Learning Calibrated Uncertainties for Domain Shift: A Distributionally Robust Learning Approach

Haoxuan Wang[1] , **Zhiding Yu**[2] , **Yisong Yue**[3] , **Animashree Anandkumar**[3] ,
Anqi Liu[4*] and **Junchi Yan**[1*]
[1]Department of Computer Science and Engineering and MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University
[2]NVIDIA
[3]Department of Computing and Mathematical Sciences, California Institute of Technology
[4]Department of Computer Science, Johns Hopkins University
{hatchet25, yanjunchi}@sjtu.edu.cn, zhidingy@nvidia.com, aliu@cs.jhu.edu,
{yyue, anima}@caltech.edu

Abstract

We propose a framework for learning calibrated uncertainties under domain shifts, where the source (training) distribution differs from the target (test) distribution. We detect such domain shifts via a differentiable density ratio estimator and train it together with the task network, composing an adjusted softmax predictive form concerning domain shift. In particular, the density ratio estimation reflects the closeness of a target (test) sample to the source (training) distribution. We employ it to adjust the uncertainty of prediction in the task network. This idea of using the density ratio is based on the distributionally robust learning (DRL) framework, which accounts for the domain shift by adversarial risk minimization. We show that our proposed method generates calibrated uncertainties that benefit downstream tasks, such as unsupervised domain adaptation (UDA) and semi-supervised learning (SSL). On these tasks, methods like self-training and FixMatch use uncertainties to select confident pseudo-labels for re-training. Our experiments show that the introduction of DRL leads to significant improvements in cross-domain performance. We also show that the estimated density ratios align with human selection frequencies, suggesting a positive correlation with a proxy of human perceived uncertainties.

1 Introduction

Uncertainty estimation is an important machine learning problem that is central to trustworthy AI [Antifakos *et al.*, 2005; Tomsett *et al.*, 2020]. In addition, many important downstream applications rely on the correct estimation of uncertainties. This includes unsupervised domain adaptation [Zou *et al.*, 2019] and semi-supervised learning [Sohn *et al.*, 2020], where they are used to solicit confident pseudo-labels for re-training. In these applications, reliable pseudo-labels help avoid error propagation and catastrophic failures in early iterations [Kumar *et al.*, 2020].

Obtaining reliable uncertainty estimation is challenging. In contrast to human annotations of labels, obtaining the ground-truth uncertainties from real-world data can be costly or even infeasible. It is also known that the commonly used uncertainty proxies in deep neural networks, such as the softmax output, tend to give overconfident estimates [Guo *et al.*, 2017]. This overconfidence is further amplified under domain shifts, where the target (test) domain and the source training domain differ significantly. Such distribution shifts tend to aggravate the existing issues in uncertainty estimation, leading to wrong but overconfident predictions on unfamiliar samples [Li and Hoiem, 2020].

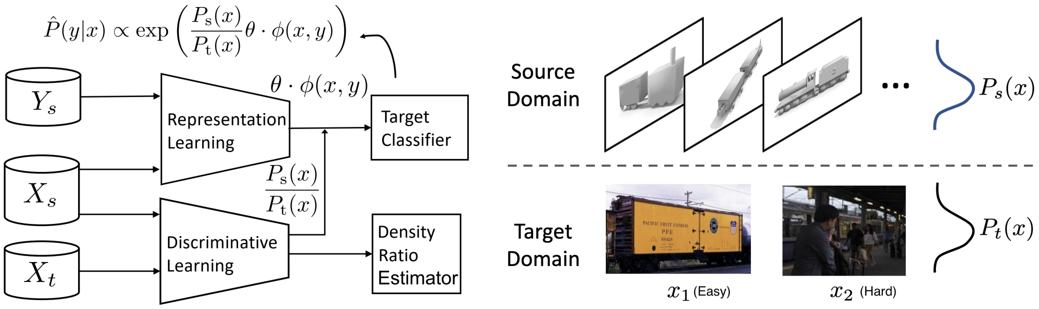
Many methods have been proposed to calibrate the confidence of deep learning models so that the uncertainty level of a model prediction reflects the likelihood of the true event [Guo *et al.*, 2017]. Label smoothing is a popular approach to reduce overconfidence and to promote more uniform outputs [Szegedy *et al.*, 2016]. Temperature scaling is another method where the logit scores are rescaled by a calibrated temperature [Platt and others, 1999]. Approaches such as Monte-Carlo sampling [Gal and Ghahramani, 2016] and Bayesian inference [Blundell *et al.*, 2015] model uncertainties from a Bayesian perspective but are computationally expensive. Though these methods lead to more calibrated uncertainties, recent studies show that their results cannot be fully trusted under domain shifts [Snoek *et al.*, 2019].

Our approach: To handle domain shifts, we characterize the "overlap" between the source training data and the test data. Intuitively, if a test sample is distant from the training distribution, then its confidence level should be lowered. We incorporate this insight by estimating a density ratio for each sample and employ it for confidence calibration.

To be concrete, recall that the probability output for a

[*] Correspondence authors.

[†] Code repository: https://github.com/hatchetProject/Deep-Distributionally-Robust-Learning-for-Calibrated-Uncertainties-under-Domain-Shift



(a) The end-to-end framework of DRL

(b) Density ratios: $\frac{P_s(\boldsymbol{x}_1)}{P_t(\boldsymbol{x}_1)}$ = 2.232, $\frac{P_s(\boldsymbol{x}_2)}{P_t(\boldsymbol{x}_2)}$ = 1.004

Figure 1: (a) Architecture for end-to-end training of our DRL framework (see Sec. 3.3). (b) Examples for category 'Train' in VisDA. The estimated density ratios for the easy and hard target images are shown. The DRL framework gives higher uncertain predictions for the harder example ($\boldsymbol{x}_2$) that is more cluttered and hence not well-represented in the source domain.

	Paradigm	Data Distribution	Prior
RBA	Two-stage	Simple (Gaussian)	Yes
Ours	End-to-end	Complex (Image)	No

Table 1: Comparison with the most related [Liu and Ziebart, 2014].

classification neural network can be expressed as $P(y|\boldsymbol{x}) \propto \exp(\boldsymbol{\theta}_y \cdot \boldsymbol{\phi}(\boldsymbol{x}))$, where $\boldsymbol{\phi}(\boldsymbol{x})$ is the data feature for input $\boldsymbol{x}$, and $\boldsymbol{\theta}_y$ is the model parameter of the y-th class. We instead propose the following predictive form for our neural network:

$$P(y|\boldsymbol{x}) \propto \exp\left(\frac{P_{\mathrm{s}}(\boldsymbol{x})}{P_{\mathrm{t}}(\boldsymbol{x})}\boldsymbol{\theta}_y \cdot \boldsymbol{\phi}(\boldsymbol{x})\right), \tag{1}$$

where $P_{\mathrm{s}}(\boldsymbol{x})$ and $P_{\mathrm{t}}(\boldsymbol{x})$ are the densities of a data sample under the source and target distributions, respectively. When a target sample is close to the source domain (large $P_{\mathrm{s}}(\boldsymbol{x})/P_{\mathrm{t}}(\boldsymbol{x})$), the prediction is confident. However, when a target sample $\boldsymbol{x}$ is far away from the source distribution (small $P_{\mathrm{s}}(\boldsymbol{x})/P_{\mathrm{t}}(\boldsymbol{x})$), the confidence is lowered and the prediction is closer to a uniform distribution. This intuition is analogous to incorporating a sample-wise temperature to adjust the confidence according to the closeness of a test sample to the training distribution.

