DIRL: Domain-Invariant Representation Learning for Sim-to-Real Transfer

Ajay Kumar Tanwani
University of California, Berkeley
ajay.tanwani@berkeley.edu

Abstract: Generating large-scale synthetic data in simulation is a feasible alternative to collecting/labelling real data for training vision-based deep learning models, albeit the modelling inaccuracies do not generalize to the physical world. In this paper, we present a domain-invariant representation learning (DIRL) algorithm to adapt deep models to the physical environment with a small amount of real data. Existing approaches that only mitigate the covariate shift by aligning the marginal distributions across the domains and assume the conditional distributions to be domain-invariant can lead to ambiguous transfer in real scenarios. We propose to jointly align the marginal (input domains) and the conditional (output labels) distributions to mitigate the covariate and the conditional shift across the domains with adversarial learning, and combine it with a triplet distribution loss to make the conditional distributions disjoint in the shared feature space. Experiments on digit domains yield state-of-the-art performance on challenging benchmarks, while sim-to-real transfer of object recognition for vision-based decluttering with a mobile robot improves from 26.8% to 91.0%, resulting in 86.5% grasping accuracy of a wide variety of objects. Code and supplementary details are available at: https://sites.google.com/view/dirl

Keywords: Domain Adaptation, Adversarial Learning, Sim-to-Real Transfer

1 Introduction

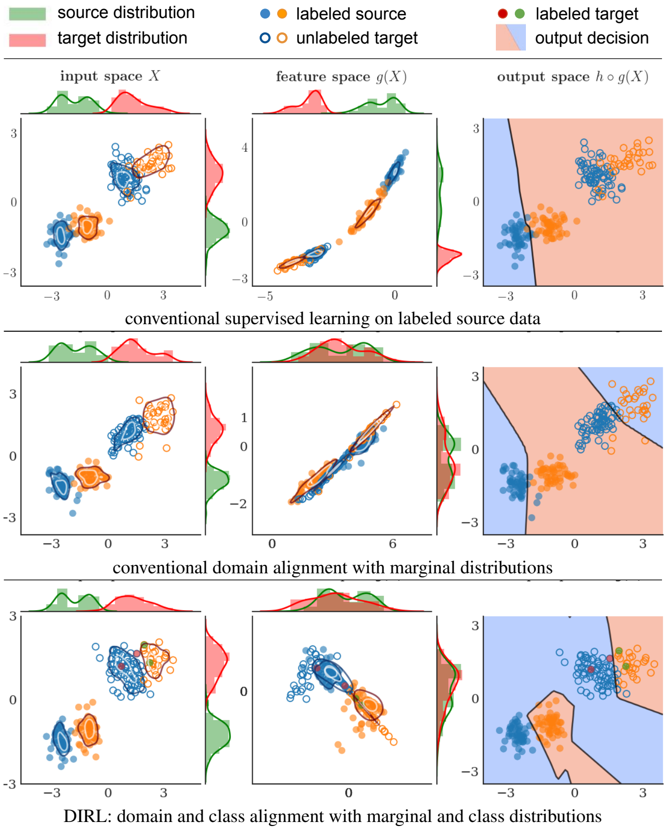
Data-driven deep learning models tend to perform well when plenty of labelled training data is available and the testing data is drawn from the same distribution as that of the training data. Collecting and labelling large scale domain-specific training data for robotics applications, however, is time consuming and cumbersome [1]. Additionally, the sample selection bias in data collection limits the model use to very specific environmental situations. Training deep models in simulation for robot manipulation is becoming a popular alternative [2, 3, 4]. Despite the efforts to build good dynamic models and high fidelity simulators for scalable data collection, the modeling inaccuracies makes it difficult to transfer the desired behaviour in real robot environments.

This paper investigates the problem of sample-efficient domain adaptation to learn a deep model that can be transferred to a new domain [5]. We analyze the situation where we have a lot of labeled training data from the simulator or the *source* distribution, and we are interested in adapting the model with limited or no labelled training data drawn from the *target* distribution. Existing popular approaches to domain adaptation learn common feature transformations by aligning marginal distributions across domains in an unsupervised manner [6, 7, 8, 9]. They implicitly assume that the class conditional distributions of the transformed features is also similar across domains, i.e., if the model performs well on the source domain and the features overlap across domains, the model is going to perform well on the target domain. This, however, creates ambiguity in class alignment across source and target domains in the shared feature space that can result in a negative transfer to a new domain as shown in Fig. 1 (middle). More intuitively, the feature space can mix apples of the simulator with oranges of the target domain and yet have have a perfect alignment of marginal distributions.

To this end, we present a **domain-invariant representation learning** (DIRL) algorithm that resolves the ambiguity in domain adaptation by leveraging upon a

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

few labeled examples of the target domain to align both the marginal and the conditional distributions in the shared feature space with adversarial learning, and increases the inter-class variance and reduces the intra-class variance of the shared feature space with a triplet distribution loss. We apply the approach to vision-based robot decluttering, where a mobile robot picks objects from a cluttered floor and sorts them into respective bins [10, 11, 12]. We seek to train the deep models with simulated images of object meshes and use a small number of labeled real world images to mitigate the domain shift between the image and the object distributions of the simulation and the real environments.



conventional supervised learning on labeled source data

conventional domain alignment with marginal distributions

DIRL: domain and class alignment with marginal and class distributions

Figure 1: Domain-invariant representation learning on 2D synthetic data with 2 classes: *(top)* conventional supervised learning on the source domain does not generalize to the target domain drawn from a different distribution, *(middle)* marginal distributions alignment across domains leads to cross mapping of class categories with negative transfer, *(bottom)* DIRL leverages upon a few labeled target examples to semantically align both the marginal and the class distributions. Note that the density plots at the top and the side indicate the marginal domain distributions, while the contour plots in the middle represent the class conditional distributions.

This paper makes three contributions: 1) we illustrate the limitations of cross label mismatch and/or label shift with unsupervised domain adaptation approaches that align marginal distributions in a shared feature space only, 2) we propose a domain-invariant representation learning (DIRL) algorithm to mitigate the covariate and conditional shift by aligning the marginal and the conditional distributions across domains, and making them disjoint with a novel triplet distribution loss in a shared feature space, and 3) we present experiments on MNIST benchmark domains and sim-to-real transfer of a single-shot object recognition model for mobile vision-based decluttering suggesting performance improvement over state-of-the-art domain adaptation methods.

2 Related Work

Domain Adaptation with Adversarial Learning: A popular class of domain adaptation methods learn invariant transferable representations by matching feature and/or pixel distributions with similarity metrics like maximum mean discrepancy [13], associative adaptation [14], reconstruction from latent features [15, 16], and/or adversarial learning [6, 17, 7, 8, 18]. A common trend among these prior methods is to align the marginal distributions across the domains. This, however, does not guarantee that that the samples from the same class across the domains are mapped nearby in the feature space (see Fig. 1 for an illustrative example). The lack of semantic alignment is a major limitation when the feature representation has conditional and/or label shift across the domains.

Label and Conditional Domain Adaptation: An under explored line of work addresses matching the conditional or the label distributions in the feature space. Prior work in this direction make use of linear projections in the shared feature space [19, 20], estimate importance weights for label shift [21, 22], use domain discriminators with class specific adaptation [23, 24], combine class predictions with shared features as input to the discriminator [25], maximize the conditional discrepancy with two classifiers [9], augment domain adversarial training with conditional entropy regularization [26], bound the density ratio with asymmetrical distribution alignment [27], and align the weighted source and target distribution with importance weights under generalized label shift [28].

