Towards Accurate Model Selection in Deep Unsupervised Domain Adaptation

Kaichao You [1 2] **Ximei Wang** [1 2] **Mingsheng Long** [1 2] **Michael I. Jordan** [3]

Abstract

Deep unsupervised domain adaptation (Deep UDA) methods successfully leverage rich labeled data in a source domain to boost the performance on related but unlabeled data in a target domain. However, algorithm comparison is cumbersome in Deep UDA due to the absence of accurate and standardized model selection method, posing an obstacle to further advances in the field. Existing model selection methods for Deep UDA are either highly biased, restricted, unstable, or even controversial (requiring labeled target data). To this end, we propose *Deep Embedded Validation* (**DEV**), which embeds adapted feature representation into the validation procedure to obtain unbiased estimation of the target risk with bounded variance. The variance is further reduced by the technique of control variate. The efficacy of the method has been justified both theoretically and empirically.

1. Introduction

Deep learning enables machine recognition (He et al., 2016; Long et al., 2015a) at the cost of large scale labeled data. It is common to trade off the limited labeling budget against the demand for more labeled data by data-hungry deep models. Domain adaptation (Pan & Yang, 2010) serves as a promising solution to such a dilemma: it transfers the knowledge from existing labeled data (*source domain*) to the unlabeled data (*target domain*) to reduce the labeling work.

The formulations of domain adaptation mainly fall into two categories, *covariate shift* and *label shift*, relating to causal and anti-causal inference (Schölkopf et al., 2012). Although some works focus on the *label shift* (Lipton et al., 2018; Azizzadenesheli et al., 2019), *covariate shift* appears more natural in recognition tasks where deep models have shown their superiority (Long et al., 2015b; Ganin et al., 2016).

While shallow learning methods have been extensively studied to tackle domain adaptation problems (Gong et al., 2012; Fernando et al., 2013), deep models (Long et al., 2015b; Ganin et al., 2016; Saito et al., 2018a) are attracting more and more attention because of their impressive performance. Distribution matching methods (Long et al., 2017; Shen et al., 2018) align domains with well-defined statistical distribution divergence between deep features, while adversarial learning methods (Ganin et al., 2016; Tzeng et al., 2017) learn domain-invariant deep representations with adversarial training (Goodfellow et al., 2014). Other approaches to deep domain adaptation include generative models (Sankaranarayanan et al., 2018), similarity learning (Pinheiro, 2018), to name a few.

The typical pipeline of machine learning is as follows: several hyperparameter configurations are tried to get candidate models, then model selection methods are employed to select the best configuration, and the final result in the test set is reported. Such a diagram struggles in domain adaptation: the performance in the test set (*Target Risk*) is what we care about, but labels for the test set are inaccessible both in the model training and selection stage. Labels in the training set are abundant, but the performance in the training set (*Source Risk*) is inconsistent to the target risk because of the domain shift, posing an obstacle to model selection in domain adaptation. Contemporarily there are several plausible hyperparameter selection methods:

(1) Fixed Hyperparameters. Tzeng et al. (2017); Saito et al. (2018a); Pinheiro (2018) stated that hyperparameters are fixed across various experiments on the same dataset. However, it remains unclear how these fixed hyperparameters were selected. A reasonable hypothesis would be that hyperparameters are selected in one task with target labeled data and applied across other tasks, which requires more than one task at hand. Such a requirement may be satisfied in the research area where there are several tasks in each dataset. Nonetheless, in practical domain adaptation scenarios, we are often interested in one task, and Fixed Hyperparameters strategy will not work.

(2) Source Risk. Ganin et al. (2016) selected hyperparameters by taking the source risk into account. Although the

[1]School of Software [2]BNRist, Research Center for Big Data, Tsinghua University, Beijing, China [3]University of California, Berkeley, USA.

Kaichao You <youkaichao@gmail.com>. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

Table 1. Comparisons among different model selection methods for Deep UDA.

Method	Working Assumptions		Technical Advantages	
	covariate shift	w/o target labels	unbiased	controlled variance
Source Risk	✗	✓	✗	✗
Target Risk	✓	✗	✓	✓
IWCV (Sugiyama et al., 2007)	✓	✓	✓	✗
TrCV (Zhong et al., 2010)	✓	✗	✓	✗
DEV (Proposed)	✓	✓	✓	✓

source domain is related to the target domain and source risk may reflect the target risk to some extent, such a method comes without theoretical guarantees and is not convincing. Specifically, source risk is a highly biased estimator of the underlying target risk in the presence of a large domain gap.

(3) Target Risk. Hoffman et al. (2018) leaved a proportion of target data to be held out for model selection and the rest target data for transductive training. While it is an unbiased estimator of ground truth target risk, it is controversial to employ labeled target data in unsupervised domain adaptation. If some labeled target data are available, then instead of using them for model selection, why not exploring them for semi-supervised domain adaptation that often yields better empirical results?

(4) Importance-Weighted Cross-Validation (IWCV) (Sugiyama et al., 2007)**.** Initially designed as a validation method for the covariate shift problem, IWCV was adopted by (Long et al., 2018) to tune hyperparameters. While it has a theoretical guarantee that it is unbiased, IWCV requires known density ratio to approximate the target distribution. If no density ratio is given, it fits a multi-dimensional normal distribution to estimate the density ratio. Furthermore, the variance of IWCV is unbounded, explaining its instability.

A detailed comparison of these model selection methods is presented in Table 1. Different domain adaptation algorithms employ different model selection methods. It is thus challenging to compare state of the art Deep UDA models if they are selected by different validation methods. For example, it is unfair to compare the performance selected by Target Risk with that selected by Source Risk. For a fair comparison, researchers may struggle to re-implement existing methods under the same validation scheme. Further, the same work may exhibit very different results due to inconsistent model selection methods in a variety of publications.

