathbb{E}_T[\log(d(\varphi(X))]$ is the adversarial loss and $\mathcal{D}$ is the set of discriminators *i.e.* mapping from $\mathcal{Z}$ to $[0,1]$. In practice, $\inf_{d \in \mathcal{D}}$ is approximated using a *Gradient Reversal Layer* [13].

Transferability loss for class-level invariance. Promoting class-level domain invariance improves the transferability of representations [22]. Recently, the work [7] introduces the *transferability loss*, noted L_{TSF}, which adds class-conditioning in the adversarial loss by computing a scalar product between labels y and a class-level discriminator d defined as a mapping from $\mathcal{Z}$ to $[0,1]^C$. Since labels are not available in the target domain at train time, predicted labels $\hat{y} := f\varphi(x)$ are used. This approach is referred to as *soft-class conditioning*:

$$L(\varphi, f) := L_S(\varphi, f) - \lambda \cdot \inf_{\mathsf{d} \in \mathsf{D}} L_{\text{TSF}}(\varphi, \hat{y}, \mathsf{d}) \tag{2}$$

where $L_{\text{TSF}}(\varphi, \hat{y}, \mathsf{d}) := \mathbb{E}_S[Y \cdot \log(1 - \mathsf{d}(\varphi(X)))] + \mathbb{E}_T[\hat{Y} \cdot \log(\mathsf{d}(\varphi(X))]$ is the transferability loss and D is the set of class-level discriminators *i.e.*, mappings from $\mathcal{Z}$ to $[0,1]^C$. In this work, we explore the role of active annotation on a small subset of the target domain in order to improve the transferability of representations. Methods based on L_{TSF} as adaptation loss are flagged as TSF.

3 Proposed Method

3.1 Motivations

Gradient-based selection, as shown in Badge [3], is promising in AL. In contrast to Badge, which focuses on the network's predictions, we discuss the role of representations' transferability. To this purpose, we introduce, in the following, the *adversarial gradient* that reflects the lack of transferability of a target sample. From this gradient, we expose a query that efficiently incorporates the domain shift problem in ADA. Let a target sample $x \sim p_T$ with representation $z := \varphi(x) \in \mathbb{R}^m$, we start by describing the effect of annotating the sample x on the gradient descent update of 2. We define the *adversarial gradient* $\mathbf{g}_x$ of x as the gradient of the discriminator loss w.r.t the representation z:

$$\mathbf{g}_x := -\frac{\partial \log(\mathsf{d}(z))}{\partial z} \in \mathbb{R}^{C \times m}, \quad \text{where } \mathsf{d}(z) \in [0,1]^C \tag{3}$$

Following the expression of the transferability loss L_{TSF}, the contribution of a sample x to the gradient update (2), before and after its annotation, is:

$$\underbrace{\left\{\theta \leftarrow \theta - \alpha \frac{\partial z}{\partial \theta} \cdot (\hat{y} \cdot \mathbf{g}_x)\right\}}_{\textit{Before annotation}} \longrightarrow \underbrace{\left\{\theta \leftarrow \theta - \alpha \frac{\partial z}{\partial \theta} \cdot (y \cdot \mathbf{g}_x)\right\}}_{\substack{\textit{After annotation} \\ y \sim \text{Oracle}(x)}}$$

where $\partial z / \partial \theta$ is the jacobian of the representations with respect to the deep network parameters θ *i.e.*, $z := \varphi_\theta(x)$, $\hat{y} := f\varphi_\theta(x)$ is the current label estimation and α is some scaling parameter. Before the annotation, the gradient vector can be written as a weighted sum of $\mathbf{g}_x$ *i.e.*, $\hat{y} \cdot \mathbf{g}_x \in \mathbb{R}^m$, reflecting the class probability of x. Annotating the sample x has the effect of setting, once and for all, a direction of the gradient $(y \cdot \mathbf{g}_x)$. Based on this observation, we can measure the annotation procedure's ability to learn more transferable representations by its tendency to change the path of the gradient descent *i.e.*, how $y \cdot \mathbf{g}_x$ may differ with $\hat{y} \cdot \mathbf{g}_x$.

3.2 Positive Orthogonal Projection (POP)

In the rest of the paper, we consider $\mathbf{g}_x \in \mathbb{R}^{C \times m}$ as a stochastic vector of $\mathbb{R}^m$ with realizations lying in $\mathcal{G}_x := \{\mathbf{g}_x^1, ..., \mathbf{g}_x^C\}$ where $\mathbf{g}_x^c = (-\partial \log(\mathsf{d}(z))/\partial z)_c$. When provided the label through an oracle *i.e.*, $y \sim \text{Oracle}(x)$, we obtain $\mathbf{g}_x^y \in \mathcal{G}_x$, a realization of $\mathbf{g}_x$. Before annotation, the direction of the gradient is the *mean* of $\mathbf{g}_x$ where $\mathcal{G}_x$ is provided with the class probability given by the classifier's output $h(x)$. More precisely, the probability of observing $\tilde{\mathbf{g}}_x = \tilde{\mathbf{g}}_x^c$ is $h(x)_c$, then, the *mean* of $\mathbf{g}_x$, noted $\mathbb{E}_h[\mathbf{g}_x]$ is defined as follows:

$$\mathbb{E}_h[\mathbf{g}_x] := \mathbb{E}_{y \sim h(x)}\left[\mathbf{g}_x^y\right] = h(x) \cdot \mathbf{g}_x \in \mathbb{R}^m \tag{4}$$