Learning class conditional distributions of the unlabeled target domain requires pseudo-labels to encourage a low-density separation between classes [29]. Tuning the thresholds for reliable prediction of pseudo-labels on unlabeled target data using only the source domain can be cumbersome and domain-specific. Semi-supervised methods provide a practical alternative when target data is scarce (for example, when acquired via kinesthetic or teleoperation interfaces). Prior examples of semi-supervised domain adaptation include coupling of domain and class input to the discriminator [30] and minimax conditional entropy optimization [31].

In comparison to these approaches, the proposed domain-invariant represent learning approach provisions for both marginal and conditional alignment of domain distributions, while encouraging the feature representation of each class to be disjoint with a triplet distribution loss. We leverage upon a few target examples that stabilizes adaptation to challenging domains with large domain shift.

Sim-to-Real Transfer: Perception and control policies learned in simulation often do not generalize to the real robots due to the modeling inaccuracies. Domain randomization methods [2, 3, 32, 33, 34] treat the discrepancy between the domains as variability in the simulation parameters, assuming the real world distribution as one such randomized instance of the simulation environment. In contrast, domain adaptation methods learn an invariant mapping function for matching distributions between the simulator and the robot environment. Related examples to the work presented here include using synthetic objects to learn a vision based grasping model [35, 36], sim-to-real transfer of visuomotor policies for goal-directed reaching movement by adversarial learning [37], adapting dynamics in reinforcement learning [38], and adapting object recognition model in new domains [39, 40, 41]. This paper focuses on a sample-efficient approach to learn invariant features across domains while reducing the cost of labeling real data. We show state-of-the-art performance on MNIST benchmarks and sim-to-real transfer of a single-shot object recognition model used in vision based surface decluttering with a mobile robot.

3 Problem Statement

We consider two environments: one belonging to the simulator or source domain D_S comprising of the dataset $\{(\boldsymbol{x}_i^S, \boldsymbol{y}_i^S)\}_{i=1}^{N_S}$ and the other belonging to the real or target domain D_T samples $\{(\boldsymbol{x}_i^T, \boldsymbol{y}_i^T)\}_{i=1}^{N_T}$ with a few labeled samples $N_S \gg N_T$. The samples are drawn from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space and the superscripts S and T indicate the draw from two different distributions of source and target random variables $(X^S \times Y^S)$ and $(X^T \times Y^T)$ respectively. Each output labeling function is represented as $\pi : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$, with π_S and π_T as the output policy of the source and the target domain. Let $\mathcal{Z}$ denote the intermediate representation space induced from $\mathcal{X}$ by a feature transformation $g : \mathcal{X} \rightarrow \mathcal{Z}$, which is mapped to the output conditional distribution by $f : \mathcal{Z} \rightarrow \mathcal{Y}$ under the transformation $X \xrightarrow{g} Z \xrightarrow{f} Y$. The composite function $f \circ g$ defines the policy π of a domain, and the loss of a candidate policy $\pi \in \mathcal{H}$ represents the disagreement with respect to the source policy π_S on domain D_S as $\epsilon_S(\pi) := \epsilon_S(\pi, \pi_S) = \mathbb{E}_{\boldsymbol{x} \sim D_S} \, [|\pi(\boldsymbol{x}) - \pi_S(\boldsymbol{x})|]$. For example, loss function for classification on target domain is represented by $\epsilon_T(\pi, \pi_T) = \mathbb{E}_{\boldsymbol{x} \sim D_T} \, [\mathbb{I}(\pi(\boldsymbol{x}) \neq \pi_T(\boldsymbol{x}))]$, while for regression as $\mathbb{E}_{\boldsymbol{x} \sim D_T} \, [\|\pi(\boldsymbol{x}) - \pi_T(\boldsymbol{x}))\|_2]$.

The goal is to learn the policy $\pi \in \mathcal{H}$ using the labeled source examples and a few or no labeled target examples such that the error on the target domain is low. We seek to minimize the joint discrepancy across the domains in a shared feature space and map the output policy on the shared features to minimize the target error. Denoting the joint distribution of input domain and output labels across source and target domain as $\Pr(X^S, Y^S) = \Pr(Y^S | X^S) \cdot \Pr(X^S)$, and $\Pr(X^T, Y^T) = \Pr(Y^T | X^T) \cdot \Pr(X^T)$ respectively, the DIRL objective is to minimize,

$$
\begin{aligned}
&\left| \log \Pr(X^S, Y^S) - \log \Pr(X^T, Y^T) \right| = \left| \log \Pr(Y^S \mid X^S) + \log \Pr(X^S) \right| - \left| \log \Pr(Y^T \mid X^T) + \log \Pr(X^T) \right|, \\
&= \left| \log \Pr(X^S) - \log \Pr(X^T) \right| + \left| \log \Pr(Y^S \mid X^S) - \log \Pr(Y^T \mid X^T) \right|, \\
&\approx \left| g(X^S) - g(X^T) \right| + \left| f \circ g(X^S) - f \circ g(X^T) \right|, \\
&= \underbrace{d_{\Pr(g(X))} \left(D_S^Z, D_T^Z \right)}_{\text{marginal discrepancy}} + \underbrace{d_{\Pr(Y | g(X))} \left(D_S^Y, D_T^Y \right)}_{\text{conditional discrepancy}}. \qquad (1)
\end{aligned}
$$

The joint discrepancy consequently depends on both the **marginal discrepancy** and the **conditional discrepancy** between the source and the target domains defined in terms of a probability distance measure, and is minimized by aligning both the marginal and conditional distributions.

Definition 3.1 ***Marginal Distributions Alignment:*** *Given two domains* $D_S = \{(\boldsymbol{x}_i^S, \boldsymbol{y}_i^S)\}_{i=1}^{N_S}$ *and* $D_T = \{(\boldsymbol{x}_i^T, \boldsymbol{y}_i^T)\}_{i=1}^{N_T}$ *drawn from two different distributions* $\Pr(X^S, Y^S) \neq \Pr(X^T, Y^T)$ *with non-zero covariate shift* $\mathrm{KL}\left(\Pr(X^S) \| \Pr(X^T)\right) > 0$*, marginal alignment corresponds to finding the feature transformation* $g : \mathcal{X} \rightarrow \mathcal{Z}$ *such that the discrepancy between the transformed marginal distributions is minimized, i.e.,* $\Pr\left(g(X^S)\right) = \Pr\left(g(X^T)\right)$.

Definition 3.2 ***Conditional Distributions Alignment:*** *Given two domains* $D_S = \{(\boldsymbol{x}_i^S, \boldsymbol{y}_i^S)\}_{i=1}^{N_S}$ *and* $D_T = \{(\boldsymbol{x}_i^T, \boldsymbol{y}_i^T)\}_{i=1}^{N_T}$ *drawn from random variables* $(X^S \times Y^S)$ *and* $(X^T \times Y^T)$ *with different output conditional probability distributions* $\Pr\left(Y^S \mid X^S\right) \neq \Pr\left(Y^T \mid X^T\right)$*, conditional alignment corresponds to finding the transformation* $X \xrightarrow{g} Z \xrightarrow{f} Y$ *such that the discrepancy between the transformed conditional distributions is minimized, i.e.,* $\Pr\left(Y^S \mid g(X^S)\right) = \Pr\left(Y^T \mid g(X^T)\right)$.