Dilemma in VisDA Challenge. Synthetic-to-Real Visual Domain Adaptation (VisDA) (Peng et al., 2018) is a large scale domain adaptation challenge. It aims to facilitate the development of unsupervised domain adaptation and provides the largest cross-domain object classification dataset to date. When it comes to model selection, however, the organizers get trapped into the dilemma: the labels of test set cannot be published and can only live in the test server; the labels of training set are given, but presented with the considerable domain gap, their effect for model selection is limited. As a compromise, the organizers released a fully labeled validation set which is different from both the source domain and target domain, only used for model selection.

To combat the above dilemma in Deep UDA, we propose *Deep Embedded Validation* (**DEV**), a new model selection method tailored to Deep UDA. It embeds adapted feature representation in the validation procedure to yield unbiased estimation of the target risk with bounded variance. Control variate method is exploited to further reduce the variance of the estimation. Theoretical analysis shows the advantage of DEV. Furthermore, several empirical experiments show the performance of models selected by DEV approaches that of Target Risk, even though no target labeled data are required.

2. Related Work

In this section, we review the model selection methods in both supervised learning and domain adaptation.

2.1. Model Selection in Single Domain

Machine learning algorithms aim to learn knowledge from data. While learning is carried out using the *training set*, our interest mainly focuses on the performance of the algorithm on unseen data, which gives rise to the usage of the *test set*. Test data is held out in the training stage. It is tested only once. Hyperparameters of machine learning algorithms are selected on another part of data dubbed as the *validation set*.

Hold-out. If abundant data is given, we can just split the data into three parts. The *training set* is only for learning, the *validation set* is only for hyperparameter tuning, and the *test set* is for the final evaluation. We call such a method hold-out because the *validation set* is not involved in training. In hold-out validation, each candidate model will only be evaluated once.

***k*-fold Cross-Validation**. In the case of limited data, however, one would strive to involve as much data in training as

possible. k-fold Cross-Validation (Kohavi, 1995) splits the given data into k folds, runs the algorithm using $\frac{k-1}{k}$ data, validates it using the rest $\frac{1}{k}$ data and then repeats k times, averaging the results. k-fold Cross-Validation exploits all the available data at the cost of k runs. **Leave-one-out** is one particular case of k-fold Cross-Validation with k equaled to the number of training data. It can be applied in the extreme circumstance where the labeled data is particularly scarce.

Although learning in a single domain is well defined, the assumption that the test distribution is the same as the training distribution is often violated in real-world applications. For example, clinical data is collected from patients while the algorithm developed from the data would be tested among ordinary people. Models learned from patients cannot apply to normal people because of the dataset bias. Thus it would be of great significance if models can learn from different domains. Still, it is difficult to formalize learning from two arbitrary domains. A well-defined paradigm is to learn from two domains under the *covariate shift* assumption (Blitzer et al., 2008; Saenko et al., 2010; Ben-David et al., 2010).

2.2. Model Selection in Domain Adaptation

Importance-Weighted Cross-Validation. When encountering covariate shift problems, the model selection methods for learning in a single domain would fail to identify the best model for the target domain. The estimation in both the *training set* and the *validation set* would be biased and cannot reflect the test risk. Sugiyama et al. (2007) proposed Importance-Weighted Cross-Validation (IWCV) to perform cross-validation under covariate shift with the aid of known density ratio. The validation risk is weighted to be an unbiased estimator of the target risk. Though unbiased, such an estimator has unbounded variance. Moreover, when the density ratio is not readily available, IWCV needs to estimate the density ratio by a single multi-dimensional normal distribution, which is cumbersome and inaccurate.

Later, Cortes et al. (2010) revealed that the variance of the importance-weighted methods can be bounded by a family of Rényi divergence (Rényi, 1961). While the variance is bounded by the Rényi divergence, neither the variance itself nor the bound of the variance is lowered.

Transfer Cross-Validation. By considering both marginal and conditional distributions in different domains, Zhong et al. (2010) proposed Transfer Cross-Validation (TrCV) for model selection under both marginal and conditional shifts. To approximate the conditional distribution, TrCV requires labeled target data to assist the model selection process. TrCV incurs similar controversy as tuning hyperparameters by Target Risk. In Deep UDA, one would prefer a model selection method that works without target labeled data but still correlates well to Target Risk with statistical guarantees.

The mentioned model selection methods both originated in the era before deep learning. They both work in the *covariate* level, i.e., they validate models directly based on the input data. In the age of deep learning, it is natural to extend these models to work in the *feature* level. While intuitive, the theoretical property of such an "embedded" validation remains unknown. We are the first to validate models on the feature level and provide the theoretical insight behind the embedded validation. The validation yields a tighter bound on its variance and remains unbiased. The variance can be further lowered by employing a control variate method.

3. Preliminaries

3.1. Rényi Divergence

We first introduce the notation of Rényi divergence defined in (Rényi, 1961; Cortes et al., 2010). Rényi divergence between distributions p and q is defined as $D_\alpha(p\|q) = \frac{1}{\alpha-1}\log_2\sum_x p(x)\left(\frac{p(x)}{q(x)}\right)^{\alpha-1}$, where hyperparameter $\alpha \geq 0$ and $\alpha \neq 1$. Note that $\lim_{\alpha\to 1} D_\alpha(p\|q) = KL(p\|q)$, which is the widely-used Kullback-Leibler divergence. Rényi divergence satisfies the properties of a well-defined divergence: it is non-negative and $D_\alpha(p\|q) = 0$ if and only if $p = q$. For brevity, another notation of Rényi divergence is adopted:

$$d_\alpha(p\|q) = 2^{D_\alpha(p\|q)} = \left[\sum_x \frac{p^\alpha(x)}{q^{\alpha-1}(x)}\right]^{\frac{1}{\alpha-1}}. \quad (1)$$