Eq. 1 is based on the distributionally robust learning (DRL) framework. DRL is an adversarial risk minimization framework that involves a two-player minimax game between a predictor and an adversary [Grünwald *et al.*, 2004]. While many previous DRL methods [Liu *et al.*, 2020; Nakka *et al.*, 2020] operate in low-dimensional spaces using kernel density estimators for the density ratio estimation, we develop a DRL method to scale up to real-world computer vision tasks, which is able to produce calibrated uncertainties under domain shift. **The highlights of the paper are**:

1) For the first time to our best knowledge, we propose a DRL method for uncertainty estimation under domain shift. We introduce a density ratio estimator which learns to predict the density ratios between source and target domains (Fig. 1(a)). The density ratio estimator and the target classifier are trained simultaneously in an end-to-end fashion. We also introduce additional regularization to improve calibration performance. A comparison of our framework with the most related DRL work is shown in Table 1.

2) We show that the estimated density ratio reflects the distance of a test sample from both training and test distributions (Fig. 1(b)). Our experiments further empirically show that these estimates are also well correlated with human selection frequencies, based on the ground-truth labels in ImageNetV2.

3) We empirically show that the top-1 class predictions of DRL are more calibrated than empirical risk minimization and temperature scaling on Office31, Office-Home, and VisDA. We measure the level of calibration using expected calibration error (ECE), Brier Score and reliability plot.

4) We integrate our method as a plug-in module in downstream applications such as unsupervised domain adaption and semi-supervised learning, leading to significant improvements. For example, incorporating self-training with DRL leads to state-of-the-art performance on VisDA-2017 and a 6% improvement on hard examples. Incorporating Fixmatch with DRL improves the original Fixmatch by a relative 17% increase in accuracy under the cross-domain setting.

2 Related Work

Domain classification for domain shift: When dealing with classification problems under distribution shifts, one prevalent way is to discriminate different distributions. A popular approach is importance weighting using density ratio estimation [Sugiyama *et al.*, 2012], which reweighs source samples using pre-estimated density ratios to match the target distribution. However, importance weighting is known to be of high variance and only few works cover high-dimensional data [Khan *et al.*, 2019; Park *et al.*, 2020]. Other methods use a classification network to differentiate the learned source and target representations (adversarially) to locate a common subspace [Ganin *et al.*, 2016; Tzeng *et al.*, 2017; Long *et al.*, 2018]. These works do not analyze the uncertainty estimation problem under domain shift. Yet our density ratio generation method is different from the above approaches in both motivation and implementation. We learn the density ratios using DRL's predictive form for generating calibrated predictions, and helps with choosing better pseudo-labeled data from the target domain in downstream tasks.

Uncertainty calibration in deep models: Uncertainty calibration aims for matching the model output probability with the true frequency of that event [Kumar *et al.*, 2019], and is achieving increasing attention in deep learning [Nixon *et al.*, 2019]. Popular methods include Bayesian deep learning [Gal and Ghahramani, 2016] and temperature scaling [Guo *et al.*, 2017]. However, these methods are either computationally expensive or not designed for the domain shift setting. Methods that focus on calibration for uncertainties under domain shift either do not fundamentally change the uncertainty generation process [Han *et al.*, 2019; Lee and Lee, 2020] or were built upon the traditional importance weighting setup [Park *et al.*, 2020; Wang *et al.*, 2020]. In this paper, we directly generate more calibrated uncertainties based on the DRL framework under domain shift without directly optimizing the calibration error.

Distributionally robust learning under domain shift: Several different learning algorithms can be derived from the DRL framework [Liu and Ziebart, 2014; Fathony *et al.*, 2016; Liu and Ziebart, 2017]. When applied to domain shift cases, an important limitation of these works is that they need density ratios to be estimated beforehand. This requires concrete prior distribution knowledge of the target domain, making them unsuitable for complicated tasks. Our work instead estimates the density ratio along with the learning process and is applicable on real-world tasks. While using a min-max framework, our work is orthogonal to adversarial training, which perturbs the covariate variable [Hu *et al.*, 2018; Najafi *et al.*, 2019] and focuses on robustness against adversarial perturbations. Our work derives a novel predictive form from class-regularized DRL by explicitly solving the min-max game and integrates differentiable density ratio estimation in the end-to-end training process.

3 Distributionally Robust Learning

In this section, we first review the preliminaries of DRL (Sec. 3.1), followed by a proposed variant of DRL with class regularization (Sec. 3.2). We then also propose an instantiation of DRL with a differentiable density ratio estimation network (Sec. 3.3). Finally, we show the applications of our framework in UDA and SSL tasks (Sec. 3.3).

3.1 Preliminaries

Notations and definitions: Denote the input and labels by random variables X and Y, respectively. Use $\boldsymbol{x} \in \mathbb{R}^d$ and $\mathcal{X}$ to represent the realization and sample space of X. Our goal is to find a predictor $\boldsymbol{f}(\boldsymbol{x}) : \mathbb{R}^d \mapsto \mathbb{R}^C$, where $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{f} \in \mathbb{R}^C \cap \Delta$, that is close to the true underlying $P_t(Y|X)$. Here, d, C and Δ denote the input dimension, class number, and probabilistic simplex, respectively. We consider the problem with labeled data sampled from a source distribution $P_{\rm s}(X, Y)$ and unlabeled data sampled from a target distribution $P_{\rm t}(X)$, and use $P_s(X, Y)$ to represent the empirical source distribution. In this work, we consider an important form of domain shift with the covariate shift assumption $P_{\rm s}(X) \neq P_{\rm t}(X)$, $P_{\rm s}(Y|X) = P_{\rm t}(Y|X)$.

Motivation: Traditional empirical risk minimization (ERM) frameworks tend to fail under covariate shift since ERM empirically learns a predictor $\hat{P}_{\rm s}(Y|X)$ from the finite source data that usually fails to generalize to the target distribution. DRL is proposed to overcome this issue, which can be formulated as a two-player adversarial risk minimization game [Grünwald *et al.*, 2004] with the predictor player minimizing a loss, while the adversary player maximizing it. The adversary is allowed to perturb the labels, subject to certain feature-matching constraints to ensure data-compatibility.