Note that the methods aligning marginal distributions only implicitly assume the same output conditional distribution $\Pr\left(Y \mid X\right)$ across the domains for the adaptation to be effective. This assumption of different marginal distribution across domains, but similar conditional distribution is known as *covariate shift* [42]. Several methods attempt to find invariant transformation to mitigate covariate shift such that $\Pr\left(g(X)\right)$ is similar across the domains by minimizing a domain discrepancy measure [6, 7, 9]. However, it is not clear if the assumption of $\Pr\left(Y \mid g(X)\right)$ also remains the same across the domains after the transformation. As the target labels may not be available in unsupervised domain adaptation, the class conditional distributions are naively assumed to be true under the transformation $\Pr\left(Y^S \mid g(X^S)\right) = \Pr\left(Y^T \mid g(X^T)\right)$. In this paper, we consider the joint marginal and conditional distributions alignment problem across the domains. Note that it is a non-trivial problem since we have access to only a few labels in the target domain. We consider the problem in a semi-supervised setting to resolve the ambiguity in conditional distributions alignment with a few labeled target examples.

4 Theoretical Insights and Limitations

Theoretical insights of a family of domain adaptation algorithms are based on the hypothesis that a provable low target error can be obtained by minimizing the marginal discrepancies between two classifiers [43, 44, 45, 46, 47].

Theorem 4.1 (BEN-DAVIDE ET AL. [43]). *Given two domains* D_S *and* D_T*, the error of a hypothesis* $\pi \in \mathcal{H}$ *in the target domain* $\epsilon_T(\pi)$ *is bounded by the sum of: 1) the error of the hypothesis in the source domain, 2) the marginal discrepancy of the hypothesis class between the domains* $d_{\mathcal{H}\Delta\mathcal{H}}(D_S^Z, D_T^Z) := 2\sup_{\pi,\pi' \in \mathcal{H}} |\epsilon_S(\pi, \pi') - \epsilon_T(\pi, \pi')|$*, and 3) the best-in-class joint hypothesis error* $\lambda_{\mathcal{H}} = \min_{\pi \in \mathcal{H}} \left[\epsilon_S(\pi) + \epsilon_T(\pi)\right]$*,*

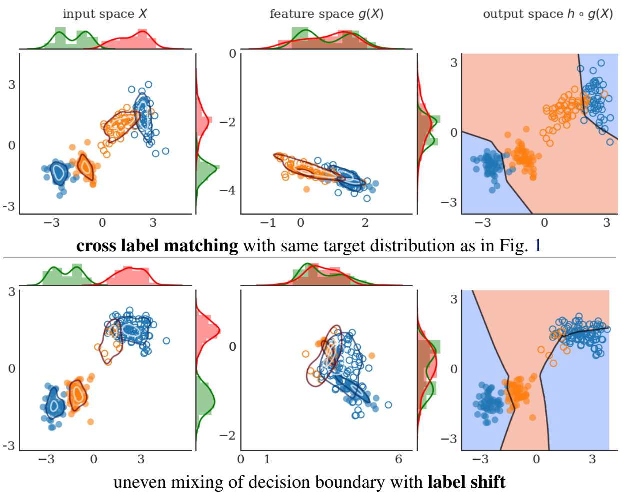
$$\epsilon_T(\pi) \leq \epsilon_S(\pi) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S^Z, D_T^Z) + \lambda_{\mathcal{H}}. \tag{2}$$

The $d_{\mathcal{H}\Delta\mathcal{H}}$ divergence can be empirically measured by training a classifier that discriminates between source and target instances, and subsequently minimized by aligning the marginal distributions between (unlabeled) source and target instances. The joint hypothesis error $\lambda_{\mathcal{H}}$ is widely assumed to be small, i.e., there exists a policy that performs well on the induced feature distributions of the source and the target examples after marginal alignment. More generally, the optimal joint error $\lambda_{\mathcal{H}}$ represents the cross-domain performance of the optimal policies $\min\{\epsilon_S(\pi_T), \epsilon_T(\pi_S)\}$. Hence, the upper bound on the target error can more appropriately be represented as,

$$\epsilon_T(\pi) \leq \epsilon_S(\pi) + d_{\mathcal{H}\Delta\mathcal{H}}(D_S^Z, D_T^Z) + \min\{\mathbb{E}_{D_S}|\pi_S - \pi_T|, \mathbb{E}_{D_T}|\pi_S - \pi_T|\}. \tag{3}$$

In Fig. 1 (middle) showing conventional domain adaptation by aligning marginal distributions, the cross-overlapping area between class categories in the shared feature space (blue and orange) represents the optimal joint error given by $\epsilon_T(\pi_S, \pi_T)$ or $\epsilon_S(\pi_T, \pi_S)$. High joint error signifies that the conventional marginal alignment approaches fail in the presence of conditional or label shift between the domains (see also Zhao et al. [44] and Zhang and Harada [46]).

Consequently, we highlight two main limitations of aligning marginal distributions only with unsupervised domain adaptation shown in Fig. 2: 1) **cross label matching**: labels of different classes are swapped in the shared feature space, and 2) **label shift**: the distribution of class labels in the source and the target domains are different, leading to samples of one class being mixed with another class. In this work, we align both the marginal and the conditional discrepancies across domains and use a few labeled target samples to avoid ad-hoc mixing of class categories and negative transfer with cross label assignments.



cross label matching with same target distribution as in Fig. 1

uneven mixing of decision boundary with **label shift**

Figure 2: Limitations of marginal distributions alignment only: *(top)* even though the source and target distributions are the same as in Fig. 1, swapping class labels of the target domain leads to cross label matching, *(bottom)* label shift with class ratio of 1.0 for the source and 0.1 for the target leads to uneven mixing of class samples across the decision boundary.

5 DIRL Algorithm

The central idea of the DIRL approach is to minimize the joint distribution error in a shared feature space by aligning the marginal and the conditional distributions across domains. To align the marginal distributions, we use a domain classifier to discriminate between the shared features of the source and the target domains. The shared features are adapted to deceive the domain classifier with adversarial learning [6, 7, 26]. Aligning the conditional distributions, however, is non-trivial due to the lack of labeled target examples. We align the conditional distributions in a semi-supervised manner by using class-wise domain discriminators for each class in a similar spirit to Chen et al. [24]. We further use a novel triplet distribution loss to make the conditional distributions disjoint in the feature space. The overall architecture of DIRL is shown in Fig. 3.

Marginal Distributions Alignment: Given the joint distribution $\Pr(X, Y) = \Pr(Y|X) \cdot \Pr(X)$, we align the marginal distributions of the transformed source and target domain with adversarial learning. The generator $g(X)$ encodes the data in a shared feature space, and the discriminator $D(X)$ predicts the binary domain label whether the data point is drawn from the source or the target distribution. The discriminator loss $\mathcal{L}_{ma}(g, D)$ is optimized using domain labels,

$$\min_{D} \ \mathcal{L}_{ma}\Big(g\left(\boldsymbol{x}_s, \boldsymbol{x}_t\right), D\left(\boldsymbol{x}_s, \boldsymbol{x}_t\right)\Big) = -\mathbb{E}_{\boldsymbol{x}_s \sim X_s}\left[\log D\left(g(\boldsymbol{x}_s)\right)\right] - \mathbb{E}_{\boldsymbol{x}_t \sim X_t}\left[\log\left(1 - D\left(g(\boldsymbol{x}_t)\right)\right)\right]. \tag{4}$$

The generator subsequently adopts the target domain features to confuse the discriminator with inverted domain labels to avoid vanishing gradients [7]. Note that the gradient reversal layer can also be used [6]. The generator loss $\mathcal{L}_{ma}(g, D)$ adapts the feature extractor,

$$\min_{g} \ \mathcal{L}_{ma}\Big(g\left(\boldsymbol{x}_t\right), D\left(\boldsymbol{x}_s, \boldsymbol{x}_t\right)\Big) = -\mathbb{E}_{\boldsymbol{x}_t \sim X_t}\left[\log D\left(g(\boldsymbol{x}_t)\right)\right]. \tag{5}$$

Without loss of generality, we denote the adaptation of the feature extractor with respect to the target data $\boldsymbol{x}_t$ only. The objective is optimized in a minimax fashion where the discriminator maximizes the empirical marginal discrepancy for a given features distribution, and the feature extractor minimizes the discrepancy by aligning the marginal distributions.