3.2. Control Variates

The *control variates* method (Lemieux, 2017), widely used in Monte Carlo methods, is an effective technique to reduce variance. Suppose the statistic z is an unbiased estimator of an unknown parameter ζ, i.e. $\mathbb{E}[z] = \zeta$. However, an unbiased estimator would never be accurate if its variance $\mathrm{Var}[z]$ is high. To reduce its variance, we can find another related unbiased estimator t such that $\mathbb{E}[t] = \tau$, where τ is the parameter that t tries to estimate. Then we can construct a new estimator parameterized by a constant η,

$$z^\star = z + \eta(t - \tau). \quad (2)$$

It is straightforward to show that $z^\star$ is still unbiased thanks to the linear property of the expectation operation:

$$\mathbb{E}[z^\star] = \mathbb{E}[z] + \eta\mathbb{E}[t - \tau] = \zeta + \eta(\mathbb{E}[t] - \mathbb{E}[\tau]) = \zeta.$$

The variance of $z^\star$ can be computed as

$$\begin{aligned}\mathrm{Var}[z^\star] &= \mathrm{Var}[z + \eta(t - \tau)] \\ &= \eta^2\mathrm{Var}[t] + 2\eta\mathrm{Cov}(z, t) + \mathrm{Var}[z],\end{aligned} \quad (3)$$

which is a quadratic form of η and have a global optimum

$$\min \mathrm{Var}[z^\star] = (1 - \rho_{z,t}^2)\mathrm{Var}[z], \text{ when } \hat{\eta} = -\frac{\mathrm{Cov}(z, t)}{\mathrm{Var}[t]}, \quad (4)$$

where $\rho_{z,t}$ is the correlation coefficient of z and t. Since $0 \leq |\rho_{z,t}| \leq 1$, the variance is reduced: $\mathrm{Var}[z^\star] \leq \mathrm{Var}[z]$.

In essence, the control variate method finds a correlated and unbiased variable and subtracts it with a proper coefficient, thus making the estimator deviate less from the expectation.

4. Method

In Deep UDA, we learn towards a joint distribution $J(\mathbf{x}, d)$, where $\mathbf{x}$ is the input associated with the label y, and d is a Bernoulli variable indicating the domain that $\mathbf{x}$ belongs to. Let $p(\mathbf{x}) = J(\mathbf{x}, d|d = 1)$ denote the *source domain* distribution and $q(\mathbf{x}) = J(\mathbf{x}, d|d = 0)$ denote the *target domain* distribution. The domain shift $p(\mathbf{x}) \neq q(\mathbf{x})$ presents a major challenge for domain adaptation. Meanwhile, *covariate shift* is a common assumption that says $p(y|\mathbf{x}) = q(y|\mathbf{x})$, i.e. the class label of the input is independent of the domain. Samples drawn *i.i.d* from the joint distribution $J(\mathbf{x}, d)$ form our dataset: *source domain* observations $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ with $d = 1$, *target domain* observations $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$ with $d = 0$ (y_i^t is not accessible in the training phase).

Model selection is to find $\hat{g} = \arg\min_{g \in G} \mathbb{E}_{\mathbf{x}\sim q} \ell(g(\mathbf{x}), y)$, where G is the model space where each model maps the input $\mathbf{x}$ to the output $\hat{y}$, and $\ell(\cdot, \cdot)$ is the loss function. In reality, though, it is infeasible to search through the whole model space, and we opt to find $\hat{g} = \arg\min_{g \in G_m} \mathbb{E}_{\mathbf{x}\sim q} \ell(g(\mathbf{x}), y)$, where $G_m = \{g_i\}_{i=1}^m$ is a finite set of candidate models.

The difficulty for model selection in Deep UDA arises from the fact that y is inaccessible when $\mathbf{x} \sim q$. Hence, we have to estimate $\mathbb{E}_{\mathbf{x}\sim q} \ell(g(\mathbf{x}), y)$ with the help of labeled source data $\mathbf{x} \sim p$. Considering that deep learning models usually learn a discriminative feature representation and then perform downstream tasks, we split g into two functions: $g(\mathbf{x}) = \mathcal{T}(\boldsymbol{f})$, where $\boldsymbol{f} = F(\mathbf{x})$. Here F is the feature extractor and $\mathcal{T}$ takes the feature $\boldsymbol{f}$ to perform specific tasks.

4.1. Importance-Weighted Cross-Validation

The main challenge for model selection in Deep UDA is that the target risk $\mathcal{R}(g) = \mathbb{E}_{\mathbf{x}\sim q} \ell(g(\mathbf{x}), y)$ is defined over the target domain distribution q without any labeled data. If density ratio (a.k.a. importance weights) $w(\mathbf{x}) = \frac{q(\mathbf{x})}{p(\mathbf{x})}$ is known, following Sugiyama et al. (2007), we can obtain

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim p} w(\mathbf{x}) \ell(g(\mathbf{x}), y) &= \mathbb{E}_{\mathbf{x}\sim p} \frac{q(\mathbf{x})}{p(\mathbf{x})} \ell(g(\mathbf{x}), y) \\
&= \int_p \frac{q(\mathbf{x})}{p(\mathbf{x})} \ell(g(\mathbf{x}), y) p(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \int_q \ell(g(\mathbf{x}), y) q(\mathbf{x}) \mathrm{d}\mathbf{x} \\
&= \mathbb{E}_{\mathbf{x}\sim q} \ell(g(\mathbf{x}), y) \\
&= \mathcal{R}(g),
\end{aligned} \tag{5}$$

which means $\frac{1}{n_s} \Sigma_{i=1}^{n_s} (w(\mathbf{x}_i^s) \ell(g(\mathbf{x}_i^s), y_i^s))$ is an *unbiased* estimator of the target risk $\mathcal{R}(g)$.