Formulation: For covariate shift, DRL [Liu and Ziebart, 2014] deals with the mismatch between the expected loss and the training data, and is defined on **target** distribution:

$$\hat{P}_{\rm t}(Y|X) = \operatorname*{argmin}_{\boldsymbol{f}} \max_{\boldsymbol{g}\in\boldsymbol{\Sigma}} \mathbb{E}_{\boldsymbol{x}\sim P_{\rm t}(X)} \mathcal{L}\left(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x})\right), \quad (2)$$

where $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^C$ are the conditional label distributions given an input $\boldsymbol{x}$. $\boldsymbol{\Sigma}$ is a constraint for $\boldsymbol{g}$ to ensure the invariant conditional label distribution under covariate shift, which we will formulate in Eq. 4. Both $\boldsymbol{f}$ and $\boldsymbol{g}$ are not parameterized yet. $\mathbb{E}_{x\sim P_{\rm t}}\mathcal{L}(\cdot)$ is an expected log loss on the **target** input:

$$\mathbb{E}_{\boldsymbol{x}\sim P_{\rm t}(X)}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x})) \triangleq \mathbb{E}_{\boldsymbol{x}\sim P_{\rm t}(X)}[-\boldsymbol{g}(\boldsymbol{x}) \cdot \log \boldsymbol{f}(\boldsymbol{x})]. \quad (3)$$

$\boldsymbol{f}$ is the predictor player minimizing the loss function while $\boldsymbol{g}$ is the adversary maximizing the loss function. After solving this game, $\boldsymbol{f}$ is our estimate of $\hat{P}_t(Y|X)$, which we will use for the classification task on the target domain.

Eq. 2 is defined on the **target** domain only. How could a predictor be properly trained while there are no target labels available? The answer is that the adversary $\boldsymbol{g}$ is implicitly constrained by the **source** features. We use the following constraints to make sure that $\boldsymbol{g}$ is close to $P_s(Y|X)$:

$$\boldsymbol{\Sigma} = \{\boldsymbol{g}|\textstyle\sum_i g_y \boldsymbol{\phi}(\boldsymbol{x}_i) = \sum_i \mathbb{I}[y_i = y]\boldsymbol{\phi}(\boldsymbol{x}_i), \forall y\}, \quad (4)$$

where $\boldsymbol{x}_i \sim P_s(X)$, g_y is the y-th dimension of $\boldsymbol{g}$ and $\boldsymbol{\phi}(\boldsymbol{x}_i)$ is the feature for $\boldsymbol{x}_i$. Eq. 4 is a necessary but not sufficient condition for $\boldsymbol{g} = P_s(Y|X)$, thus serving as an implicit constraint for $\boldsymbol{g}$ to be close to the true $P_t(Y|X)$ under the covariate shift assumption. Given a predefined feature function $\boldsymbol{\phi}$, when the adversary perturbs the conditional label distribution, certain aggregate function of $\boldsymbol{\phi}$ on $\boldsymbol{g}$ should equal to the counterpart on the empirical source data.

From DRL to density ratio (derivation of *f*): When using the expected target log loss in Eq. 2, Eq. 1 is derived by solving the predictor $\boldsymbol{f}$. Here we refer the derivation details to [Liu and Ziebart, 2014] but emphasize an important property of the prediction: representation-level conservativeness, which means the predictions are more certain for inputs closer to the source domain (larger $P_{\rm s}(\boldsymbol{x})/P_{\rm t}(\boldsymbol{x})$) and more uncertain when $P_{\rm s}(\boldsymbol{x})/P_{\rm t}(\boldsymbol{x})$ is small. This property reflects the model's ability to convey information about what it does not know through the model uncertainty.

3.2 Class-regularized Distributionally Robust Learning

Inspired by label smoothing [Szegedy *et al.*, 2016] and regularization of neural network outputs [Pereyra *et al.*, 2017], we further add class-regularization to the prediction form in Eq. 1. Instead of doing it in a post-hoc way, we incorporate class-regularization into the original DRL formulation. We propose to use a weighted log loss to penalize the high confidence in the adversary's label prediction:

$$\hat{P}_{\text{t}}(Y|X) = \operatorname*{argmin}_{\boldsymbol{f}} \max_{\boldsymbol{g}\in\boldsymbol{\Sigma}} \mathbb{E}_{\boldsymbol{x}\sim P_{\text{t}}(X)}[-\boldsymbol{g}(\boldsymbol{x})\cdot\log\boldsymbol{f}(\boldsymbol{x})] \\ - r\mathbb{E}_{\boldsymbol{x}\sim P_{\text{t}}(X)}\left[\boldsymbol{y}\odot\boldsymbol{g}(\boldsymbol{x})\cdot\log\boldsymbol{f}(\boldsymbol{x})\right], \qquad (5)$$

where $\boldsymbol{y}$ is the one-hot class vector, $\odot$ is the element-wise product, and $r \in [0, 1]$ is a hyper-parameter that controls the level of regularization. $\boldsymbol{\Sigma}$ here is the same as in Eq. 2. We call this formulation **class-regularized** distributionally robust learning. Eq. 5 is a convex-concave function in terms of $\boldsymbol{f}$ and $\boldsymbol{g}$. According to the strong duality, we switch the order of the $\min$ and $\max$. With a fixed $\boldsymbol{g}$, $\boldsymbol{f} = \boldsymbol{g}$ is the optimal solution of the inner $\min$ problem. So we have the following lemma (refer to appendix for proof) :

Lemma 1. *Eq. 5 can be reduced to a regularized maximum entropy problem with the estimator constrained:*

$$\max_{\boldsymbol{f}\in\Sigma} \mathbb{E}_{\boldsymbol{x}\sim P_t(X)}[-\boldsymbol{f}(\boldsymbol{x})\cdot\log\boldsymbol{f}(\boldsymbol{x})] \\ - r\mathbb{E}_{\boldsymbol{x}\sim P_{\text{t}}(X)}\left[\boldsymbol{y}\odot\boldsymbol{f}(\boldsymbol{x})\cdot\log\boldsymbol{f}(x)\right], \qquad (6)$$

where $\boldsymbol{\Sigma}$ is the same as in Eq. 4, meaning that $\boldsymbol{f}$ should be close to the empirical source $P_s(Y|X)$.