Conditional Distributions Alignment: Conditional distributions alignment can overcome the issues of cross-label matching and the shift in labeling distributions with aligning marginal distributions alignment only. Effective alignment of conditional distributions depends upon two factors: 1) prediction of target pseudo-labels, and 2) balanced sampling of features per class across domains.

To this end, we leverage upon a few labeled target examples and train the output network with labeled source and target examples using the cross-entropy loss,

$$\mathcal{L}_{ca\_sc}\Big(f \circ g\left(\boldsymbol{x}_s, \boldsymbol{y}_s, \boldsymbol{x}_t, \boldsymbol{y}_t\right)\Big) = \mathbb{E}_{\boldsymbol{x}_s, \boldsymbol{y}_s \sim (X_s, Y_s)}\left[-\boldsymbol{y}_s \log f\left(g(\boldsymbol{x}_s)\right)\right] + \mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{y}_t \sim (X_t, Y_t)}\left[-\boldsymbol{y}_t \log f\left(g(\boldsymbol{x}_t)\right)\right]. \tag{6}$$

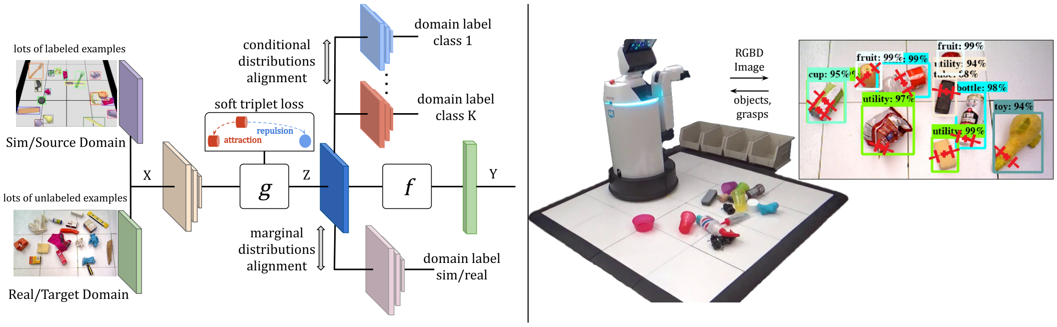


Figure 3: *(left)* DIRL aligns marginal and conditional distributions of source and target domains, and uses a soft metric learning triplet loss to make the feature distributions disjoint in a shared feature space, *(right)* experimental setup for decluttering objects into bins with HSR.

We predict the pseudo-labels for the unlabeled target data, $\hat{\boldsymbol{y}}_t = \arg\max f\left(g(\boldsymbol{x}_t)\right)$, by querying the network pre-trained on labeled data at an earlier stage during training and retain only top-n pseudo-labels of each class category based on their confidence. We sample with replacement to create a balanced mini-batch with half source and half labeled target examples, and augment the mini-batch with pool of pseudo-labeled target examples after the pre-training stage only.

A minimax game with adversarial learning aligns the conditional distribution with a domain discriminator for each class. First, the class discriminator $C(X)$ estimates the conditional discrepancy with respect to the source and the target data for a fixed feature extractor. Second, the generator adapts the feature extractor to minimize the conditional discrepancy for a fixed discriminator. Formally, the adversarial loss for each (ground-truth and predicted) class $k = 1 \ldots |\mathcal{Y}|$ is,

$$\begin{aligned}
\min_{C} \; \mathcal{L}_{ca_k}\left(g(\boldsymbol{x}_s^{(k)}, \boldsymbol{x}_t^{(k)}), C(\boldsymbol{x}_s^{(k)}, \boldsymbol{x}_t^{(k)}\right) &= -\mathbb{E}_{\boldsymbol{x}_s^{(k)} \sim X_s}\left[\log C\left(g(\boldsymbol{x}_s^{(k)})\right)\right] - \mathbb{E}_{\boldsymbol{x}_t^{(k)} \sim X_t}\left[\log\left(1 - C\left(g(\boldsymbol{x}_t^{(k)})\right)\right)\right], \\
\min_{g} \; \mathcal{L}_{ca_k}\left(g(\boldsymbol{x}_s^{(k)}, \boldsymbol{x}_t^{(k)}), C(\boldsymbol{x}_s^{(k)}, \boldsymbol{x}_t^{(k)}\right) &= -\mathbb{E}_{\boldsymbol{x}_t^{(k)} \sim X_t}\left[\log C\left(g(\boldsymbol{x}_t^{(k)})\right)\right].
\end{aligned} \tag{7}$$

Minimizing the conditional discrepancy penalizes the feature extractor to separate the cross domain overlap for each class to give low joint error $\epsilon_T(\pi_s, \pi_T)$ in Eq. (3) for provably effective adaptation to the target domain.

Triplet Distribution Loss: To increase the inter-class variance and reduce the intra-class variance, we introduce a variant of triplet loss [48, 49] that operates on distributions of anchor, positive and negative examples, instead of tuples of individual examples (see also [50]). Given M labeled examples in a mini-batch, the loss posits that the Kullback-Leibler (KL)-divergence between anchor and positives distribution in the shared feature space is less than the KL-divergence between anchor and negatives distribution by some constant margin $\alpha_{\text{tl}} \in \mathbb{R}_+$. Mathematically,

$$\mathcal{L}_{tl} = \overbrace{\sum_{a=1}^{M}}^{\text{all anchors}} \left[\overbrace{\frac{1}{M_p - 1} \sum_{\substack{p=1 \\ p \neq a}}^{M_p} \text{KL}\left(\mathcal{N}\left(\bar{g}(\boldsymbol{x}_a), \sigma^2\right) \middle\| \mathcal{N}\left(\bar{g}(\boldsymbol{x}_p), \sigma^2\right)\right)}^{\text{all positives}} - \overbrace{\frac{1}{M_n} \sum_{n=1}^{M_n} \text{KL}\left(\mathcal{N}\left(\bar{g}(\boldsymbol{x}_a), \sigma^2\right) \middle\| \mathcal{N}\left(\bar{g}(\boldsymbol{x}_n), \sigma^2\right)\right)}^{\text{all negatives}} + \alpha_{\text{tl}} \right]_+ , \tag{8}$$

where $\{.\}_+$ is the hinge loss, $\bar{g}(\boldsymbol{x})$ is normalized to extract scale-invariant features similar to [48], $\mathcal{N}\left(\bar{g}(\boldsymbol{x}_a), \sigma^2\right)$ is short for distribution of examples, $\left\{\mathcal{N}\left(\bar{g}(\boldsymbol{x}_i); \bar{g}(\boldsymbol{x}_a), \sigma^2\right) = \frac{\exp^{-\frac{1}{\sigma^2}\|\bar{g}(\boldsymbol{x}_i) - \bar{g}(\boldsymbol{x}_a)\|_2^2}}{\sum_{j=1}^{K} \exp^{-\frac{1}{\sigma^2}\|\bar{g}(\boldsymbol{x}_j) - \bar{g}(\boldsymbol{x}_a)\|_2^2}}\right\}_{i=1}^{K}$, with a Gaussian situated on normalized anchor example in the feature space $\bar{g}(\boldsymbol{x}_a)$, M_p and M_n are the number of positive and negative examples in a mini-batch, and $\sigma^2 \in \mathbb{R}$ is the hyper-parameter to control the variance of the Gaussian distribution. For each anchor, the positive examples belong to the same class as that of anchor, while the negative examples are sampled from other classes. The soft variant of triplet loss encourages the features distribution to be robust to outliers, while increasing the inter-class variance and reducing the intra-class variance across similar examples in the shared feature space.