For brevity, we denote $w(\mathbf{x})\ell(g(\mathbf{x}), y)$ by ℓ_w. As shown in Cortes et al. (2010) (Lemma 2), the variance of importance-weighted cross-validation is bounded by Rényi divergence:

$$\begin{aligned}
\mathrm{Var}_{\mathbf{x}\sim p}[\ell_w] &= \mathbb{E}_{\mathbf{x}\sim p}[(\ell_w)^2] - (\mathbb{E}_{\mathbf{x}\sim p}[\ell_w])^2 \\
&\leq d_{\alpha+1}(q\|p) \mathcal{R}(g)^{1-\frac{1}{\alpha}} - \mathcal{R}(g)^2.
\end{aligned} \tag{6}$$

4.2. Deep Embedded Validation

As shown in Eq. (6), the variance of importance-weighted cross-validation (IWCV) is bounded by Rényi divergence between distributions p and q. However, neither the variance of IWCV nor its bound is lowered as domain adaptation goes on, given that p and q stay still. Recent work (Long et al., 2018) (Figure 2(c)) shows that the distribution divergence becomes smaller after feature adaptation. While p and q stay still in the input space, better adaptation model tends to show lower distribution divergence in the feature space. Note that these models can only reduce the distribution divergence rather than closing it, implying that it is still necessary to develop a validation method for Deep UDA.

These observations inspire us to step from the *covariate* space to the *feature* space. Let $p_{\boldsymbol{f}}$ and $q_{\boldsymbol{f}}$ be the feature distributions of the source domain and the target domain respectively. Deep domain adaptation models usually close the domain gap by learning domain-invariant features, which implies $d_{\alpha+1}(q_{\boldsymbol{f}}\|p_{\boldsymbol{f}})$ is generally smaller than $d_{\alpha+1}(q\|p)$. Thereby, we propose to embed the learned deep features into the validation procedure, resulting in an embedded density ratio estimation $w_{\boldsymbol{f}}(\mathbf{x}) = \frac{q_{\boldsymbol{f}}(\mathbf{x})}{p_{\boldsymbol{f}}(\mathbf{x})}$. By changing from the *covariate* level to the *feature* level, we can also conclude that $\frac{1}{n_s} \Sigma_{i=1}^{i=n_s} (w_{\boldsymbol{f}}(\mathbf{x}_i^s) \ell(g(\mathbf{x}_i^s), y_i^s))$ is an unbiased estimator of the target risk, with its variance $\mathrm{Var}_{\mathbf{x}\sim p_{\boldsymbol{f}}}[\ell_{w_{\boldsymbol{f}}}]$ bounded by $d_{\alpha+1}(q_{\boldsymbol{f}}\|p_{\boldsymbol{f}}) \mathcal{R}(g)^{1-\frac{1}{\alpha}} - \mathcal{R}(g)^2$, which is generally smaller than $d_{\alpha+1}(q\|p) \mathcal{R}(g)^{1-\frac{1}{\alpha}} - \mathcal{R}(g)^2$.

As can be verified in Eq. (5), IWCV requires an important assumption that the support of p contains the support of q, i.e. $\mathbf{supp}(p) \supset \mathbf{supp}(q)$, where $\mathbf{supp}(p) = \{\mathbf{x}|p(\mathbf{x}) \neq 0\}$. If the assumption is violated, the importance weights can grow to infinity. Before aligning distributions p and q, it is highly possible that the assumption is violated, especially image and text data with high-dimensional input covariates. After feature learning and adaptation, the deep features are made more compact and domain-invariant. The assumption on the support of q in p can hold well in the learned feature space.

4.3. Discriminative Density Ratio Estimation

Density ratio is not readily accessible in pragmatic applications. Here we adopt an approach similar in (Bickel et al., 2007), using Bayesian formula to derive density ratio from a

model that discriminates between source and target samples:

$$\begin{aligned} w_{\boldsymbol{f}}(\mathbf{x}) &= \frac{q_{\boldsymbol{f}}(\mathbf{x})}{p_{\boldsymbol{f}}(\mathbf{x})} = \frac{J_{\boldsymbol{f}}(\mathbf{x}|d=0)}{J_{\boldsymbol{f}}(\mathbf{x}|d=1)} \\ &= \frac{J_{\boldsymbol{f}}(d=1)}{J_{\boldsymbol{f}}(d=0)} \frac{J_{\boldsymbol{f}}(\mathbf{x}) J_{\boldsymbol{f}}(d=0|\mathbf{x})}{J_{\boldsymbol{f}}(\mathbf{x}) J_{\boldsymbol{f}}(d=1|\mathbf{x})} \\ &= \frac{J_{\boldsymbol{f}}(d=1)}{J_{\boldsymbol{f}}(d=0)} \frac{J_{\boldsymbol{f}}(d=0|\mathbf{x})}{J_{\boldsymbol{f}}(d=1|\mathbf{x})} \\ &= \frac{n_s}{n_t} \frac{J_{\boldsymbol{f}}(d=0|\mathbf{x})}{J_{\boldsymbol{f}}(d=1|\mathbf{x})}, \end{aligned} \tag{7}$$

where $J_{\boldsymbol{f}}$ is the joint distribution in the feature space.

As can be seen in Eq. (7), density ratio can be decomposed into two parts: $\frac{J_{\boldsymbol{f}}(d=0|\mathbf{x})}{J_{\boldsymbol{f}}(d=1|\mathbf{x})}$ and a constant factor $\frac{J_{\boldsymbol{f}}(d=1)}{J_{\boldsymbol{f}}(d=0)}$. The former can be estimated by a discriminative model to distinguish source examples from target examples. The latter constant factor does not vary with models and can be estimated with the sample sizes of both domains. Note that the discriminative model is trained on the readily-available domain information d, which follows a fully supervised learning scheme. A two-layer logistic regression model will be accurate enough for estimating the density ratio.