Theorem 1. *The solution of Eq. 6 takes the form: $\boldsymbol{f}_{\theta,r}(y|\boldsymbol{x}) \propto \exp\left(\frac{\frac{P_s(\boldsymbol{x})}{P_t(\boldsymbol{x})}\boldsymbol{\theta}_y\cdot\boldsymbol{\phi}(\boldsymbol{x})+r\mathbb{I}(y)}{r\mathbb{I}(y)+1}\right)$, where $\boldsymbol{\theta}$ represents the model parameters and $\mathbb{I}(y)$ is the yth dimension of the one-hot encoding $\boldsymbol{y}$.*

The proof of Theorem 1 (refer to appendix) follows the same principles of deriving Eq. 1. Different from [Liu and Ziebart, 2014], we use Theorem 1 as the new prediction form. In training, $\boldsymbol{y}$ is the one-hot encoding of each class. In inference, we set $\boldsymbol{y}$ to be an all-one vector.

Class-level regularization: Hyperparameter r adjusts the smoothness of $\boldsymbol{g}$'s label prediction in Eq. 5. It translates to the $r\boldsymbol{y}$ term in the prediction form. Intuitively, it increases the correct label's prediction logits when it is smaller than 1 and decreases the logits when it is larger than 1. Thus r provides additional regularization and smoothness to the conservative prediction.

3.3 Differentiable Density Ratio Estimation

Estimating $P_{\text{s}}(\boldsymbol{x})/P_{\text{t}}(\boldsymbol{x})$ is challenging [Sugiyama *et al.*, 2012], especially in high-dimensional spaces. Usually an estimator that calculates the densities in advance is used. But these estimates are usually sub-optimal in practice due to the different downstream tasks' objectives. We propose an end-to-end framework where **the density ratio estimator is trained together with the target classifier**. We introduce the novel differentiable density ratio estimation before proposing a joint training loss and the parameter learning process.

Differentiable density ratio estimation: Based on the Bayes' rule, $P_{\text{s}}(\boldsymbol{x})/P_{\text{t}}(\boldsymbol{x})$ can be computed from a conditional domain classifier [Bickel *et al.*, 2007]: $\frac{P_{\text{t}}(\boldsymbol{x})}{P_{\text{s}}(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|\text{t})}{P(\boldsymbol{x}|\text{s})} = \frac{P(\text{t}|\boldsymbol{x})P(\text{s})}{P(\text{s}|\boldsymbol{x})P(\text{t})}$. Concretely, they can be estimated via binary classification using unlabeled source and target data with $\frac{P(\text{s})}{P(\text{t})}$ as a constant relating to the number of source and target samples. On the other hand, we observe that $P_{\text{s}}(\boldsymbol{x})/P_{\text{t}}(\boldsymbol{x})$ can be a trainable weight for each sample, and can be updated via the training objective of DRL. Therefore, we propose to train a binary classifier as the density ratio estimator, using both the binary cross-entropy loss and DRL's objective loss to update the network's parameters.

Our approach is different from existing methods (e.g. [Che *et al.*, 2021]) in three aspects: First, our binary classification network is proposed for the goal of achieving calibrated uncertainties under domain shift; Second, the weights trained this way lose their original properties as density ratios, but still reflect the relation between the two domains; Third, our obtained density ratios are mathematically the inverse of other methods' density ratios due to the DRL formulation.

Joint training loss: Assume $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{w}_r)$ is the representation learning neural network with parameter $\boldsymbol{w}_r$. We further define $P_d(X)$ as the joint distribution of both source data and target data with their domain labels $D = \{\boldsymbol{d}(\boldsymbol{x})\}$. We denote $\boldsymbol{\tau}(\boldsymbol{x}, \boldsymbol{w}_d) = (\tau_s, \tau_t)$ (where $\tau_s + \tau_t = 1$) as the two dimensional probability output of the source and target domains from a domain classifier $\boldsymbol{\tau}$ with parameter $\boldsymbol{w}_d$. Our joint training loss is defined as:

$$\min_{\boldsymbol{w_r},\theta,\boldsymbol{w_d}} \mathbb{E}_{\boldsymbol{x}\sim P_{\text{t}}(X)}\left[-\boldsymbol{g}_t(\boldsymbol{x})\cdot\log\boldsymbol{f}(\boldsymbol{x};\boldsymbol{w}_r,\boldsymbol{\theta},\boldsymbol{w}_d)\right] \\ + \mathbb{E}_{\boldsymbol{x}\sim P_d(X)}\left[-\boldsymbol{d}(\boldsymbol{x})\cdot\log\boldsymbol{\tau}(\boldsymbol{x},\boldsymbol{w}_d)\right], \qquad (7)$$

where $\boldsymbol{g}_t(\boldsymbol{x}) = P_t(Y|X = \boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{w}_r, \boldsymbol{\theta}, \boldsymbol{r}_d)$ takes the form in Theorem 1:

$$\boldsymbol{f}(\boldsymbol{x};\boldsymbol{w}_r,\boldsymbol{\theta},\boldsymbol{r}_d) \propto \\ \exp\left(\left(\frac{\tau_s(\boldsymbol{x},\boldsymbol{w}_d)}{\tau_t(\boldsymbol{x},\boldsymbol{w}_d)}\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{w}_r) + r\mathbb{I}(y)\right)/(r\mathbb{I}(y)+1)\right). \qquad (8)$$

Parameter learning: Note that $\boldsymbol{g}_t$ in the first loss term of Eq. 7 concerns the conditional label distribution on target which is assumed to be not available. However, with the DRL formulation, we can compute and evaluate the gradients of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{w}_r)$ directly with a change of measure in the derivation of the gradients so that the gradients' calculation does not depend on the target distribution (see details in appendix). We show this in Fig. 1(a) that the representation learning network $\boldsymbol{\phi}$ only uses source data as the input. In this way, the parameter learning process that **originally depend on source and target data distributions (loss function) are now only associated with the source data and labels (gradient)**. We then update $\boldsymbol{\theta}$ and $\boldsymbol{w}_r$ using the computed gradients and also directly back-propagate from the second loss term in Eq. 7 to update $\boldsymbol{w}_d$. Finally, we treat the densities as trainable variables and derive gradients for them from the

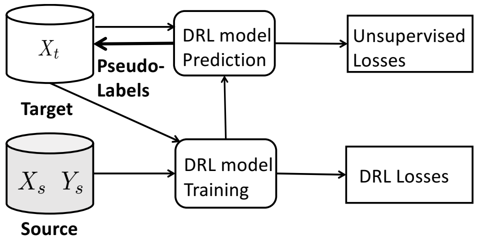


Figure 2: Formulation of the pseudo label based UDA or SSL methods with DRL. The unsupervised losses represent the loss imposed on the unlabeled target data. DRST conducts this procedure multiple iterations, while DRSSL minimizes the unsupervised losses on the augmented target data.