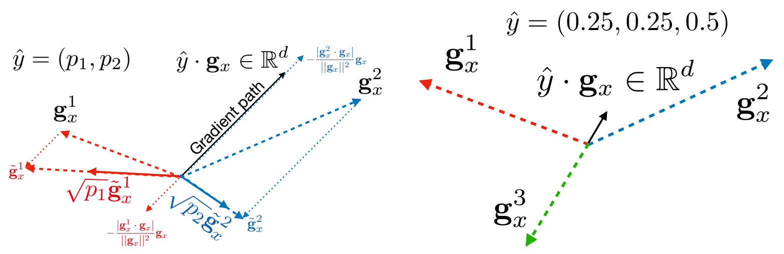
Therefore, the tendency to modify the direction of the gradient is reflected by a high discrepancy between $\mathbb{E}_h[\mathbf{g}_x]$ and $\mathbf{g}_x^y$ for $y \sim \text{Oracle}(x)$. To quantify this discrepancy, we consider variations in both direction and magnitude. To find a good trade-off between these two requirements, we remove the mean direction of the gradient $\mathbb{E}_h[\mathbf{g}_x]$ to $\mathbf{g}_x$ by computing a *Positive Orthogonal Projection* (POP), noting $\lambda := |\mathbf{g}_x \cdot \mathbb{E}_h[\mathbf{g}_x]| / ||\mathbb{E}_h[\mathbf{g}_x]||^2$;

$$\tilde{\mathbf{g}}_x := \mathbf{g}_x - \lambda \mathbb{E}_h[\mathbf{g}_x] \tag{5}$$

We motivate the use $|\mathbf{g}_x \cdot \mathbb{E}_h[\mathbf{g}_x]|$ rather than $\mathbf{g}_x \cdot \mathbb{E}_h[\mathbf{g}_x]$ for the standard orthogonal projection. On the one hand, if the annotation provides a gradient with the same direction as the expected gradient *i.e.*, the annotation reinforces the prediction, $\tilde{\mathbf{g}}_x$ is null. On the other hand, if the annotation provides a gradient with an opposite direction to the expected gradient *i.e.*, the annotation contradicts the prediction, the norm of $\tilde{\mathbf{g}}_x$ increases. Therefore, target samples x for which we expect the highest impact on the transferability, are those with the highest norm of $\tilde{\mathbf{g}}_x$. Since λ involves an absolute value, we refer to it as a *positive* orthogonal projection. An illustration is provided in Figure 2. Since $\tilde{\mathbf{g}}_x$ is stochastic, we need additional tools to define a norm operator properly on it.

3.3 Stochastic Adversarial Gradient Embedding

It seems natural to quantify the norm of the stochastic vector $\tilde{\mathbf{g}}_x$ as the square root of the mean of $\tilde{\mathbf{g}}_x$'s norm: $||\tilde{\mathbf{g}}_x||_h := (\mathbb{E}_{y \sim h(x)}\left[||\tilde{\mathbf{g}}_x^y||^2\right])^{1/2}$. However, given x_1 and x_2, how to quantify the discrepancy between $\mathbf{g}_{x_1}$ and $\mathbf{g}_{x_2}$? The difficulty results from the fact that $h(x_1) \neq h(x_2)$ in general. Simply using $\mathbb{E}_{y_1 \sim h(x_1), y_2 \sim h(x_2)}$ $\left[||\mathbf{g}_{x_1}^{y_1} - \mathbf{g}_{x_2}^{y_2}||^2\right])^{1/2}$ leads to an operator that returns a non-null discrepancy between x and itself if $h(x)$ is not a one-hot vector. To address this issue, we



$\hat{y} = (p_1, p_2)$
$\hat{y} \cdot \mathbf{g}_x \in \mathbb{R}^d$
$-\frac{|\mathbf{g}_x^2 \cdot \mathbf{g}_x|}{||\mathbf{g}_x||^2}\mathbf{g}_x$
$\mathbf{g}_x^2$
Gradient path
$\mathbf{g}_x^1$
$\tilde{\mathbf{g}}_x^1$
$\sqrt{p_1}\tilde{\mathbf{g}}_x^1$
$\sqrt{p_2}\tilde{\mathbf{g}}_x^2$
$\tilde{\mathbf{g}}_x^2$
$-\frac{|\mathbf{g}_x^1 \cdot \mathbf{g}_x|}{||\mathbf{g}_x||^2}\mathbf{g}_x$
$\hat{y} = (0.25, 0.25, 0.5)$
$\mathbf{g}_x^1$
$\hat{y} \cdot \mathbf{g}_x \in \mathbb{R}^d$
$\mathbf{g}_x^2$
$\mathbf{g}_x^3$

(a) Sage (b) Poor local min

Fig. 2. (a) Visualisation of $\mathsf{S}(x) = (\sqrt{p_1}\tilde{\mathbf{g}}_x^1, \sqrt{p_2}\tilde{\mathbf{g}}_x^2)$. Here $\tilde{\mathbf{g}}_x^2 \perp (\hat{y} \cdot \mathbf{g}_x)$ since $\hat{y} \cdot \mathbf{g}_x$ and $\mathbf{g}_x^2$ have a similar direction while $|\tilde{\mathbf{g}}_x^1 \cdot \mathbf{g}_x| \geq |\mathbf{g}_x^1 \cdot \mathbf{g}_x|$ since $\mathbf{g}_x^1$ as a component in the opposite direction of $\hat{y} \cdot \mathbf{g}_x$. (b) Illustration of a case where the transferability loss is close to a local minimum ($\hat{y} \cdot \mathbf{g}_x \approx 0$), but the stochastic gradients ($\mathbf{g}_x^y$ for $y \in \{1, 2, 3\}$) have a high norm. Here, the annotation chooses one of the gradients resulting in a strong update of the model.