4.4. Variance Reduction by Control Variate

In Section 4.2, it is clear that using embedded deep feature representations for model selection will benefit from a lower bound on the variance of the target risk estimation. Meanwhile, we can explicitly reduce the variance by adopting the *control variate* method described in Section 3.2.

To reduce the variance $\mathrm{Var}_{\mathbf{x}\sim p_{\boldsymbol{f}}}[\ell_{w_{\boldsymbol{f}}}]$ of target risk estimate, there are two candidates for the control variate: $w_{\boldsymbol{f}}$ and ℓ. However, the expectation of ℓ is not only unknown but also task-specific, depending on the choice of $\ell(\cdot,\cdot)$. By contrast, the expectation of $w_{\boldsymbol{f}}$ remains independent of the model $\mathcal{T}$,

$$\begin{aligned} \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{f}}} w_{\boldsymbol{f}}(\mathbf{x}) &= \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{f}}} \frac{q_{\boldsymbol{f}}(\mathbf{x})}{p_{\boldsymbol{f}}(\mathbf{x})} \\ &= \int \frac{q_{\boldsymbol{f}}(\mathbf{x})}{p_{\boldsymbol{f}}(\mathbf{x})} p_{\boldsymbol{f}}(\mathbf{x}) \mathrm{d}\mathbf{x} = 1. \end{aligned} \tag{8}$$

By plugging $w_{\boldsymbol{f}}$ into Eq. (2) as a control variate for the target risk estimation, we can embed features $\boldsymbol{f}$ in the estimator as

$$\begin{aligned} \mathcal{R}_{\mathrm{DEV}} &= \frac{1}{n_s} \sum_{i=1}^{n_s} w_{\boldsymbol{f}}(\mathbf{x}_i^s) \ell(g(\mathbf{x}_i^s), y_i^s) \\ &+ \frac{\eta}{n_s} \sum_{i=1}^{n_s} [w_{\boldsymbol{f}}(\mathbf{x}_i^s) - \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{f}}} w_{\boldsymbol{f}}(\mathbf{x})] \\ &= \frac{1}{n_s} \sum_{i=1}^{n_s} [\ell_{w_{\boldsymbol{f}}} + \eta(w_{\boldsymbol{f}}(\mathbf{x}_i^s) - 1)], \end{aligned} \tag{9}$$

where η is the optimal coefficient estimated by

$$\eta = -\frac{\widehat{\mathrm{Cov}}(\ell_{w_{\boldsymbol{f}}}, w_{\boldsymbol{f}})}{\widehat{\mathrm{Var}}[w_{\boldsymbol{f}}]}. \tag{10}$$

The complete validation procedure, which is called *Deep Embedded Validation* (**DEV**), is described in Algorithms 1 and 2. DEV is tailored to Deep UDA models by embedding adapted deep feature representations into model selection. It is an unbiased estimation of the target risk while its variance is bounded by a theoretical guarantee and is further reduced by the control variate method. Note that the control variate method can be applied not only on deep models but also on shallow models. But deep models benefit more by having smaller Rényi divergence, thus having a lower upper bound.

Algorithm 1 GetRisk

Input: Candidate model $g(\mathbf{x}) = \mathcal{T}(F(\mathbf{x}))$
Training set $\mathcal{D}_{\mathrm{tr}} = \{(\mathbf{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$
Validation set $\mathcal{D}_{\mathrm{v}} = \{(\mathbf{x}_i^{\mathrm{v}}, y_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Test set $\mathcal{D}_{\mathrm{ts}} = \{(\mathbf{x}_i^{\mathrm{ts}})\}_{i=1}^{n_{\mathrm{ts}}}$
$\mathcal{D}_s$ is partitioned into $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{v}}$
Output: DEV Risk $\mathcal{R}_{\mathrm{DEV}}(g)$ of model g

Compute features and predictions using model g:
$\mathcal{F}_{\mathrm{tr}} = \{\boldsymbol{f}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, $\mathcal{F}_{\mathrm{ts}} = \{\boldsymbol{f}_i^{\mathrm{ts}}\}_{i=1}^{n_{\mathrm{ts}}}$
$\mathcal{F}_{\mathrm{v}} = \{\boldsymbol{f}_i^{\mathrm{v}}\}_{i=1}^{n_{\mathrm{v}}}$, $\mathcal{Y}_{\mathrm{v}} = \{\hat{y}_i^{\mathrm{v}}\}_{i=1}^{n_{\mathrm{v}}}$
Train a two-layer logistic regression model M to classify $\mathcal{F}_{\mathrm{tr}}$ and $\mathcal{F}_{\mathrm{ts}}$ (label $\mathcal{F}_{\mathrm{tr}}$ as 1 and $\mathcal{F}_{\mathrm{ts}}$ as 0)
Compute $w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}}) = \frac{n_{\mathrm{tr}}}{n_{\mathrm{ts}}} \frac{1-M(\boldsymbol{f}_i^{\mathrm{v}})}{M(\boldsymbol{f}_i^{\mathrm{v}})}$, $W = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Compute weighted loss $L = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\mathrm{v}}) \ell(\hat{y}_i^{\mathrm{v}}, y_i^{\mathrm{v}})\}_{i=1}^{n_{\mathrm{v}}}$
Estimate coefficient $\eta = -\frac{\widehat{\mathrm{Cov}}(L,W)}{\widehat{\mathrm{Var}}[W]}$
Compute DEV Risk:
$$\mathcal{R}_{\mathrm{DEV}}(g) = \mathrm{mean}(L) + \eta \mathrm{mean}(W) - \eta$$