Algorithm 1 End-to-end Training for DRL

```
Input: DNN $\boldsymbol{\phi}$ and DNN $\boldsymbol{\tau}$, with optimizer SGD$_1$ and SGD$_2$, respectively. Learning rates $\gamma_1$ and $\gamma_2$, epoch number $T$.
Initialization: $\boldsymbol{\phi}, \boldsymbol{\tau} \leftarrow$ random initialization, epoch $\leftarrow 0$
While epoch $< T$
    For each data mini-batch
        Update $\boldsymbol{\tau}$ by SGD$_1(\gamma_1)$ using the combined gradients from both loss terms in Eq. 7;
        Compute $\boldsymbol{f}$ using $\boldsymbol{\theta}$, $\boldsymbol{w}_r$, and $\boldsymbol{w}_d$;
        Update $\boldsymbol{\phi}, \boldsymbol{\theta}$ by SGD$_2(\gamma_2)$ using derived gradients;
    epoch $\leftarrow$ epoch $+1$
Output: Trained networks $\boldsymbol{\phi}$, $\boldsymbol{\tau}$.
```

first loss term (details shown in appendix). By the Bayes rule, $\frac{P_s(\boldsymbol{x})}{P_t(\boldsymbol{x})} = \frac{\tau_s P(t)}{\tau_t P(s)}$, where $P(t)$ and $P(s)$ are the amount of unlabeled data from each domain during the training process. Since we use the same amount of source and target data in each batch, they are canceled out by following $\frac{P(s)}{P(t)} = 1$. Then $\boldsymbol{f}$ in Theorem 1 is reduced to Eq. 8. Therefore, besides the binary classification loss, the parameter $\boldsymbol{w}_d$ of the discriminative network is also trained with gradients from the first loss term. Algorithm 1 shows the details.

Applications to UDA and SSL: We show how to incorporate end-to-end DRL within a framework that takes unlabeled data for training. We introduce the general setting with two examples: self-training based UDA and cross-domain SSL, followed by a new self-training algorithm DRST (distributionally robust self-training) and a new semi-supervised learning algorithm DRSSL (distributionally robust semi-supervised learning).

General settings: In many cases, the source domain may have abundant labels but the target domain lacks enough labels. Typical problems under this setting include unsupervised domain adaptation and semi-supervised learning. In both cases, a common strategy is to treat the prediction results on the target data as pseudo labels to train the model on target data. In these methods, model confidence (softmax output) is often leveraged as the proxy to rank the reliability of pseudo labels, with the underlying assumption that there exists a positive correlation between model confidence and pseudo label quality. However, such an assumption requires accurate uncertainty estimation to avoid false usage of wrong pseudo labels which may poison the training. To this end, we incorporate DRL into these frameworks in order to provide calibrated uncertainties. An illustration of this setting is shown in Fig. 2.

Distributionally robust self-training: In UDA, we are given labeled source data and unlabeled target data and aim to achieve adaptation from the source to the target domain. Self-training is an effective method for UDA, where the training procedure in Fig. 2 is conducted multiple times. Here we propose DRST to plug the class-regularized DRL model into self-training. The idea is to regard each training epoch as a new domain shift problem in DRL. After each training epoch, we make predictions on the target domain and select more confident data and generate pseudo-labels for them to merge into the source training data. Both the pseudo-labels and the model confidence are achieved from Theorem 1. Then the labeled source data and the newly pseudo-labeled target data become the new source set for the next training epoch.

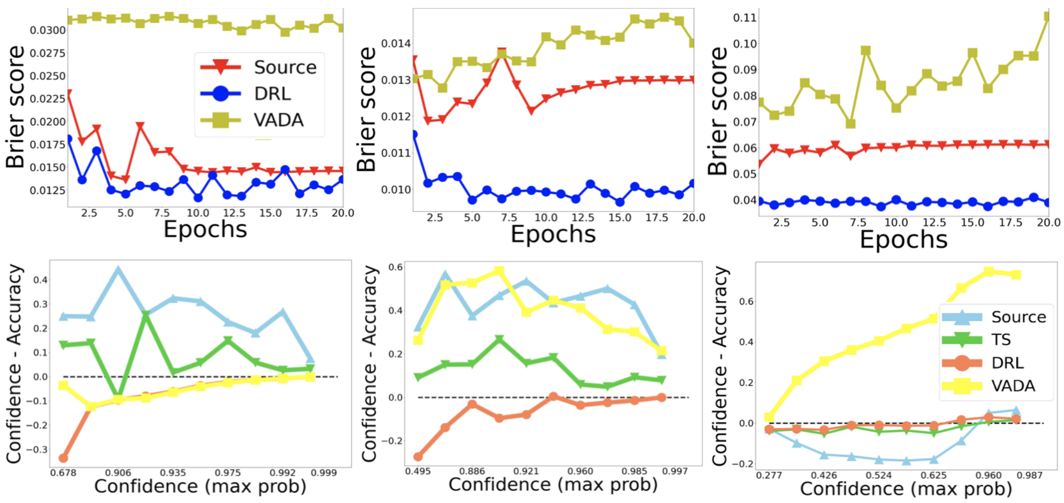
Distributionally robust semi-supervised learning: In cross-domain SSL where there is little labeled source data and much unlabeled target data, we aim to utilize unlabeled data in the target domain to help representation learning and save the effort needed for labeling. One effective strategy is to use pseudo-labels generated from weakly-augmented data to supervise strongly-augmented data. Here 'weakly' means simple flip-and-shift data augmentation while 'strongly' follows the same strategy as FixMatch [Sohn *et al.*, 2020]. Using DRL's prediction, we propose DRSSL, which assigns pseudo-labels more conservatively. Practically, we plug DRL into Fixmatch. Here the unsupervised loss in Fig. 2 is the 'consistency loss' in FixMatch: $\mathcal{L}_u = \frac{1}{M}\sum_{m=1}^{M}\mathbb{I}(\max(\hat{P}(y_t^w|x_t^w)) > \eta)H(\hat{y}_t^w, \hat{P}(y_t^s|x_t^s))$, where x_t^w and y_t^w represent the weakly-augmented target data, x_t^s and y_t^s represent the strongly-augmented version of the same image data, and η is a threshold for generating pseudo-labels $\hat{y}_t^w$.

4 Experiments

We evaluate our method on benchmark datasets and compare our performance with other uncertainty quantification, calibration, and domain adaptation baseline methods. DRL is evaluated as a method providing more calibrated uncertainties (Sec. 4.1), DRST as an effective UDA method (Sec. 4.2) and DRSSL as a cross-domain SSL method. We show additional results and details in the appendix.

Datasets and methods: We use Office31 [Saenko *et al.*, 2010], Office-Home [Venkateswara *et al.*, 2017] and VisDA2017 [Peng *et al.*, 2017] for evaluating DRL's uncertainties. We compare DRL with temperature scaling (TS), VADA [Shu *et al.*, 2018] and source-only. We also train models using ImageNet [Deng *et al.*, 2009] as the source domain and ImageNetV2 [Recht *et al.*, 2019] as the target domain to check the relationship between our estimated weights and the human selection frequencies (HSF) [Chen *et al.*, 2020a].