suggest to embed x, through a mapping S named *Stochastic adversarial gradient embedding* (Sage):

$$\mathsf{S}(x) := (\sqrt{h(x)_1}\tilde{\mathbf{g}}_x^1, ..., \sqrt{h(x)_C}\tilde{\mathbf{g}}_x^C) \in \mathbb{R}^{C \times m} \tag{6}$$

By choosing $\sqrt{h}$, we guarantee that $||\mathsf{S}(x)|| = ||\tilde{\mathbf{g}}_x||_h$ while offering a proper discrepancy between $\mathbf{g}_{x_1}$ and $\mathbf{g}_{x_2}$ with $||\mathsf{S}(x_1) - \mathsf{S}(x_2)||$. Crucially, both the norm and the distance computed on Sage do not involve the target labels, making it relevant for UDA since target labels are unknown. An illustration of Sage is provided in Figure 2.

3.4 Increasing Diversity of Sage (`k-means++`)

As aforementioned, the higher the norm of $||\mathsf{S}(x)||$, the greater the expected impact of annotating sample x on the transferability of representations. A naive strategy of annotation would be to rank target samples by their Sage norm ($||\mathsf{S}(x)||$). The drawback is to acquire labels for a not IID batch from the target distribution, a problem referred to as the challenge of *diversity* in AL [33]. In certain pathological cases (e.g. the selection of very similar samples or samples of the same class), the IID violation may degrade the performance in the target domain. To label useful target samples (*i.e.*, high $||\mathsf{S}(x)||$) while acquiring a representative batch of the target distribution, we follow [3] by selecting samples with high $||\mathsf{S}(x)||$ which span in various directions. This is performed using the `k-means++` initialization [1]. The procedure `Sage` is detailed in Algorithm 1 for a given budget b of annotation. Importantly, sampling diverse target samples with high impact on transferability results from the construction of an embedding (Sage) suitable with `k-means++`.

Algorithm 1 $\texttt{Sage}(\mathcal{U}_T, b, f, \varphi, \mathsf{d})$: Sage with diversity (`k-means++`)

```
Input: $\mathcal{U}_T$: Unlabelled target data, budget $b$, representation $\varphi$, classifier $f$, discriminator $\mathsf{d}$
1: Computes $\mathsf{S}(x_u)$ for $x_u \in \mathcal{U}_T$        ▷ Depends on both $f$ and $\varphi$.
2: $\mathcal{A} \leftarrow \{\text{argmax}_{x_u \in \mathcal{U}_T} ||\mathsf{S}(x_u)||\}$        ▷ Select sample with the highest Sage norm.
3: while $|\mathcal{A}| < b$ do        ▷ Apply k-means++ on Sage embedding.
4:     $\mathcal{A} \leftarrow \mathcal{A} \cup \{\underset{x_u \in \mathcal{U}_T}{\text{argmax}} \underset{x_a \in \mathcal{A}}{\min} ||\mathsf{S}(x_u) - \mathsf{S}(x_a)||\}$
5: end while
6: Return $\mathcal{A}$
```

3.5 Semi-Supervised Domain Adaptation (SSDA)

When acquiring labels in the target domain, we are in the Semi-Supervised Domain Adaptation (SSDA) setting. To this purpose, we note $\mathcal{L}_S$ and $\mathcal{L}_T$ the sets of labelled samples from the source and the target domains, respectively. We study three strategies, referred to as $S{\cup}T$, $S{+}T$ and MME [31]. They incorporate labelled samples into adaptation through an additional loss Ω, called a SSDA regularizer:

$$\Omega_{S\cup T}(f, \varphi) := L_{\mathcal{L}_S \cup \mathcal{L}_T}(f, \varphi) \tag{7}$$

$$\Omega_{S+T}(f, \varphi) := L_{\mathcal{L}_S}(f, \varphi) + L_{\mathcal{L}_T}(f, \varphi) \tag{8}$$

noting $L_{\mathcal{L}}(f, \varphi)$ the empirical cross-entropy of $f\varphi$ computed on some labelled dataset $\mathcal{L}$. Note that Ω_{S+T} gives more importance to target labelled samples compared to $\Omega_{S\cup T}$, especially in the small budget regime (*i.e.*, when the budget b is such that $b \ll |\mathcal{L}_S|$). As a strong baseline exists in SSDA, we design Ω following the minimax entropy (MME) [31]. Noting $H_{\mathcal{U}_T}(h) := -\frac{1}{|\mathcal{U}_T|}\sum_{x \in \mathcal{U}_T} h(x) \cdot \log h(x)$, the entropy of unlabelled samples $\mathcal{U}_T$, the MME objective is:

$$\begin{cases} \Omega_{\text{MME}}(f) := \Omega_{S+T}(f, \varphi) - \lambda H_{\mathcal{U}_T}(f\varphi) \\ \Omega_{\text{MME}}(\varphi) := \Omega_{S+T}(f, \varphi) + \lambda H_{\mathcal{U}_T}(f\varphi) \end{cases} \tag{9}$$

where $f := \sigma\left(\frac{1}{T} W \circ \ell_2\right)$ ($\ell_2(\mathbf{f}) := \mathbf{f}/||\mathbf{f}||_2$ is the L^2 normalization of features and $W \in \mathbb{R}^{C\times m}$ is a linear layer), $\lambda = 0.1$, $T = 0.05$ and σ is the softmax layer.