Overall Algorithm: The overall DIRL algorithm comprises of the classification loss on the labeled source and target examples, marginal and conditional distributions alignment loss on the features generator, and triplet distribution loss on the labeled source and target examples. Given the weight coefficients $\lambda_i > 0$ of the respective losses, the overall loss function $\mathcal{L}_{\text{DIRL}}$ that DIRL optimizes is,

$$\mathcal{L}_{\text{DIRL}} = \lambda_1 \mathcal{L}_{ca\_sc}\Big(f \circ g\left(\boldsymbol{x}_s, \boldsymbol{y}_s, \boldsymbol{x}_t, \boldsymbol{y}_t\right)\Big) + \lambda_2 \mathcal{L}_{ma}\Big(g\left(\boldsymbol{x}_t\right), D\left(\boldsymbol{x}_s, \boldsymbol{x}_t\right)\Big) + \lambda_3 \sum_{k=1}^{|\mathcal{Y}|} \mathcal{L}_{ca_k}\Big(g(\boldsymbol{x}_t^{(k)}), D(\boldsymbol{x}_s^{(k)}, \boldsymbol{x}_t^{(k)})\Big) + \lambda_4 \mathcal{L}_{tl}\Big(g\left(\boldsymbol{x}_s, \boldsymbol{y}_s, \boldsymbol{x}_t, \boldsymbol{y}_t\right)\Big). \quad (9)$$

6 Experiments, Results and Discussions

In this section, we first benchmark the DIRL algorithm on digits domains, followed by sim-to-real transfer of vision-based decluttering with a mobile robot (see Appendix for 2D synthetic example in Fig. 1). We empirically investigate what representations transfer better with unsupervised approaches, and the effect of a few labeled target examples in transfer learning across domains.

6.1 Digits Datasets

We compare the DIRL approach with state-of-the-art methods including DANN [6], associative domain adaptation (ADA) [14], reconstruction based domain adaptation (RDA) [16], MCD [9], and FADA [30] on six source→target benchmarks, namely: MNIST→MNISTM, MNIST→USPS, SVHN→MNIST, USPS→SVHN, USPS→MNIST, and MNIST→SVHN.

Table 1: Average test accuracy on target domains of Digits datasets with unsupervised and semi-supervised domain adaptation. DIRL consistently performs well across all target domains in comparison to other baselines. *results from [30].

Methods		MNIST→ MNISTM	MNIST→ USPS	SVHN→ MNIST	USPS→ SVHN	USPS→ MNIST	MNIST→ SVHN
unsupervised							
RDA		0.534	0.823	0.627	0.155	0.715	0.195
DANN		0.798	0.873	0.674	0.153	0.751	0.194
Triplet		0.704	0.886	0.805	0.180	0.832	0.212
ADA		**0.895**	0.212	0.960	0.256	0.570	**0.359**
MCD		0.770	**0.941**	**0.978**	**0.288**	**0.932**	0.330
semi-supervised							
DANN	1	0.758	0.872	0.764	0.151	0.777	0.228
	5	0.763	0.920	0.793	0.252	0.850	0.353
	10	0.796	0.923	0.806	0.354	0.914	0.405
FADA*	1	–	0.891	0.728	0.275	0.811	0.377
	5		0.934	0.861	0.379	0.911	0.461
	7		0.944	0.872	0.429	0.915	0.470
DIRL	1	0.786	0.894	0.773	0.400	0.949	0.495
	5	0.941	0.946	0.864	0.610	0.945	0.683
	10	**0.948**	**0.951**	**0.903**	**0.802**	**0.962**	**0.837**

Results in Table 1 suggest that the unsupervised methods aligning marginal distributions only often do not perform well when the target domain discrepancy increases relative to the source domain. MCD performs better among unsupervised approaches by minimizing the conditional discrepancy loss using two classifiers, but gives unsatisfactory results with challenging adaptation situations such as USPS→SVHN and MNIST→SVHN. DIRL addresses the large domain shift by aligning both the marginal and the conditional distributions using only a few target examples, and consistently outperforms the compared unsupervised and semi-supervised DANN and FADA approaches. As an example, MNIST → SVHN accuracy increases by 28.2% from 1-shot to 5-shot and by 12.0% from 5-shot to 10-shot target examples of each class.

6.2 Vision-Based Decluttering by Sim-to-Real Transfer

Robots picking diversely shaped and sized novel objects in cluttered environments has a wide range of near-term applications in homes, schools, warehouses, offices and retail stores. We consider this scenario with a mobile Toyota Human Support Robot (HSR) that observes the state of the floor as a RGB image $\boldsymbol{I}_t^c \in \mathbb{R}^{640\times480\times3}$ and a depth image $\boldsymbol{I}_t^d \in \mathbb{R}^{640\times480}$. The robot recognizes the objects $\{o_i\}_{i=1}^N$ as belonging to the object categories $o_i \in \{1 \ldots |\mathcal{O}|\}$, and subsequently plans a grasp action $\boldsymbol{y}_t \in \mathbb{R}^4$ corresponding to the 3D object position and the planar orientation of the most likely recognized object. After grasping an object, the robot places the object into appropriate bins (see Fig. 3 *(right)* for an overview).

In this work, we investigate the feasibility of vision-based decluttering with a mobile robot by sim-to-real transfer with the proposed DIRL approach in a semi-supervised manner. We simulate the decluttering environment in Pybullet similar to the setup in Fig. 3, and collect $20,000$ synthetic RGBD images of cluttered object meshes on the floor, each containing $5-25$ objects drawn from a distribution of screwdriver, wrench, fruit, cup, bottle, assembly part, hammer, scissors, tape, toy, tube and utility object meshes collected from publicly available repositories. We vary the camera viewpoint, the background texture and color of object meshes in each image and store the ground-truth bounding box locations, object categories and analytically evaluated grasps for uniformly sampled antipodal pairs on the object meshes in an image. Additionally, we collect 212 RGBD images with the HSR on 1.2 sq. meter white tiled floor in a similar manner from a distribution of 102 household and machine shop objects, and hand-label the bounding boxes and object categories.

Table 2: Performance evaluation of domain-invariant object recognition by sim-to-real transfer on target test set using mean Average Precision (mAP), classification accuracy on synthetic test images $\mathrm{sim\_eval}$, real test images $\mathrm{real\_eval}$ and silhouette score SS. DIRL performs better than other compared approaches across both domains.