VisDA2017 is also used to evaluate UDA performance. We compare with (1) traditional UDA baselines: MCD [Saito *et al.*, 2018b] and ADR [Saito *et al.*, 2018a]; (2) self-training



(a) Office31 (A→W) (b) Office-Home (P→A) (c) VisDA

Figure 3: Brier score (top) and reliability diagrams (bottom) on **Office31**, **Office-Home** and **VisDA**. DRL generates more calibrated uncertainties than source-only and temperature scaling and VADA. Brier score measures the mean squared difference between the predicted probability and the actual outcome. For a fully calibrated classifier, the confidence should match the accuracy across the full range of confidence. Thus the closer the lines are to the dashed line, the more calibrated the method is. Our method gets more and more calibrated as the confidence increases. Note that in the first row, TS's Brier scores are much larger and excluded to not effect the scale.

HSF	Source	Temp.Scal.	DRL (Ours)
[0.0, 0.2]	0.2694	0.2624	**0.0129**
[0.2, 0.4]	0.1818	0.1745	**0.0036**
[0.4, 0.6]	0.1344	0.1281	**0.0012**
[0.6, 0.8]	0.0667	0.0601	**0.0019**
[0.8, 1.0]	0.0319	0.0246	**0.0019**

Table 2: Expected calibration error (ECE) comparison on ImageNetV2 under different HSF.

baselines: CBST [Zou *et al.*, 2018] and CRST [Zou *et al.*, 2019]; (3) methods that tackle domain adaptation with uncertainty: BRER [Han *et al.*, 2019] and MUDA [Lee and Lee, 2020]; (4) uncertainty quantification methods combined with self-training: AVH [Chen *et al.*, 2020a]+CBST.In addition, we use CIFAR10, STL10 [Coates *et al.*, 2011], MNIST [Lecun and Bottou, 1998] and SVHN [Netzer *et al.*, 2011] to construct cross-domain SSL settings, which has few source labeled data and much unlabeled target data and show DRSSL's advantages in cross-domain SSL over Fixmatch [Sohn *et al.*, 2020].

Evaluation metrics: Apart from accuracy, we also use Brier score [Brier, 1950], expected calibration error (ECE) [Guo *et al.*, 2017], and reliability plots [Guo *et al.*, 2017] to evaluate the performance of our proposed method and the baselines. Brier score measures the mean squared difference between the predicted probability assigned to the possible outcome and the actual outcome. ECE is defined as the sum of average difference between prediction accuracy and confidence of different confidence bins (we use 15 bins in practice). Despite the potential problems of ECE [Nixon *et al.*, 2019], it is still the most prevalent metric for the top-1 prediction. For both the Brier score and ECE, the lower the score, the more calibrated is the model.

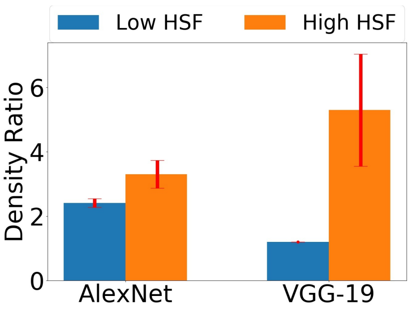
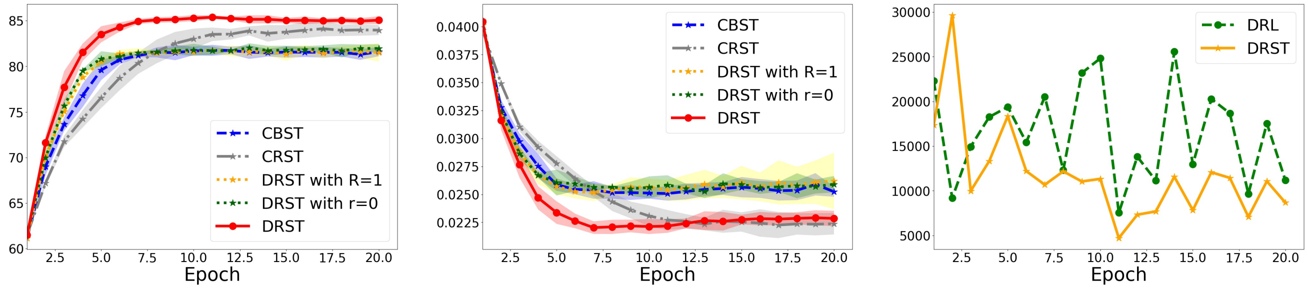


Figure 4: Density ratios vs HSF on ImageNetV2.

Experimental setup: For Office31 and Office-Home tasks, we use ResNet50 [He *et al.*, 2016] as the backbone for all models. We train with SGD for 100 epochs and set the learning rate to 0.001. For VisDA, we use ResNet101 and SGD optimizer. During the 20 epochs of training, the initial learning rate is set as 10^{-5} and the weight decay parameter is set as 5×10^{-4}. For ImageNet, we follow the standard training process of AlexNet [Krizhevsky *et al.*, 2012] and VGG-19 [Simonyan and Zisserman, 2014], where the initial learning rate is 0.01 and we decay the learning rate by a factor of 10 for every 30 epochs. All of the training are done on DGX V100 Tesla V100 GPUs with 32GB memory. The main packages and corresponding versions are: PyTorch 0.4.0, CUDA 10.1.

4.1 Calibrated Uncertainties from DRL

DRL's calibrated confidence: Fig. 3 shows our method achieves better uncertainty calibration. Note that the type of uncertainty we measure here is the epistemic uncertainty [Gal



(a) Test accuracy (b) Brier score (c) Covariate shift

Figure 5: (a)-(b) Results on VisDA-17 (performed with 5 random seeds) with test accuracy and Brier score. DRST outperforms the baselines significantly. (c) We adopt distribution gap $P_s(\boldsymbol{\phi}(\boldsymbol{x}))/P_t(\boldsymbol{\phi}(\boldsymbol{x})) - P_s(\boldsymbol{\phi}(\boldsymbol{x}), y)/P_t(\boldsymbol{\phi}(\boldsymbol{x}), y)$ as a proxy of covariate shift. DRST helps further reduce this gap with self-training.

and Ghahramani, 2016], caused by the lack of data and domain gaps. The aleatoric uncertainty is caused by noise inherent in data and is out of our scope. DRL tends to be underconfident and conservative but still stays closer to the calibration line (dashed line). Table 2 shows our method obtains better ECE on ImageNetV2 in different human selection frequency (HSF) bins. HSF is defined as the average number of times an image from a given class is classified correctly by a group of annotators. Say there are n annotators and an image with ground-truth label "car", p annotators recognize the image as "car", then the image's HSF is $\frac{p}{n}$. The lower the HSF of an image, the harder for humans to correctly classify it. Appendix provides additional results and shows that the accuracy of DRL is also competitive.