3.6 Training procedure

The training procedure is described in Algorithm 2. First, we train the model by UDA following the training procedure from [7]. Second, for a given number of iterations, we select by `Sage` (See Algorithm 1) b samples to send to the Oracle. Then, we perform UDA provided with the knowledge of newly labelled samples, that is using a SSDA regularizer Ω combined with soft-class conditioning loss L_{TSF}. We describe the gradient descent step in the following. First, given a loss L, Given a SSDA regularizer Ω (See Section 3.5), the gradient descent step is defined as follows, for some $\alpha > 0$:

$$(f, \varphi, \mathsf{d}) \leftarrow (f, \varphi, \mathsf{d}) - \alpha \nabla_{(f,\varphi,\mathsf{d})}\left(\hat{\Omega}(f, \varphi) + \lambda \hat{L}_{\text{TSF}}(f, \varphi)\right) \tag{10}$$

Algorithm 2 Training procedure

```
Input: Labelled source samples $\mathcal{L}_S$, Unlabelled target samples $\mathcal{U}_T$, budget $b$, annotation rounds $r$, iterations $n_{\text{it}}$, SSDA regularizer $\Omega$:
 1: $\mathcal{L}_T \leftarrow \{\}, \mathcal{U}'_T \leftarrow \mathcal{U}_T$                 ▷ Initializes the labelled target samples.
 2: $f, \varphi, \mathsf{d} \leftarrow$ UDA as described in [7]               ▷ Pretraining before Active Learning.
 3: for $b$ rounds of annotations do
 4:     $\mathcal{A} \leftarrow \texttt{Sage}(\mathcal{U}'_T, b, f, \varphi, \mathsf{d})$                 ▷ Selects samples for annotation.
 5:     $\mathcal{L} \leftarrow$ Oracle($\mathcal{A}$)                              ▷ Sends samples to an Oracle.
 6:     $\mathcal{L}_T \leftarrow \mathcal{L}_T \cup \mathcal{L}$                           ▷ Adds newly labelled samples.
 7:     $\mathcal{U}'_T \leftarrow \mathcal{U}'_T \backslash \mathcal{A}$                          ▷ Removes newly labelled samples.
 8:     for $n_{\text{it}}$ iterations do
 9:         Sample a source labelled batch $\mathcal{B}^\ell_S$ from $\mathcal{L}_S$
10:         Sample a source labelled batch $\mathcal{B}^\ell_T$ from $\mathcal{L}_T$
11:         Sample a source labelled batch $\mathcal{B}^u_T$ from $\mathcal{U}_T$            ▷ (Not from $\mathcal{U}'_T$).
12:         $f, \varphi, \mathsf{d} \leftarrow$ Gradient descent update from Equation 10.
13:     end for
14: end for
15: Return: $f, \varphi$
```

where for a given loss L, we note its batch-wise computation $\hat{L}$ when provided with batches of source labelled samples $\mathcal{B}^\ell_S$ from $\mathcal{L}_S$, a source labelled samples $\mathcal{B}^\ell_T$ from $\mathcal{L}_T$, a source labelled samples $\mathcal{B}^u_T$ from $\mathcal{U}_T$. Notably, $\mathcal{B}^\ell_S$ and $\mathcal{B}^\ell_T$ are involved for computing $\hat{\Omega}$ (eventually $\mathcal{B}^u_T$ for $\hat{\Omega}_{\text{MME}}$) while $\mathcal{B}^\ell_S$ and $\mathcal{B}^u_T$ are involved for computing $\hat{L}_{\text{TSF}}$.

4 Theoretical Analysis

4.1 General bound

We provide a theoretical analysis of guiding adaptation with AL. It leverages recent results from [7]. Our insight is that some labelled data from the target domain, when combined with source labelled data, are likely to improve the target error. For instance, minimizing Ω_{S+T} may result in a better performing classifier than simply minimizing the source cross-entropy loss $L_{\mathcal{L}_S}$. The theoretical framework from Bouvier et al. [7] allows to quantify precisely how it impacts representations' transferability. Noting $h_S := \text{argmin}_{h\in\mathcal{H}} \varepsilon_S(h)$ and $h_\Omega := \text{argmin}_{h\in\mathcal{H}} \Omega(h)$ such that $\varepsilon_T(h_\Omega) \leq \beta\varepsilon_T(h_S)$ for some $\beta < 1$ *i.e.*, h_Ω improves the target error compared to h_S, the work [7] bounds the target error;

$$\varepsilon_T(h_\Omega) \leq \rho(\varepsilon_S(h_S) + 8\tau + \eta) \quad \text{where} \quad \rho := \frac{\beta}{1-\beta} \tag{11}$$

where $\tau := \sup_{\mathsf{f}\in\mathsf{F}}\{\mathbb{E}_T\left[h_\Omega(X)\cdot\mathsf{f}(\varphi(X))\right] - \mathbb{E}_S[Y\cdot\mathsf{f}(\varphi(X))]\}$ is the transferability error, F is the set of continuous functions from $\mathcal{Z}$ to $[-1,1]^C$, $\eta := \inf_{\mathsf{f}\in\mathsf{F}} \varepsilon_T(\mathsf{f}\varphi)$. Thus, to guarantee a small target error, the following conditions have to be met: a small source error of h_S (small $\varepsilon_S(h_S)$), a small transferability error of

h_{Ω} (small τ) and a strong inductive bias (small β) while we assume η small [7]. ADA incorporates a small set of target labelled samples into Ω to strengthen the inductive bias while enforcing a small transferability error of h_{Ω}. More details on the choice of Ω are given in Section 3.5.