Methods	**mAP**	**sim_eval**	**real_eval**	**SS**
Sim Only	0.13	95.7	26.8	0.08
Real Only	0.62	24.6	85.9	0.42
Sim + Real	0.33	95.5	70.6	0.19
Triplet	0.52	94.5	76.2	0.35
DANN	0.61	93.2	84.4	0.30
MCD	0.65	94.1	89.6	0.48
DIRL	**0.69**	**94.2**	**91.0**	**0.69**

We modify the single shot multi-box detector (SSD) [51] with focal loss and feature pyramid as the base model for domain-invariant object recognition. Domain classifiers for marginal and conditional discrepancy are added on top of feature pyramid networks (see supplementary materials for architecture and training details). Results with 127 labeled target examples on test set are summarized in Table 3. Note that the Silhouette score (SS) metric in Table 3 measures the tightness of a cluster relative to the other clusters without using any labels (unsupervised); while the classification accuracy and the mAP are supervised metrics that use the target test set labels. We observe that the object recognition model trained on synthetic data only gives poor performance on real data with 26.8% accuracy, in comparison to 85.9% accuracy obtained with training on real labeled data only. Naively combining the synthetic and real data in a mini-batch is also sub-optimal. Using triplet loss on labeled source examples preserves the structure of the features for transfer to real examples. DANN improves performance in both domains by aligning marginal distributions. MCD further improves the performance in domain adaptation with conditional alignment of distributions. DIRL outperforms the compared approaches by combining marginal and conditional distributions alignment with triplet distribution loss.

Decluttering with Toyota HSR: We test the performance on the Toyota HSR for picking objects from the floor and depositing them in target bins as shown in Fig. 3 *(right)*. We load $5-25$ objects in a bin from a set of 65 physical objects and drop them on the floor in front of the robot. The objects may overlap after dropping; a pushing primitive is used to singulate the cluttered objects if the overlap is more than a threshold. The domain-invariant object recognition model gives 89.4% accuracy in real experiments. The cropped depth image from the output bounding box of the object recognition model is fed as input to the grasp planning model trained on simulated depth images, adapted from [12, 36]. The grasping network gives 86.5% accuracy of picking up the identified object. Without using the grasping network and only grasping orthogonal to the principal axis of the point cloud of the predicted object location gives 76.2% accuracy. We observe that the robot performs well in grasping compliant objects and objects with well-defined geometry such as cylinders, screwdrivers, tape, cups, bottles and utilities; while assembly parts and small bowls in inverted pose induced repeated failures in grasping the target object (see supplementary materials for details).

7 Conclusion

In this paper, we have presented a sample-efficient domain-invariant representation learning approach for adapting deep models to a new environment. The proposed DIRL approach overcomes the ambiguity in transfer with unsupervised approaches by mitigating both the marginal and the conditional discrepancies across domains with a small amount of labeled target examples, and uses a triplet distribution loss to make the feature distributions disjoint in a shared feature space. Experiments with digit domains yield state-of-the-art performance across challenging transfer scenarios with large domain shift, while vision-based decluttering with a mobile robot suggest the feasibility of grasping diversely shaped and sized novel objects by sim-to-real transfer. In future work, we plan to investigate domain-invariance across multiple environments with a fleet of robots and close the real-to-sim loop in transferring models across new environments.

Acknowledgments

This research was performed at UC Berkeley in affiliation with the AutoLab, Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), and the Real-Time Intelligent Secure Execution (RISE) Lab. The authors would like to thank Daniel Zeng, Matthew Trepte, Kate Sanders, Yi Liu, Lerrel Pinto, Trevor Darrell and Ken Goldberg for their feedback.

References

[1] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018. doi:10.1177/0278364917710318.

[2] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907, 2017.

[3] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *CoRR*, abs/1710.06537, 2017.

[4] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand, 2019.

[5] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *CoRR*, abs/1802.03601, 2018. URL `http://arxiv.org/abs/1802.03601`.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, Jan. 2016. ISSN 1532-4435.

[7] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *CoRR*, abs/1702.05464, 2017. URL `http://arxiv.org/abs/1702.05464`.

[8] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017. URL `http://arxiv.org/abs/1711.03213`.

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

[10] M. Gupta, J. Müller, and G. S. Sukhatme. Using manipulation primitives for object sorting in cluttered environments. *IEEE Trans. Automation Science and Engineering*, 12(2):608–614, 2015.

[11] A. Gupta, A. Murali, D. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *CoRR*, abs/1807.07049, 2018.

[12] A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, and K. Goldberg. A Fog Robotics Approach to Deep Robot Learning: Application to Object Recognition and Grasp Planning in Surface Decluttering. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2019.

[13] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Neural Information Processing Systems*, NIPS'16, pages 136–144, 2016.

[14] P. Häusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. *CoRR*, abs/1708.00938, 2017.

[15] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems 29*, pages 343–351. 2016.

[16] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. *CoRR*, abs/1607.03516, 2016.

[17] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *CoRR*, abs/1612.05424, 2016.

[18] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL `http://arxiv.org/abs/1703.10593`.

[19] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.

[20] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, pages 2839–2848, 2016.

[21] Z. C. Lipton, Y. Wang, and A. J. Smola. Detecting and correcting for label shift with black box predictors. *CoRR*, abs/1802.03916, 2018.

[22] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar. Regularized learning for domain adaptation under label shifts. *CoRR*, abs/1903.09734, 2019. URL `http://arxiv.org/abs/1903.09734`.

[23] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[24] Y. Chen, W. Chen, Y. Chen, B. Tsai, Y. F. Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. *CoRR*, abs/1704.08509, 2017.

[25] M. Long, Z. CAO, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Neural Information Processing Systems*, pages 1640–1650. 2018.

[26] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A dirt-t approach to unsupervised domain adaptation, 2018.

[27] Y. Wu, E. Winston, D. Kaushik, and Z. C. Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. *CoRR*, abs/1903.01689, 2019. URL `http://arxiv.org/abs/1903.01689`.

[28] R. T. des Combes, H. Zhao, Y.-X. Wang, and G. Gordon. Domain adaptation with conditional distribution matching and generalized label shift, 2020.

[29] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. *CoRR*, abs/1901.00976, 2019. URL `http://arxiv.org/abs/1901.00976`.

[30] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto. Few-shot adversarial domain adaptation. *CoRR*, abs/1711.02536, 2017.

[31] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko. Semi-supervised domain adaptation via minimax entropy. *CoRR*, abs/1904.06487, 2019. URL `http://arxiv.org/abs/1904.06487`.

[32] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. D. Ratliff, and D. Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *CoRR*, abs/1810.05687, 2018.

[33] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. *CoRR*, abs/1812.07252, 2018.

[34] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, K. Yamane, S. Iba, J. Canny, and K. Goldberg. Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[35] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *Int. J. Rob. Res.*, 27(2):157–173, 2008. doi:10.1177/0278364907087172.

[36] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. 2017.

[37] F. Zhang, J. Leitner, Z. Ge, M. Milford, and P. Corke. Adversarial discriminative sim-to-real transfer of visuo-motor policies. *The International Journal of Robotics Research*, 38(10-11):1229–1245, 2019.

[38] B. Eysenbach, S. Asawa, S. Chaudhari, R. Salakhutdinov, and S. Levine. Off-dynamics reinforcement learning: Training for transfer with domain classifiers, 2020.

[39] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision*, ECCV, pages 213–226, 2010.

[40] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool. Domain adaptive faster R-CNN for object detection in the wild. *CoRR*, abs/1803.03243, 2018. URL `http://arxiv.org/abs/1803.03243`.

[41] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin. Adapting object detectors via selective cross-domain alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051, 9780262170055.