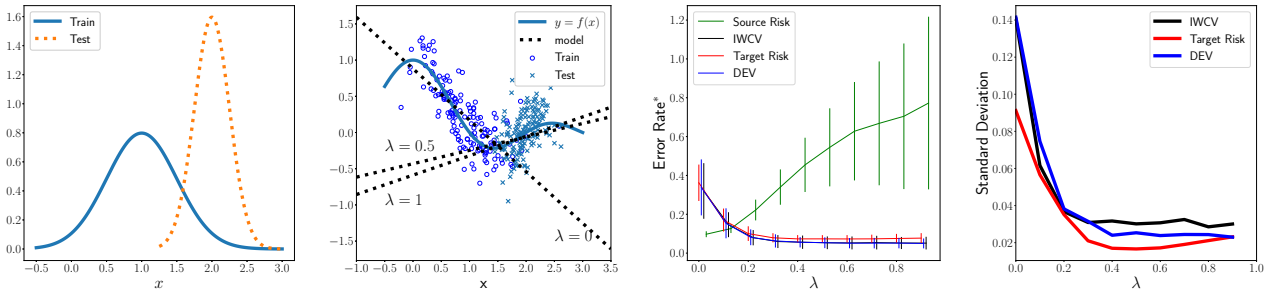
Algorithm 2 Deep Embedded Validation (DEV)

Input: A set of candidate models $G_m = \{g_i(\mathbf{x})\}_{i=1}^m$
Output: The best model $(G_m)_{\hat{i}}$

Get DEV Risks of all models $\mathcal{R} = \{\mathrm{GetRisk}(g_i)\}_{i=1}^m$
Rank the best model $\hat{i} = \arg\min_{1\le i\le m} \mathcal{R}_i$

4.5. Beyond Feature Adaptation Methods

The above discussion focuses on feature adaptation models for Deep UDA. These models learn features to reduce the distribution shift, which also reduces the variance of target risk estimation. However, DEV is not limited to the feature adaptation models. For example, generative models that generate an auxiliary domain (Murez et al., 2018) can be formalized as generating another distribution $\hat{p}$ that is closer to q. In such a scenario, samples generated by $\hat{p}$ naturally have labels, and DEV can be carried out between $\hat{p}$ and q.



(a) (b) (c) (d)

Figure 1. An example of model selection in a toy dataset for regression. (a) Input data density with covariate shift. (b) Dataset and the underlying objective function as well as several candidate models with different hyperparameters λ. (c) Comparison among different validation methods. (d) The standard deviation in details. (*) The error rate is used to select models. It is not the final reported accuracy.

5. Experiments

In this section, we conduct a series of experiments to empirically evaluate the proposed DEV approach. We first play around with a toy dataset and then dive into Deep UDA models. With deep models, we try the following learning rates: $\{10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}, 10^{-4}\}$. Other hyperparameters are specified in their subsections respectively. Model selection is conducted based on different methods: (**1**) using source error to select models (**Source Risk**); (**2**) using target error to select models (**Target Risk**); (**3**) **IWCV** (Sugiyama et al., 2007), with importance weights generated based on pre-trained feature representations; (**4**) **DEV**. The code of DEV is available at `https://github.com/thuml/Deep-Embedded-Validation`.

There are several clarifications. (**1**) Sugiyama et al. (2007) fits a multi-variate Gaussian to approximate the density ratio. Gaussians always underfit deep features and yield extremely unstable weights. Thus we give it the benefit to estimate the density ratio by using pre-trained deep features to train a binary classifier as described in Section 4.3. (**2**) In Section 5.4, there is no standard model for feature extraction on Digits dataset, so IWCV is not reported. For other datasets, the features used by IWCV are extracted by ResNet-50 (He et al., 2016). (**3**) For a fair comparison, we also list the results reported in their original papers (denoted as Original).

As mentioned in Section 2.1, cross-validation is needed when labeled data is scarce. In Deep UDA, while we do not have labeled data in the target domain, a large number of labeled data in the source domain are available. Since DEV is carried out on the source data, we can split the source data into train/validation set before learning. That said, we use the hold-out validation method throughout all experiments.

5.1. Toy Dataset

Figures 1(a) and 1(b) show a toy regression data following the protocol of Sugiyama et al. (2007). Data points lie on $y = \frac{\sin(\pi x)}{\pi x}$ with random noise sampled from normal distribution $\mathcal{N}(0, (\frac{1}{4})^2)$. The marginal distribution of x differs in the training set and test set (Figure 1(a)), explaining the *covariate shift* problem. Here $p(x) = \mathcal{N}(x|1, (\frac{1}{2})^2), q(x) = \mathcal{N}(x|2, (\frac{1}{4})^2)$. The density ratio $w(x)$ can be computed analytically. Candidate models for the toy problem are AIWLS models (Sugiyama et al., 2007) with different hyperparameters λ ranging from 0.0 to 1.0. As can be seen in Figure 1(b), when λ gets larger, the AIWLS model fits the test set better.

We ran $1,000$ experiments to compute the risk estimated by different methods. The mean and standard deviation of the estimation are plotted in Figure 1(c). Source Risk tends to deviate from Target Risk and is not a reasonable estimator. DEV and IWCV both correspond well with Target Risk, but after a closer look at Figure 1(d), we observe that DEV shows significantly smaller variance compared to IWCV, which justifies the efficacy of the control variate method.

5.2. VisDA Dataset

VisDA (Peng et al., 2018) is a large-scale cross-domain dataset designed for domain adaptation in computer vision. The source domain in VisDA consists of synthetic rendered images, while the target domain images are cropped from either Microsoft COCO dataset (Lin et al., 2014) or Youtube Bounding Boxes dataset (Real et al., 2017).