4.2 A particular case with closed form

Setup and additional notations. In this section, we provide a simple example where the bound presented in Section 4 has a closed form. To conduct the analysis, we consider $\mathcal{X}$ as a measurable set provided with a probability measure noted p_T. We present an extension of an annotation selection to a measurable set. Selecting samples for annotation with budget b consists in determining some measurable subset $\mathcal{B}$ such that $p_T(X \in \mathcal{B}) = b$. In the particular case where $p_T := \sum_{x \in \mathcal{D}_T} \delta_x$ (δ_x is the Dirac distribution in x) is an empirical distribution, determining some measurable subset $\mathcal{B}$ such that $p_T(X \in \mathcal{B}) = b$ consists in determining a subset of b samples of $\mathcal{D}_T$.

Naive Active Classifier. Given a classifier h and an annotated subset $\mathcal{B}$ (with probability b), we suggest a slight modification of the classifier h based on the annotation provided by the Oracle of $\mathcal{B}$. To this purpose, we introduce the *naive active classifier*, noted $h_{\mathcal{B}}(x)$, and defined as follows:

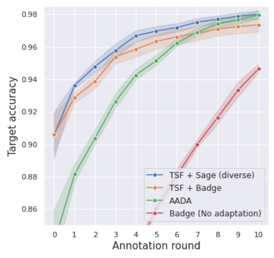
$$h_{\mathcal{B}}(x) = \text{Oracle}(x) \text{ if } x \in \mathcal{B}, h(x) \text{ otherwise.} \tag{12}$$

Thus, $h_{\mathcal{B}}(x)$ returns the classifier's output $h(x)$ if x is not annotated and returns the oracle's output $\text{Oracle}(x)$ if x is annotated.

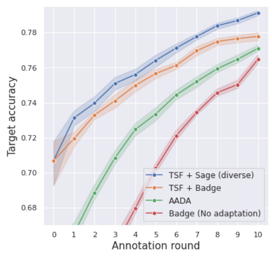
A closed bound. We want to exhibit a closed form of ρ when considering the active classifier. To this purpose, we introduce the *purity* π of $\mathcal{B}$, $\pi := p_T(h_S(X) \neq \text{Oracle}(X) | X \in \mathcal{B})$. It reflects our capacity to identify misclassified target samples. With this notion, we observe that the naive classifier improves the target error; $\varepsilon_T(h_{\mathcal{B}}) \leq \varepsilon_T(h_S) - b\pi$. Put simply, the error is reduced by $b\pi$ corresponding to annotated samples for which the prediction is different from the Oracle output. The higher the budget of annotation b and the higher the purity π, the lower the target error of the naive classifier. It corresponds to $\varepsilon_T(h_S) - b\pi = \left(1 - \frac{b\pi}{\varepsilon_S(h_S)}\right) \varepsilon_S(h_S) \leq (1 - b\pi)\varepsilon_S(h_S)$; resulting into $\beta = (1 - b\pi)$, and finally:

$$\varepsilon_T(h_{\mathcal{B}}) \leq \left(\frac{1}{b\pi} - 1\right)(\varepsilon_S(h_S) + 8\tau + \eta) \tag{13}$$

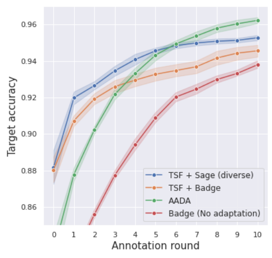
The target error of the active classifier is a decreasing function of both the purity and the annotation budget and an increasing function of the transferability error. The budget b, the purity π and the transferability of representations τ are levers to improve the naive classifier target error. The budget b must be considered as a cost constraint and not as a parameter to be optimized. The purity of π is not tractable since it involves labels in the target domain. Some proxy measures,



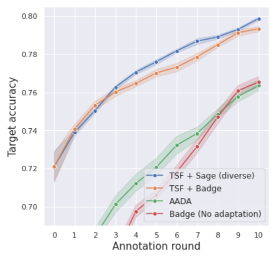
(a) A→W



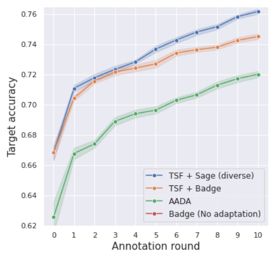
(b) W→A



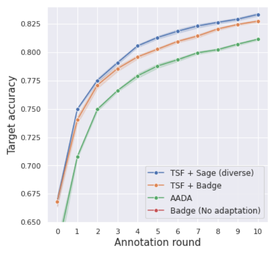
(c) A→D



(d) D→A



(e) VisDA($b = 16$)



(f) VisDA($b = 128$)

Fig. 3. Annotation of target samples improves adaptation drastically for the considered tasks. TSF+Sage (in blue) improves upon the state-of-the-art of ADA (AADA, in green), except for task A→D. AL (Badge, in red) performs poorly in this context (Badge without adaptation does not appear on VisDA tasks since it performs poorly: 47.0% and 63.4% after 10 rounds of annotation for $b = 16$ and $b = 128$, respectively) showing the importance of addressing the problem of adaptation for AL under distribution shift. Naively combining Badge with TSF (TSF+Badge, in orange) performs worsen than Sage. Sage takes into account the problem of domain shift when querying samples.

such as the entropy of predictions [14], can provide a fair estimation of purity. However, it is known that deep nets tend to be overconfident on misclassified samples [12]. Therefore, we focus our efforts on understanding the role of active annotation in improving transferability error τ.