[43] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, May 2010. doi:10.1007/s10994-009-5152-4.

[44] H. Zhao, R. T. des Combes, K. Zhang, and G. J. Gordon. On learning invariant representation for domain adaptation. *CoRR*, abs/1901.09453, 2019.

[45] F. D. Johansson, D. Sontag, and R. Ranganath. Support and invertibility in domain-invariant representations. In *International Conference on Artificial Intelligence and Statistics*, pages 527–536, 2019.

[46] D. Zhang and T. Harada. A general upper bound for unsupervised domain adaptation, 2019.

[47] B. Li, Y. Wang, S. Zhang, D. Li, T. Darrell, K. Keutzer, and H. Zhao. Learning invariant representations and risks for semi-supervised domain adaptation, 2020.

[48] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. URL `http://arxiv.org/abs/1503.03832`.

[49] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*, 2015.

[50] P. Florence. *Dense Visual Learning for Robot Manipulation*. PhD thesis, Massachusetts Institute of Technology, 2019.

[51] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[52] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

[53] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, 1994. doi:10.1109/34.291440.

[54] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

[55] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. URL `http://arxiv.org/abs/1512.02325`.

[56] B. Staub, A. K. Tanwani, J. Mahler, M. Breyer, M. Laskey, Y. Takaoka, M. Bajracharya, R. Siegwart, and K. Goldberg. Dex-Net MM: Deep Grasping for Surface Decluttering with a Low-Precision Mobile Manipulator. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1–7, 2019.

[57] A. K. Tanwani, R. Anand, J. E. Gonzalez, and K. Goldberg. Rilaas: Robot inference and learning as a service. *IEEE Robotics and Automation Letters*, 5(6):4423–4430, 2020.

A Experimental Details

A.1 2D Synthetic Example

We consider a 2-dimensional problem comprising of 2 classes. Source data is generated from Gaussian distributions with means centered around $[-2.5, -1.5]$ and $[-1.0, -1.0]$ respectively for the two classes. Covariance matrices for all the Gaussians are represented by: $\sigma^2 \boldsymbol{I} + \boldsymbol{W}\boldsymbol{W}^\top$ with $\sigma^2 = 0.1$ and $\boldsymbol{W} = 0.25\,\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Similarly, Target data is drawn from Gaussian distributions with respective class means centered around $[1.0, 1.0]$ and $[2.5, 1.5]$. We sample 1000 instances from the source and the target distribution each for training, and 100 instances from the target distribution are used for the testing the learned model. We use the same weight for all the constituent loss functions with $\lambda_1 \ldots \lambda_4 = 1.0$. Network architecture comprises of 3 hidden layers of 7 neurons with ReLU activation for each of the shared feature space, output classifier, domain discriminator and class-conditional domain discriminator. During training, we use a mini-batch of 80 samples comprising of half source, labeled target (if applicable) and unlabeled target examples. For conditional domain discriminators, we sub-sample a new mini-batch comprising of half labeled source and half labeled target examples for each class. We use Adam optimizer with a learning rate of 0.0001 for 60K iterations. Note that we do not use labeled target instances for classification loss, and only use them for triplet distribution and conditional discriminator loss, in order to better analyze the effect of constituent losses on the target accuracy. Resulting animations are available on the project website: `https://sites.google.com/view/dirl`

A.2 Digits Datasets

We choose four commonly used digits datasets for benchmarks (see Fig. 4): MNIST [52], MNISTM [6], USPS [53], and SVHN [54]. We select first dataset as the source and the second one as the target dataset for adaptation in the following configurations: **MNIST→MNISTM**, **MNIST→USPS**, **SVHN→MNIST**, **USPS→SVHN**, **USPS→MNIST**, **MNIST→SVHN**.

Experimental details are as follows: Input dimension of all datasets are reshaped to $28 \times 28 \times 3$, output classifier dimension is 10 corresponding to the unit digits. Network architecture consists of the shared feature space: 3 sets of convolution layers with $32, 64, 128$ hidden layers each separated by a max pool layer with stride of length 2. The output of the final layer is flattened to 256 dimensional space and fed to the output classification network, domain discriminator and conditional domain discriminator with 3 dense layers of size $100, 50$ and the output dimensions of $10, 2, 2$ respectively.

Table 3: Digits Dataset Instances.

Dataset	**Train Instances**	**Test Instances**
MNIST	$55,000$	$10,000$
MNISTM	$55,000$	$10,000$
SVHN	$73,257$	$26,032$
USPS	$7,291$	$2,007$

We use the same weight for all the constituent loss functions with $\lambda_1 \ldots \lambda_4 = 1.0$. We perform experiments with $\{1, 5, 10\}$ labeled instances to evaluate the proposed approach. Overall batch size comprises of half-source and half-target examples. Source examples are always labeled, while target examples are 25% labeled and 75% unlabeled within a mini-batch. No pseudo-labels are used for the target examples with the digits experiments. Adam optimizer with learning rate of 0.001 for $10,000$ iterations to minimize the constituent loss functions. We also attempted instance normalization with this setup similar to [26], but it did not have any significant effect on the results.

We compare the average test accuracy of the target domains in unsupervised and semi-supervised setting. Methods aligning marginal distributions only such as DANN often do not perform well due to lack of conditional alignment across source domains. Metric learning with triplet loss on the source domain increases the separation between class categories, which helps in transfer to the new environment. Adding reconstruction loss on top of DANN to force the feature transformation to be invertible decreases the performance on the target domain. This performance degradation is likely due to the additional constraints of having distinct features for each sample, making it difficult to align the marginal distributions as also suggested in Johansson et al. [45]. Associative domain adaptation imposes a cyclic loss to bring source and target examples close in the shared feature space, however, yields unreliable performance across datasets. MCD performs better across unsupervised baselines by minimizing the conditional discrepancy loss using two classifiers, however, gives unsatisfactory results with challenging adaptation situations such as **USPS→SVHN** and **MNIST→SVHN** like other unsupervised domain adaptation methods.

DIRL addresses the limitations of the existing approaches and outperforms other unsupervised and semi-supervised approaches with DANN, FADA in 1-shot, 5-shot and 10-shot scenario. DIRL uses a few target labels for effective conditional distributions alignment (see Fig. 4 for a qualitative comparison). Results presented in this paper do not make use of the pseudo-labeled target data. Note that DIAL performs well across all datasets, especially on challenging problems of **USPS→SVHN** and **MNIST→SVHN** with large domain shift by aligning the marginal and the conditional distributions using only a few samples for the target class. Specifically, **MNIST** → **SVHN** increases 28.2% from 1-shot to 5-shot and 12.0% from 5-shot to 10-shots.

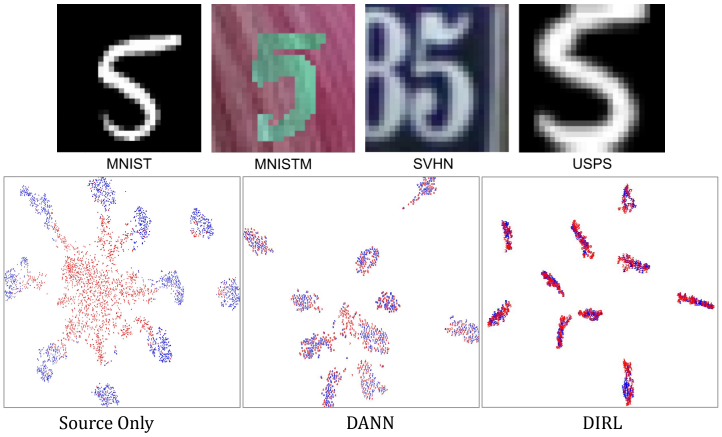


Figure 4: *(top)* Sample image of label 5 from MNIST, MNISTM, SVHN and USPS digits datasets, *(bottom)* T-SNE visualization of **MNIST→MNISTM** (source in blue, target in red). DIRL compactly clusters the class distributions across datasets for transfer learning in comparison to DANN and source only transfer.