We choose a state of the art model on the VisDA dataset, Maximum Classifier Discrepancy (**MCD**) (Saito et al., 2018a), to explore the model selection efficacy of DEV. A key hyperparameter in MCD is *the number of generator update iterations*, denoted as k. We try $k = 1, 2, 3, 4, 5$ together with various learning rates to select the best model.

Note that we tune MCD in terms of its mean accuracy, not by tuning each class and aggregating the best results. So the accuracy of Target Risk is only an upper bound with respect to the mean accuracy. According to Table 2, IWCV does not work well on VisDA. IWCV relies on pre-trained features to

Table 2. Accuracy (%) of **MCD** (Saito et al., 2018a) by different validation methods on VisDA dataset.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	mean
Original (Saito et al., 2018a)	87.00	60.90	83.70	64.00	88.90	79.60	84.70	76.90	88.60	40.30	83.00	25.80	71.90
Source Risk	84.39	54.11	69.15	46.37	80.49	80.45	85.04	65.24	87.22	36.86	78.04	28.91	66.36
IWCV	80.80	48.26	68.89	69.89	59.62	37.49	46.51	23.83	75.25	46.56	83.43	16.08	54.72
DEV (w/o control variate)	84.21	63.95	79.00	59.53	85.83	75.06	87.05	71.65	89.85	47.37	72.29	26.81	70.22
DEV	81.83	53.48	82.95	71.62	89.16	72.03	89.36	75.73	97.02	55.48	71.19	29.17	72.42
Target Risk (Upper Bound)	81.95	53.60	83.07	72.02	89.25	72.15	89.55	75.83	97.10	55.57	71.19	29.27	72.55

Table 3. Accuracy (%) of **CDAN** (Long et al., 2018) by different validation methods on Office-31 dataset.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
Original (Long et al., 2018)	93.10	98.60	100.00	92.90	71.00	69.30	87.50
Source Risk	85.95	98.60	100.00	84.59	65.00	61.34	82.58
IWCV	88.95	95.30	97.00	87.59	65.37	67.95	83.69
DEV	93.23	98.40	100.00	92.81	70.89	71.15	87.75
Target Risk (Upper Bound)	93.33	100.00	100.00	93.06	71.10	71.45	88.16

compute the density ratio estimation, but the domain gap in pre-trained features is large, leading to unstable importance weights and degraded performance of IWCV. In contrast, DEV successfully selects out the best model by working on the adapted features which endow smaller domain gap than the pre-trained features. According to the importance weighting theory in terms of Rényi divergence (Cortes et al., 2010), this implies a bounded variance of DEV.

5.3. Office-31 Dataset

Office-31 (Saenko et al., 2010) is a standard dataset for visual domain adaptation. With 4652 images in total, it is divided into 3 domains: images downloaded from **A**mazon, photos taken by **D**SLR and **W**eb camera. Since photos from W and D contain the same objects, they are visually similar and present small domain gap. Images from A are usually dissimilar with images from W and D.

We select another state of the art Deep UDA model on the Office-31 dataset, Conditional Domain Adversarial Network (**CDAN**) (Long et al., 2018) for evaluating model selection performance of different validation methods. In CDAN, an important hyperparameter is the trade-off coefficient λ, which balances between the transferability and the discriminability of the learned representations. We implement several trade-offs ($\lambda = 0.5, 0.75, 1.0, 1.25, 1.5$, with $\lambda = 1$ as its default setting) along with several learning rate configurations. Results are reported in Table 3. Performance tuned by Target Risk is the upper bound and we are glad to observe that DEV performs nearly as well as Target Risk, surpassing IWCV, Source Risk and results in the original papers. We observe that the selected model according to Source Risk only works well when the domain gap is pretty small as in the cases of D → W and W → D. By contrast, DEV works quite well even when the domain gap is large as in D → A.

5.4. Digits Dataset

Digits (Ganin et al., 2016) dataset consists of three domains: MNIST, USPS and SVHN. Among them, USPS and MNIST are similar, with white digits written by hand on black backgrounds. SVHN, however, is cropped from real street view images and introduces rich background noise.

A state of the art generative model for domain adaptation on the Digits dataset is Generate to Adapt (**GTA**) (Sankaranarayanan et al., 2018). We test the model selection performance of DEV on GTA in Table 4. Besides the learning rate, we also tune the hyperparameters α and β of GTA. Without altering the GTA model, DEV improves GTA (Original) by 1.4%. The model selected by Source Risk works poorly on task SVHN → MNIST. SVHN is visually very different from MNIST. The large domain gap makes the source risk deviate far from the target risk, resulting in inaccurate model selection. By contrast, even under large domain gap, DEV can exploit adapted features for accurate model selection.

5.5. Beyond Standard Domain Adaptation

Standard domain adaptation relies on the assumption that source domain and target domain share the same label set. Several recent works relax the assumption and propose new variants of domain adaptation: Partial Domain Adaptation (PDA) (Cao et al., 2018a;b) and Open Set Domain Adap-

Table 4. Accuracy (%) of **GTA** (Sankaranarayanan et al., 2018) by different validation methods on Digits dataset.

Method	USPS → MNIST	MNIST → USPS	SVHN → MNIST	Avg
Original (Sankaranarayanan et al., 2018)	95.30	90.80	92.40	92.83
Source Risk	92.03	85.92	77.58	85.18
DEV	96.93	92.54	93.18	94.22
Target Risk (Upper Bound)	97.03	92.97	93.51	94.50

Table 5. Accuracy (%) of **PADA** (Cao et al., 2018b) by different validation methods on Office-31 dataset.