5 Experiments

5.1 Setup

Tasks. We evaluate our approach on **Office-31** [29], **VisDA-2017** [25] and **DomainNet** [24]. Office-31 contains 4,652 images classified in 31 categories across three domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). We explore tasks **A** → **W**, **W** → **A**, **A** → **D** and **D** → **A**. We do not report results for tasks **D** → **W** and **W** → **D** since these tasks have already nearly perfect results in UDA [22]. For VisDA, we explore **Synthetic**: 3D models with different lighting conditions and different angles; **Real**: real-world images. We explore the **Synthetic** → **Real** task. **DomainNet** [24] is a large scale dataset with six domains and 345 classes (Clipart (**C**), Infograph (**I**), Painting (**P**), Quickdraw (**Q**), Real (**R**) and Sketch (**S**)). As **DomainNet** suffers of noisy labels, thus violates the assumption of a perfect Oracle, we focus on the subset of 126 classes and the 7 tasks **R**→**C**, **R**→**P**, **P**→**C**, **C**→**S**, **S**→**P**, **R**→**S** and **P**→**R** [31].

Protocol. The standard protocol in UDA uses the same target samples during train and test phases. In AL's context, this induces an undesirable effect where sample annotation mechanically increases the accuracy. At train time, the model has access to input and label of annotated samples which are also present at test time. We suggest instead to split the target domain into a *train target domain* (samples used for adaptation and pool of data used for annotation) and *test target domain* (samples used for evaluating the model) with a ratio of 1/2. Therefore, samples from the test target domain have never been seen at train time. As a result, our protocol evaluates the model generalization in an inductive scenario. Reported results are based on 8 seeds for each method.

Budget, rounds and backbone. As the selected datasets are of different volumetry and difficulty, we used different budgets b: $b = 8$ for A→W and A→D (referred to as *easy tasks*), $b = 16$ for W→A and D→A (referred to as *medium tasks*), both $b = 16$ and $b = 128$ for VisDA (referred to as *hard tasks*). This allows to appreciate versatility of methods in *small* ($b = 8$), *medium* ($b = 16$) and *high* ($b = 128$) budget regimes. We conduct 10 rounds of annotation for these tasks. Additional details for DomainNet experiments are provided in comparison with SSDA in Section 5.2. Our backbone is a ResNet50 [16] trained by 10k steps of SGD by UDA before annotation. We use DANN [13] for AADA, MME [31] for MME based methods and TSF [7] for TSF based methods.

Baselines. **AADA** [34] is the closest algorithm to Sage. AADA adapts representations by fooling a domain discriminator d trained to output 1 for source data and 0 for target data [13] and scores target samples x; $s(x) := H(\hat{y})w(z)$ where $H(\hat{y})$ is the entropy of predictions $\hat{y}$ and $w(z) = (1 - d(z))/d(z)$. $H(\hat{y})$ brings information about uncertainty while $w(z)$ brings diversity to the score. We have reproduced the implementation of AADA. To demonstrate the effectiveness of Sage for Active DA, we report TSF with **Badge** query [3] (**TSF+Badge**), which is the state-of-the-art query in AL. For these methods, we used $\Omega = \Omega_{S+T}$. To compare Sage with an AL method which ignores domain shift between labelled samples and queried samples, we report **Badge** with $\Omega_{S \cup T}$. Finally, to compare with SSDA approaches, we build two methods upon **MME** [31] with **Entropy** query (selection samples with highest prediction entropy [35]), noted **MME+Entropy**, which is the most natural query for MME since it relies on max/min entropy, and with **Random** query noted **MME+Random**. We have reproduced the implementation of MME.

5.2 Results

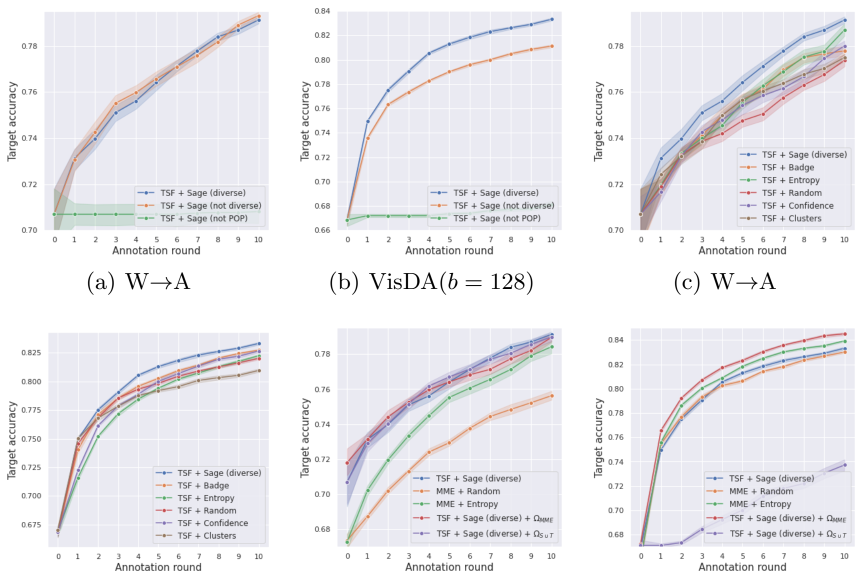
Comparison with SOTA. Results are reported in Figure 3. First, active annotation brings substantial improvements to UDA (round 0 of annotation). This validates the effort and the focus that should be put on ADA, in our opinion. Sage outperforms the current state-of-the-art (AADA) with a comfortable margin for tasks with medium or hard difficulty, except for tasks A→D after the 5-th round. Importantly, Sage performs similarly or better than naively combining TSF with a state-of-the-art query in AL (Badge) demonstrating that Sage takes into account the problem of domain shift in the query process. Finally, using a direct AL method (Badge) fails in the context of domain shift.