Density ratio v.s. HSF: Fig. 4 shows that images with low HSF (visually harder ones) have smaller density ratios, indicating that the estimated density ratios are positively correlated with HSF on ImageNetV2 under different network architectures. Here we regard the value of HSF from $[0, 0.2]$ as low and $(0.2, 1]$ as high.

4.2 Unsupervised Domain Adaptation

Initialization with ASG: The pretrained models impact performance. Automated synthetic-to-real generalization (ASG) model [Chen *et al.*, 2020b], which uses ImageNet pretrained knowledge, can be used to improve synthetic training and self-training. Hence, we use the ASG pretrained model as initialization, with results in Table 3's last four rows.

Accuracy and calibration: Fig. 5(a) and Table 3 show that DRST performs best in accuracy. Our vanilla version of DRST outperforms CRST by over 5% with ASG initialization. We improve the SOTA self-training accuracy on VisDA by over 1%. In the rest of the results, we use DRST to represent DRST-ASG. In comparison with uncertainty based methods (not shown in the table), BRER achieves an accuracy of 80.59±1.39% while ours is 83.75% using the same source model. MUDA aims to minimize model uncertainty and only yields a result of 78.5%. We only identified two methods with higher accuracy on VisDA2017: CDTrans [Xu *et al.*, 2021] (using transformer) and CPGA [Qiu *et al.*, 2021] (two stage learning by calculating prototypes). Both of them cannot estimate uncertainty and it is nontrivial for them to incorporate uncertainty or pseudo-label modeling. Fig. 5(b) shows that DRST also achieves better calibrated performance.

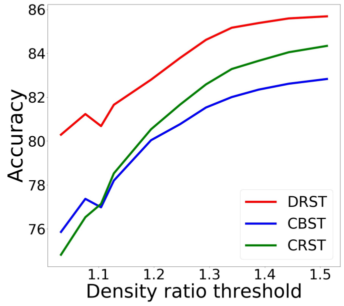


Figure 6: Accuracy vs. estimated density ratio weights. Improvement from DRST increases on harder examples (lower weights).

Ablation study: Fig. 5(a)(b) include two ablation methods. In the first ablation, we set r to 0 so that there is no class regularization in DRL ("r = 0"). The prediction then follows the form in Eq. 1. In the second ablation, we set the density ratios to 1 instead of calculating it to mute the differentiable density ratio estimation in our method so that there is no representation level conservativeness ("R = 1"). When one method is existent, the performance is not degraded but also not boosted. However, DRST achieves the best results when both components are present, showing faster convergence and better performance.

Covariate shift: Fig. 5(c) shows how well the covariate shift assumption holds over the training process. We calculate $P_s(\boldsymbol{\phi}(\boldsymbol{x}))/P_t(\boldsymbol{\phi}(\boldsymbol{x})) - P_s(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{y})/P_t(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{y})$ using discriminative density ratio estimators (per class) as a proxy of covariate shift as it becomes 0 when covariate shift holds. We can see that the gap decreases with self-training, showing that even though covariate shift may not hold in the beginning, self-training helps to promote this assumption with better aligned domains and more discriminative feature distributions. Note that our model is tailored and effective to covariate shift no matter other shifts (e.g. label shift) exist or not. For example, we find label shift exists in VisDA2017 and our method still shows effectiveness: we calculate the division of the number of source samples over the number of

Method	Aero	Bike	Bus	Car	Horse	Knife	Motor	Person	Plant	Skate	Train	Truck	Mean
Source [Saito *et al.*, 2018a]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
MCD [Saito *et al.*, 2018b]	87.0	60.9	**83.7**	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
ADR [Saito *et al.*, 2018a]	87.8	79.5	**83.7**	65.3	92.3	61.8	88.9	73.2	87.8	60.0	85.5	32.3	74.8
CBST [Zou *et al.*, 2018]	87.2	78.8	56.5	55.4	85.1	79.2	83.8	77.7	82.8	88.8	69.0	**72.0**	76.4
CRST [Zou *et al.*, 2019]	88.0	79.2	61.0	60.0	87.5	81.4	86.3	78.8	85.6	86.6	73.9	68.8	78.1
CBST-AVH [Chen *et al.*, 2020a]	93.3	80.2	78.9	60.9	88.4	89.7	88.9	79.6	89.5	86.8	81.5	60.0	81.5
DRST (Ours)	93.47	86.30	65.74	68.03	93.99	95.08	87.34	83.30	92.97	88.65	83.66	66.42	83.75
ASG [Chen *et al.*, 2020b]	88.81	68.55	65.31	78.06	**95.78**	9.11	84.89	29.58	82.13	33.76	**86.00**	12.04	61.17
CBST-ASG [Chen *et al.*, 2020b]	**95.12**	**86.53**	79.83	76.01	94.61	92.34	85.94	75.08	89.23	82.16	73.42	56.49	82.23
CRST-ASG [Chen *et al.*, 2020b]	92.38	81.30	74.63	**84.40**	90.90	92.43	**91.65**	**83.78**	**94.92**	88.12	74.88	61.10	84.21
DRST-ASG (Ours)	94.51	85.58	76.50	77.18	94.39	**95.33**	88.89	81.23	94.22	**90.36**	81.75	63.10	**85.25**

Table 3: Accuracy comparison with different UDA and self-training methods on VisDA2017. “Skate” denotes “Skateboard”.

Dataset	CIFAR10	STL10	MNIST	SVHN
Fixmatch	91.60	59.61	99.43	26.50
DRSSL (Ours)	**95.17**	**69.38**	**99.46**	**30.96**

Table 4: Accuracy of cross-domain SSL in comparison with the peer method Fixmatch [Sohn *et al.*, 2020].

target samples for each class, the numbers range from 1.42 to 4.23, indicating the existence of label shift in the dataset. Seeing the co-existence of multiple shifts and especially covariate shift, we believe our method is practical. This in fact has also well been verified by our real-world image experiments beyond covariate shift.

Improvement on hard examples: Fig. 6 demonstrates that compared to the baselines, DRST achieves larger performance gain on target samples with smaller density ratios. Recall that data is not well-represented in the source domain and is regarded as visually hard when the density ratio $P_{\rm s}(\boldsymbol{x})/P_{\rm t}(\boldsymbol{x})$ is small (Fig. 1(b)). Therefore, DRST provides more robust performance on harder examples.

Density ratios: Our density ratios are estimated from a differentiable domain classifier and is not guaranteed to match the true density ratios. However, our density ratios are interpretable as they reflect the closeness of a sample to the two domains and benefit the downstream tasks. In Fig. 1(b), a harder example obtains a lower density ratio due to its vague shape (more examples are in the appendix). Moreover, the magnitude of our estimated density ratios are modest in general within the range of $[0.1, 10]$ (instead of approaching 0 or infinity) due to the regularization by the target task classification network's learning signals.