A.3 Vision-Based Decluttering by Sim-to-Real Transfer

Simulation and Real Dataset: We simulate the decluttering environment in a Pybullet simulator. The simulated dataset comprises of $20,000$ synthetic RGB and depth images of cluttered object meshes on floor. Each image randomly contains between $5-25$ objects that are split across 12 categories, namely screwdriver, wrench, fruit, cup, bottle, assembly part, hammer, scissors, tape, toy, tube and utility. The object meshes are collected from Turbosquid, Kit, 3dNet, ShapeNet repositories. We use domain randomization to vary the camera viewpoint, background texture and color of object meshes in each image, and store the ground-truth bounding box locations, object categories, segmentation masks and analytically evaluated grasps for uniformly sampled antipodal pairs on the object meshes in an image.

The physical dataset comprises of 212 real RGB and depth images collected with the Toyota HSR looking at 1.2 sq. meter white tiled floor from a distribution of 102 household and machine shop objects, and hand-label the bounding boxes and object categories. We hand-label the bounding boxes and object categories similar to the synthetic classes above.

Object Recognition Network: We use the MobileNet-Single Shot MultiBox Detector (SSD) [55, 51] algorithm with focal loss and feature pyramids as the base model for object recognition. The input RGB image $\in \mathbb{R}^{640\times480\times3}$ is fed to a pre-trained VGG16 network, followed by feature resolution maps and a feature pyramid network. The feature pyramid network produces feature maps across 5 resolutions: 80×80, 40×40, 20×20, 10×10, 5×5 that are concatenated before being fed to the class prediction and box prediction networks.

We modify the base model by flattening the output of the class predictions with background features and adding domain and class discriminators on top. Domain discriminator consists of three fully connected layers of size $1024, 200, 100$ and 2 output neurons for classifying simulator vs real images. Class conditional domain discriminator consists of two fully connected layers of size 200, 100 before the output layer of 2 neurons. We use an input batch size of 8 with 4 clones that are split across half-sim, 25% labeled and 25% unlabeled real data. We sample 128 most like class predictions (without background) in the same proportion as the batch size for triplet distribution loss, marginal and conditional discriminators. After pretraining the network for 4000 iterations, we assigned pseudo-labels to the unlabeled real images and used them with triplet distribution loss and conditional discriminators. The class embeddings were uniformly sampled for each conditional discriminator to mitigate the effect of imbalanced proportion of classes.

Results with 127 labeled target examples on test set are summarized in the main body of the paper. We use three performance metrics, namely: mean Average Precision (mAP), classification accuracy on real and sim images on a held-out test set, and Silhouette score (SS). Note that the Silhouette score (SS) is an unsupervised metric measures the tightness of a cluster relative to the other clusters without using any labels; while the classification accuracy and the mAP are supervised metrics that use the labels of the test set. We observe that the object recognition model trained on synthetic data only gives poor performance on real data with 26.8% accuracy, in comparison to 85.9% accuracy obtained with training on real labeled data only. Naively combining the synthetic and real data in a mini-batch is also sub-optimal. Using triplet loss on labeled source and target examples preserves the structure of the features for transfer to real examples. DANN improves performance in both domains by aligning marginal distributions. MCD further improves the performance in

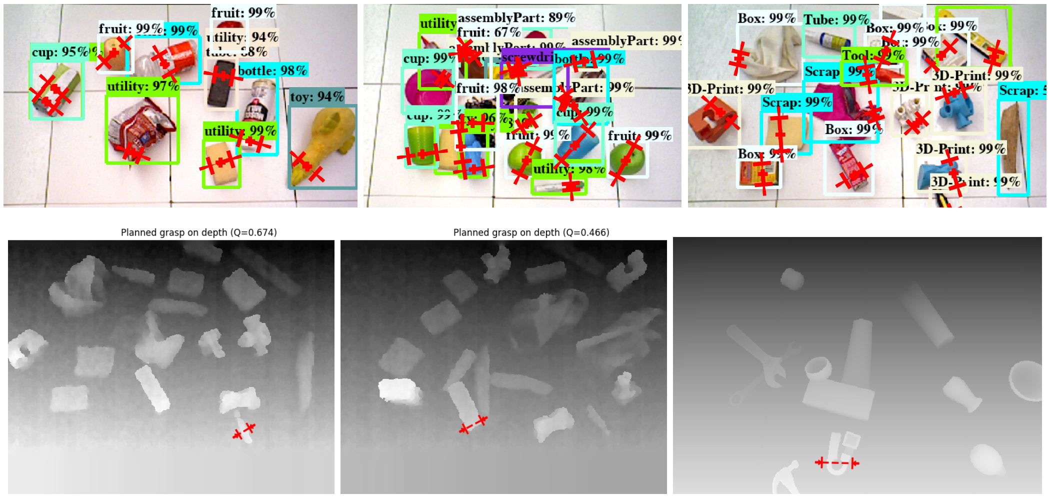


Figure 5: Vision-Based decluttering examples of object recognition and grasp planning. Output bounding boxes from the object recognition network are fed to the grasp planning network to yield the most likely grasp to success shown with red whisker plots on top.

domain adaptation with conditional alignment of distributions. DIRL outperforms the compared approaches by combining marginal and conditional distributions alignment with triplet distribution loss.

Grasp Planning Network: The grasp planning model comprises of two parts: *grasp sampling* and *grasp evaluation*. The grasp planning model samples antipodal pairs on the cropped depth image of the object and and feeds them to a convolutional neural network to predict the probability of successful grasp as determined by the wrench resistance metric. Good grasps are successively filtered with a cross-entropy method to return the most likely grasp for the robot to pick and place the object into corresponding bin.

The cropped depth image is centered around the sampled grasp center and orientation to make the network predictions rotation-invariant. The network takes a downsampled 96×96 depth image around the grasp center, and processes it with a convolution layer of size 9×9 with 16 filters and 3 convolution layers of size 5×5 with 16 filters each, followed by fully connected layer of size 128. The average height of the grasp center region from the camera is processed through a separate fully connected layer of size 16 before it is concatenated with the image stream features and a fully connected layer of size 128, followed by the output layer of 2 neurons to predict the grasp quality score $\in [0, 1]$.

We test the performance of the trained models on the Toyota HSR for picking objects from the floor and depositing them in target bins [12, 56, 57]. We load $5 - 25$ objects in a bin from a set of 65 physical objects and drop them on the floor in front of the robot. The objects may overlap after dropping; a pushing primitive is used to singulate the cluttered objects if the overlap is more than a threshold. The domain-invariant object recognition model gives 89.4% accuracy in real experiments. The grasping network gives 86.5% accuracy of picking up the identified object. Without using the grasping network and only grasping orthogonal to the principal axis of the point cloud of the predicted object location gives 76.2% accuracy. We observe that the robot performs well in grasping compliant objects and objects with well-defined geometry such as cylinders, screwdrivers, tape, cups, bottles and utilities; while assembly parts and small bowls in inverted pose induced repeated failures in grasping the target object (see video on the project website for more details).