Ablation of Sage. We ablate the core components of Sage *i.e.*, POP and the `k-means++` in Figures 4(a) and 4(b). Interestingly, Sage without POP fails to improve performances in the target domain. This demonstrates that POP brings information about uncertainty into the embedding. Sage without diversity performs poorly on VisDA($b = 128$), demonstrating that `k-means++` based sampling brings diversity. Diversity on Sage has a small effect on W→A.

Ablation of queries. We ablate in Figures 4(c) and 4(d) more AL strategies : (**Random**), where target samples are selected at random, **Clusters** that selects the closest samples to b clusters of representations obtained with k-means, **Entropy** based on the highest entropy $\max_{x \in \mathcal{U}_T} -h(x) \cdot \log h(x)$ [35] and **Confidence** that used the smallest confidence $\min_{x \in \mathcal{U}_T} \max_c h(x)_c$ [35]. Sage is compared with a wide spectrum of AL queries based on representative (Random), diversity (Clusters) and uncertainty sampling (Entropy, Confidence). Sage outperforms them substantially on the two tasks, demonstrating it is well-suited for ADA.

Ablation of Ω. We report **TSF+$\boldsymbol{\Omega}_{\mathbf{S \cup T}}$** and **TSF+$\boldsymbol{\Omega}_{\mathrm{MME}}$** which consists in adding MME as a regularization of TSF *i.e.*, Ω used here is $\Omega_{S \cup T}$ and Ω_{MME}, respectively. Results are reported in Figures 4(e) and 4(f). We observe that using Ω_{S+T} and Ω_{MME} improve consistently wrt $\Omega_{S \cup T}$ on VisDA($b = 128$) while performing similarly on W→A. Furthermore, we observe that adding MME to TSF+Sage achieves the best performances on VisDA($b = 128$). Importantly, MME+Entropy is already strong for VisDA($b = 128$) explaining the substantial improvement when adding MME to TSF for this task.

ADA vs SSDA: ADA is a more realistic setting. We compare SSDA (a fixed number of labelled target samples per class are available) with ADA (an Oracle provides ground-truth for queried target samples) when the number of target labelled samples are equal. Crucially, enforcing a fix number of labelled samples per class is unrealistic in practice. We report performances on DomainNet of MME (1 and 3 shot) [31] and Sage (here we used TSF + Sage + Ω_{MME}). AL is performed during 6 rounds with $b = 21$ and $b = 63$ for 1 and 3 shot respectively,



(a) W→A (b) VisDA($b = 128$) (c) W→A

(d) VisDA($b = 128$) (e) W→A (f) VisDA($b = 128$)

Fig. 4. (a) and (b): Both the POP and `k-means++` are crucial components for the empirical success of Sage. (c) and (d): Sage outperforms AL query based on representative, diversity and uncertainty samplings. (e) and (f): Effect of adding MME to TSF+Sage.

leading to the same number of target labelled samples[5]. Results are presented in Table 1. In the 3-shot scenario Sage improves upon MME on all the tasks, except P→R. In the 1-shot scenario, Sage and MME perform similarly. This demonstrates that active annotation with Sage performs equally, or better, than MME, and benefits from more realistic assumptions.

6 Related works

Transferability of Invariant Representations. Recent works warn that domain invariance may deteriorate transferability of invariant representations [18,38]. Prior works enhance their transferability with multi-linear conditioning of representations with predictions [22], by introducing weights [8,37,11], by penalizing high singular value of representations batch [10] or by hallucinating consistent target samples for bridging the domain gap [20].

Active Learning. There is an extensive literature on Active Learning [33] that can be divided into two schools; *uncertainty* and *diversity*. The first aims to annotate samples for which the model has uncertain prediction *e.g.*, samples are selected

[5] $|\mathcal{L}_T| = 21 \times 6 = 126$ (1 shot) and $|\mathcal{L}_T| = 63 \times 6 = 3 \times 126$ (3 shot)

Tasks	**1-shot**			**3-shot**		
	MME	AADA	Sage	MME	AADA	Sage
R→C	67.5	64.4	**69.3**	70.1	68.8	**73.9**
R→P	**69.6**	65.5	69.4	70.8	67.0	**71.4**
P→C	69.0	63.2	**69.9**	71.4	67.3	**74.1**
C→S	**62.2**	57.4	61.5	64.7	60.1	**65.4**
S→P	**67.9**	62.6	**67.9**	69.6	64.9	**69.8**
R→S	61.2	57.0	**62.1**	63.6	59.9	**65.8**
P→R	**79.3**	74.9	79.0	80.9	76.9	**81.2**
Mean	68.1	63.6	**68.5**	70.2	66.3	**71.7**