Improved attention: We visualize the model attention of DRST using Grad-CAM [Selvaraju *et al.*, 2017] and compare with the CBST and CRST baselines in the appendix. The results show that DRST renders improved attention with better object coverage, presenting a better reception field.

4.3 Distributionally Robust Cross-Domain SSL

To introduce domain gaps into semi-supervised learning, we choose the emerging cross-domain semi-supervised learning (CDSSL) [Yu *et al.*, 2019] setting. Specifically, CDSSL aims to use few labeled source training examples and many unlabeled target training examples to predict on source testing examples and target testing examples. Two pairs of source and target domains are used: Source: CIFAR10 / Target: STL10 and Source: MNIST / Target: SVHN. Under the single domain setting, FixMatch trains the model using source labeled data and source unlabeled data. However, under the cross-domain setting, we train the model using source labeled data and target unlabeled data.

Table 4 shows that DRL-powered SSL method improves the Fixmatch baseline significantly on CDSSL tasks. Note that our setting is different from UDA where source labeled data is abundant. SSL focuses on using unlabeled augmented data to learn from few labeled data. In the CIFAR10 to STL10 case, we only have 4k labeled source data. For MNIST to SVHN, we have 40k labeled source data. The results show that DRL is beneficial for generating high-quality pseudo supervision for unlabeled data under the cross-domain SSL setting. More details of the experiments are provided in the appendix.

5 Conclusion

This paper studied uncertainty estimation under distribution shift with the distributionally robust learning framework. We show that density estimation can be integrated into the learning process by using a domain classifier. We propose differentiable density ratio estimation and develop end-to-end training techniques for our method. Using DRL's more calibrated model confidence helps to generate better pseudo-labels for self-training in UDA and cross-domain SSL.

We also empirically show that the density ratios learned from our domain classifier reflect the hardness of an image, showing a positive correlation with the human selection frequencies. Future work involves relaxing the assumptions made and study different shifts in the DRL framework, which is in fact a limitation of our work.

Acknowledgments

Haoxuan Wang started the work during the Summer Undergraduate Research Fellowship Program in Caltech. Anqi Liu was supported by the PIMCO and Amazon Postdoc Fellowship in Caltech and the Amazon-JHU AI2AI faculty award in JHU. Junchi Yan was supported by NSFC 62222607.

References

[Antifakos *et al.*, 2005] Stavros Antifakos, Nicky Kern, Bernt Schiele, and Adrian Schwaninger. Towards improving trust in context-aware systems by displaying system confidence. In *International Conference on Human Computer Interaction with Mobile Devices and Services*, 2005.

[Bickel *et al.*, 2007] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, 2007.

[Blundell *et al.*, 2015] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015.

[Brier, 1950] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 1950.

[Che *et al.*, 2021] Tong Che, Xiaofeng Liu, Site Li, Yubin Ge, Ruixiang Zhang, Caiming Xiong, and Yoshua Bengio. Deep verifier networks: Verification of deep discriminative models with deep generative models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7002–7010, May 2021.

[Chen *et al.*, 2020a] Beidi Chen, Weiyang Liu, Zhiding Yu, Jan Kautz, Anshumali Shrivastava, Animesh Garg, and Anima Anandkumar. Angular visual hardness. In *ICML*, 2020.

[Chen *et al.*, 2020b] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *ICML*, 2020.

[Coates *et al.*, 2011] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Fathony *et al.*, 2016] Rizal Fathony, Anqi Liu, Kaiser Asif, and Brian Ziebart. Adversarial multiclass classification: A risk minimization perspective. In *NIPS*, 2016.

[Gal and Ghahramani, 2016] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016.

[Grünwald *et al.*, 2004] Peter D Grünwald, A Philip Dawid, et al. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *Annals of statistics*, 2004.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[Han *et al.*, 2019] Ligong Han, Yang Zou, Ruijiang Gao, Lezi Wang, and Dimitris Metaxas. Unsupervised domain adaptation via calibrating uncertainties. In *CVPR Workshops*, volume 9, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Hu *et al.*, 2018] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

[Khan *et al.*, 2019] Haidar Khan, Lara Marcuse, and Bülent Yener. Deep density ratio estimation for change point detection. *arXiv preprint arXiv:1905.09876*, 2019.

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

[Kumar *et al.*, 2019] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Kumar *et al.*, 2020] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *ICML*, 2020.

[Lecun and Bottou, 1998] Y. Lecun and L. Bottou. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[Lee and Lee, 2020] JoonHo Lee and Gyemin Lee. Model uncertainty for unsupervised domain adaptation. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020.

[Li and Hoiem, 2020] Zhizhong Li and Derek Hoiem. Improving confidence estimates for unfamiliar examples. In *CVPR*, 2020.

[Liu and Ziebart, 2014] Anqi Liu and Brian Ziebart. Robust classification under sample selection bias. In *NIPS*, 2014.

[Liu and Ziebart, 2017] Anqi Liu and Brian D Ziebart. Robust covariate shift prediction with general losses and feature views. *arXiv preprint arXiv:1712.10043*, 2017.

[Liu *et al.*, 2020] Anqi Liu, Guanya Shi, Soon-Jo Chung, Anima Anandkumar, and Yisong Yue. Robust regression for safe exploration in control. In *Learning for Dynamics and Control*, 2020.

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

[Najafi *et al.*, 2019] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*, 2019.

[Nakka *et al.*, 2020] Yashwanth Kumar Nakka, Anqi Liu, Guanya Shi, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems. *IEEE Robotics and Automation Letters*, 2020.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[Nixon *et al.*, 2019] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, 2019.

[Park *et al.*, 2020] Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *AISTATS*, 2020.

[Peng *et al.*, 2017] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[Pereyra *et al.*, 2017] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[Platt and others, 1999] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

[Qiu *et al.*, 2021] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *ArXiv*, abs/2106.15326, 2021.

[Recht *et al.*, 2019] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019.

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

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

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

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

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

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Snoek *et al.*, 2019] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[Sugiyama *et al.*, 2012] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[Tomsett *et al.*, 2020] Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid trust calibration through interpretable and uncertainty-aware ai. *Patterns*, 2020.

[Tzeng *et al.*, 2017] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

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

[Wang *et al.*, 2020] Ximei Wang, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable calibration with lower bias and variance in domain adaptation. *arXiv preprint arXiv:2007.08259*, 2020.

[Xu *et al.*, 2021] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *ArXiv*, abs/2109.06165, 2021.

[Yu *et al.*, 2019] Fuxun Yu, Di Wang, Yinpeng Chen, Nikolaos Karianakis, Pei Yu, Dimitrios Lymberopoulos, and Xiang Chen. Unsupervised domain adaptation for object detection via cross-domain semi-supervised learning. *ArXiv*, abs/1911.07158, 2019.

[Zou *et al.*, 2018] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

[Zou *et al.*, 2019] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.