Method	A31 → W10	D31 → W10	W31 → D10	A31 → D10	D31 → A10	W31 → A10	Avg
Original (Cao et al., 2018b)	86.54	99.32	100.00	82.17	92.69	95.41	92.69
Source Risk	70.17	98.30	99.32	76.17	88.51	90.92	87.23
IWCV	82.38	97.00	96.42	78.96	89.16	92.23	89.36
DEV	87.80	100.00	100.00	82.94	92.84	95.23	93.15
Target Risk (Upper Bound)	87.80	100.00	100.00	83.59	93.00	95.66	93.34

tation (OSDA) (Panareda Busto & Gall, 2017; Saito et al., 2018b). In partial domain adaptation, the source label set subsumes the target label set, which naturally satisfies the assumption of DEV: **supp** $p_f \supset$ **supp** q_f, where p_f and q_f are the adapted feature distributions. It is interesting to find out that DEV can handle model selection problems in partial domain adaptation without any modification.

We justify this by choosing a state of the art method, Partial Adversarial Domain Adaptation (**PADA**) (Cao et al., 2018b) and tuning it with DEV. Besides the learning rate, we also tune the hyperparameter *update_iteration* of PADA in $\{300, 400, 500, 600, 700\}$ (with 500 as its default setting). The results are shown in Table 5. DEV continues to correspond well with Target Risk, even exceeding the original results reported in (Cao et al., 2018b).

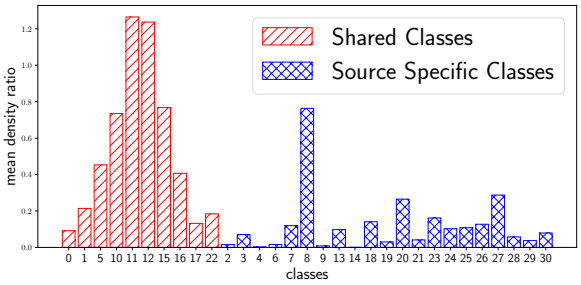


Figure 2. Estimated density ratio averaged across source classes on partial domain adaptation (Cao et al., 2018b) task A → W. Shared classes are marked in red while the other classes marked in blue.

We further analyze the estimated density ratio in Figure 2. Estimating the ground truth density ratio in practical datasets is intractable. However, in partial domain adaptation, $w_f(\mathbf{x})$ for samples associated with source-specific classes should be smaller than for those of the shared classes. We plot the mean density ratio for each class, marking the shared classes in red and the source-specific classes in blue, which justifies the assumption. It seems counterintuitive that the source-specific class 8 has too high density ratio. By zooming in the dataset, we find that class 8 is "*desktop_computer*", which is often confused with the shared classes "*laptop_computer*" and "*monitor*". In summary, the density ratio estimated by DEV is generally accurate, which enables unbiased estimate of the target risk under controlled variance.

5.6. Ablation Study

To disentangle the contributions behind the success of DEV, we conduct an ablation study on VisDA dataset as shown in Table 2. The observation that DEV without control variate is superior to IWCV implies that model selection in Deep UDA can benefit largely from adapting feature representations. By plugging in the control variate method, DEV is further improved by over 2%, indicating the importance of variance control towards an accurate model selection in Deep UDA.

6. Conclusion

This paper introduced Deep Embedded Validation (DEV), an accurate model selection method in Deep UDA. DEV embeds deep adapted representations into the validation procedure to yield more reliable density ratio estimate, and leverages the control variate method to reduce the variance. Theoretical analysis and extensive experiments justify that DEV performs nearly on par with the Target Risk, significantly surpassing the previous methods. The superiority of DEV makes it an accurate, non-intrusive model selection method in the absence of labeled data in the target domain.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61772299, 71690231, and 61672313).

References

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations (ICLR)*, 2019.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 2010.

Bickel, S., Brückner, M., and Scheffer, T. Discriminative learning for differing training and test distributions. In *International Conference on Machine Learning (ICML)*, pp. 81–88, 2007.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 129–136, 2008.

Cao, Z., Long, M., Wang, J., and Jordan, M. I. Partial transfer learning with selective adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a.

Cao, Z., Ma, L., Long, M., and Wang, J. Partial adversarial domain adaptation. In *European Conference on Computer Vision (ECCV)*, pp. 135–150, 2018b.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems*, pp. 442–450, 2010.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2960–2967, 2013.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 2016.

Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 1994–2003, 2018.

Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1137–1145, 1995.

Lemieux, C. *Control Variates*. American Cancer Society, 2017.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Lipton, Z., Wang, Y.-X., and Smola, A. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, pp. 3122–3130, 2018.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015a.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015b.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning (ICML)*, pp. 2208–2217, 2017.

Long, M., Cao, Z., Wang, J., and Jordan, M. I. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 1647–1657, 2018.

Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., and Kim, K. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2010.

Panareda Busto, P. and Gall, J. Open set domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., Saenko, K., Roynard, X., Deschaud, J.-E., Goulette, F., and Hayes, T. L. VisDA: A synthetic-to-real benchmark for visual domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2021–2026, 2018.

Pinheiro, P. O. Unsupervised domain adaptation with similarity learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8004–8013, 2018.

Real, E., Shlens, J., Mazzocchi, S., Pan, X., and Vanhoucke, V. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7473, 2017.

Rényi, A. On measures of information and entropy. In *Berkeley Symposium on Mathematics, Statistics and Probability*, 1961.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018a.

Saito, K., Yamamoto, S., Ushiku, Y., and Harada, T. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision (ECCV)*, 2018b.

Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8503–8512, 2018.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. In *International Conference on Machine Learning (ICML)*, pp. 459–466, 2012.

Shen, J., Qu, Y., Zhang, W., and Yu, Y. Wasserstein distance guided representation learning for domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Sugiyama, M., Krauledat, M., and MÃžller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 2007.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. Cross validation framework to choose amongst models and datasets for transfer learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 547–562, 2010.