Table 1. SSDA (MME) vs ADA (AADA and Sage) on DomainNet. MME's results deviate from [31] due to train/test split, ResNet50 as backbone and minor implementation changes.

according to their entropy [35] or prediction margin [28], with some theoretical guarantees [15,4]. The second focuses on annotating a representative sample of the data distribution *e.g.*, the Core-Set approach [32] selects samples that geometrically cover the distribution. Several approaches also propose a trade-off between uncertainty and diversity, *e.g.*, [17] that is formulated as a bandit problem. Recently, the work [2] introduces Badge, a gradient embedding, which, like SAGE, takes the best of uncertainty and diversity. Our work is inspired by Badge and adapts the core ideas in the context of ADA.

Active Domain Adaptation. Despite its great practical interest, only a few previous works address the problem of *Active Domain Adaptation.* [9] annotates target samples by importance sampling while [27,30] annotates samples with high discrepancy with source samples based on the prediction of a domain discriminator. However, those strategies do not fit modern adaptation with deep nets. To our knowledge, AADA [34] is the only prior work that learns actively domain invariant representations and achieves the state-of-the-art for Active Domain Adaptation. AADA is the most relevant work to compare with Sage.

7 Conclusion

We have introduced Sage, an efficient method for ADA which identifies target samples that are likely to improve representations' transferability when annotated. It relies on two core components; a stochastic embedding of the gradient of the transferability loss and a `k-means++` initialization, which guarantees that each annotation round annotates a diverse set of target samples. Through various experiments, we have demonstrated the effectiveness of Sage and its capacity to take the best of uncertainty, representative, and diversity sampling. New SSDA strategies when using Sage is an interesting direction for future works.

References

1. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Tech. rep., Stanford (2006)
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. arXiv preprint arXiv:1906.03671 (2019)
3. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)
4. Balcan, M.F., Beygelzimer, A., Langford, J.: Agnostic active learning. Journal of Computer and System Sciences **75**(1), 78–89 (2009)
5. Beery, S., Van Horn, G., Perona, P.: Recognition in terra incognita. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 456–473 (2018)
6. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in neural information processing systems. pp. 137–144 (2007)
7. Bouvier, V., Very, P., Chastagnol, C., Tami, M., Hudelot, C.: Robust domain adaptation: Representations, weights and inductive bias. ECML-PKDD (2020)
8. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
9. Chattopadhyay, R., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Joint transfer and batch-mode active learning. In: International Conference on Machine Learning. pp. 253–261 (2013)
10. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: International Conference on Machine Learning. pp. 1081–1090 (2019)
11. Tachet des Combes, R., Zhao, H., Wang, Y.X., Gordon, G.J.: Domain adaptation with conditional distribution matching and generalized label shift. Advances in Neural Information Processing Systems **33** (2020)
12. Corbière, C., THOME, N., Bar-Hen, A., Cord, M., Pérez, P.: Addressing failure prediction by learning model confidence. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 2902–2913. Curran Associates, Inc. (2019)
13. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning. pp. 1180–1189 (2015)
14. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in neural information processing systems. pp. 529–536 (2005)
15. Hanneke, S., et al.: Theory of disagreement-based active learning. Foundations and Trends® in Machine Learning **7**(2-3), 131–309 (2014)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hsu, W.N., Lin, H.T.: Active learning by learning. In: Twenty-Ninth AAAI conference on artificial intelligence. Citeseer (2015)
18. Johansson, F., Sontag, D., Ranganath, R.: Support and invertibility in domain-invariant representations. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 527–536 (2019)

19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
20. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: A general approach to adapting deep classifiers. In: International Conference on Machine Learning. pp. 4013–4022 (2019)
21. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37. pp. 97–105. JMLR. org (2015)
22. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1640–1650 (2018)
23. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
24. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1406–1415 (2019)
25. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
26. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. The MIT Press (2009)
27. Rai, P., Saha, A., Daumé III, H., Venkatasubramanian, S.: Domain adaptation meets active learning. In: Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing. pp. 27–32. Association for Computational Linguistics (2010)
28. Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: European Conference on Machine Learning. pp. 413–424. Springer (2006)
29. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European conference on computer vision. pp. 213–226. Springer (2010)
30. Saha, A., Rai, P., Daumé, H., Venkatasubramanian, S., DuVall, S.L.: Active supervised domain adaptation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 97–112. Springer (2011)
31. Saito, K., Kim, D., Sclaroff, S., Darrell, T., Saenko, K.: Semi-supervised domain adaptation via minimax entropy. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8050–8058 (2019)
32. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: ICLR (2018)
33. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
34. Su, J.C., Tsai, Y.H., Sohn, K., Liu, B., Maji, S., Chandraker, M.: Active adversarial domain adaptation. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 739–748 (2020)
35. Wang, D., Shang, Y.: A new active labeling method for deep learning. In: 2014 International joint conference on neural networks (IJCNN). pp. 112–119. IEEE (2014)
36. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)

37. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2720–2729 (2019)
38. Zhao, H., Des Combes, R.T., Zhang, K., Gordon, G.: On learning invariant representations for domain adaptation. In: International Conference on Machine Learning. pp. 7523–7532 (2019)